USEFUL FORMULA FIELDS

Available in: All Editions

<table>
<thead>
<tr>
<th>User Permissions Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>To view formula field details:</td>
</tr>
<tr>
<td>To create, change, or delete formula fields:</td>
</tr>
</tbody>
</table>

Use the following formula samples when creating custom formula fields. For samples of other types of formulas, see “Examples of Validation Rules” in the Salesforce.com online help and “Useful Default Field Value Formulas” in the Salesforce.com online help.

This document contains the following categories of custom formula samples:

- Account Management
- Account Media Service Links
- Case Management
- Commission Calculations
- Contact Management
- Data Categorization
- Date Calculations
- Discounting
- Employee Services
- Expense Tracking
- Financial Calculations
- Image Links
- Integration Links
- Lead Management
- Metrics
- Opportunity Management
- Pricing
- Project Management
- Scoring Calculations
Account Management

Account Rating

This formula evaluates Annual Revenue, Billing Country, and Type, and assigns a value of “Hot,” “Warm,” or “Cold.”

```
IF (AND (AnnualRevenue > 10000000,
IF(ISPICKVAL(Type, "Manufacturing Partner"), "Hot",
IF(OR (ISPICKVAL (Type, "Channel Partner/Reseller"),
ISPICKVAL(Type, "Installation Partner")), "Warm", "Cold")),
"Cold")
```

For details about using these functions, see IF on page 53, AND on page 40, CONTAINS on page 45, CASE on page 42, and ISPICKVAL on page 64.

In addition, you can reference this Account Rating formula field from the contact object using cross-object formulas.

Account.Account_Rating__c

Account Region

This formula returns a text value of “North,” “South,” “East,” “West,” or “Central” based on the Billing State/Province of the account.

```
IF(ISBLANK(BillingState), "None",
"Other"))))))
```

For details about using these functions, see IF on page 53, LEN on page 66, and CONTAINS on page 45.

Contract Aging

This formula calculates the number of days since a contract with an account was activated. If the contract Status is not “Activated,” this field is blank.

```
IF(ISPICKVAL(Contract_Status__c, "Activated"),
NOW() - Contract_Activated_Date__c, null)
```

For details about using these functions, see IF on page 53, ISPICKVAL on page 64, and NOW on page 74.

Contract Approval Process Aging

This formula calculates how many days a contract is in the approval process. This example is a number formula field on contracts that uses a custom date field called Date in approval.

Useful Formula Fields
**TODAY()** - *Date_in_approval__c*

For details about using this function, see **TODAY** on page 84.

**Month of Last Account Activity**

This formula field displays the month of the last account activity or “None” if there are no activities for the account.

```
CASE(MONTH(LastActivityDate),
  1, "January",
  2, "February",
  3, "March",
  4, "April",
  5, "May",
  6, "June",
  7, "July",
  8, "August",
  9, "September",
 10, "October",
11, "November",
12, "December",
"None")
```

For details about using these functions, see **CASE** on page 42 and **MONTH** on page 73.

**Month of Service-Level Agreement Expiration**

This formula returns the month that your service-level agreement expires. This example uses a custom date field called **SLA Expiration Date**.

```
MONTH(SLAExpirationDate__c)
```

For details about using this function, see **MONTH** on page 73.

**Account Media Service Links**

**BBC™ News Search**

This formula creates a link to a BBC news search site based on the **Account Name**.

```
HYPERLINK("http://newssearch.bbc.co.uk/cgi-bin/search/results.pl?scope=newsifs;tab=news&q="&Name, "BBC News")
```

**Bloomberg™ News Search**

This formula creates a link to an account's ticker symbol on the Bloomberg website.

```
```
CNN™ News Search
This formula creates a link to a CNN news search site using the Account Name.

HYPERLINK( "http://websearch.cnn.com/search/search?source=cnn&invocationType=search&Prop=sites&web&query="&Name, "CNN News")

MarketWatch™ Search
This formula creates a link to an account's ticker symbol on the Marketwatch.com website.


Google™ Search
This formula creates a link to a Google search site using the Account Name.

HYPERLINK( "http://www.google.com/search?en&q="&Name, "Google")

Google News Search
This formula creates a link to a Google news search site using the Account Name.

HYPERLINK( "http://www.google.com/news?en&q="&Name, "Google News")

Yahoo!™ Search
This formula creates a link to a Yahoo! search site using the Account Name.

HYPERLINK( "http://search.yahoo.com/search?p="&Name, "Yahoo Search")

Yahoo News Search
This formula creates a link to a Yahoo! news search site using the Account Name.


For details about using the function used in these formulas, see HYPERLINK on page 51.
Case Management

**Autodial**

This formula creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace "servername" and "call" with the name of your dialing tool and the command it uses to dial. The merge field, `Id`, inserts the identifier for the contact, lead, or account record. The first `Phone` merge field tells the dialing tool what number to call and the last `Phone` merge field uses the value of the `Phone` field as the linkable text the user clicks to dial.

```
HYPERLINK("http://servername/call?id=" & Id & "&phone=" & Phone, Phone)
```

For details about using this function, see HYPERLINK on page 51.

**Case Aging (Assignments)**

Use this example of a custom formula field called `Days Open` to display different text depending on the number of days a case has been open:

```
CASE(Days_Open__c, 3, "Reassign", 2, "Assign Task", "Maintain")
```

The following text is displayed:

- “Reassign” for any case open three days.
- “Assign Task” for any case open two days.
- “Maintain” for all other cases.

For details about using this function, see CASE on page 42.

**Case Aging (Open Cases)**

This formula calculates the number of days a case has been open. If the case is closed, it sets the result to null. Add this formula to a related list as the sort column to quickly see which open cases have been open the longest. The formula returns zero if the case has been open for less than 24 hours.

```
IF(IsClosed, null, NOW() - CreatedDate )
```

For details about using these functions, see IF on page 53 and NOW on page 74.

**Case Aging (Open and Closed Cases)**

This formula calculates the number of days a closed case was open or the number of days an open case has been open since the date the case was created. The formula returns zero if the case has been open for less than 24 hours.

```
IF(IsClosed, ROUND(ClosedDate - CreatedDate, 0), ROUND((NOW() - CreatedDate),0))
```
For details about using these functions, see IF on page 53, ROUND on page 80, and NOW on page 74.

**Case Categorization**

This formula displays a text value of “RED,” “YELLOW,” or “GREEN,” depending on the value of a case age custom text field.

```
IF(DaysOpen__c > 20, "RED",
   IF(DaysOpen__c > 10, "YELLOW",
      "GREEN")
)
```

For details about using this function, see IF on page 53.

**Case Data Completeness Tracking**

This formula calculates the percentage of specific custom fields that contain data. The formula checks the values of two custom number fields: Problem Num and Severity Num. If the fields are empty, the formula returns the value “0.” The formula returns a value of “1” for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

```
(IF(ISBLANK(Problem_Num__c), 0, 1) + IF(ISBLANK(Severity_Num__c), 0, 1)) * 50
```

For details about using these functions, see IF on page 53 and ISBLANK on page 58.

**Case Due Date Calculation**

This formula sets the due date of a case based on the priority. If it is high, the due date is two days after it opens. If it is medium, the due date is five days after opening. Otherwise, the due date is seven days.

```
CASE (Priority,
    "High", CreatedDate + 2,
    "Medium",CreatedDate + 5,
    CreatedDate + 7)
```

For details about using this function, see CASE on page 42.

**Suggested Agent Prompts**

This formula prompts an agent with cross-sell offers based on past purchases.

```
CASE(Product_Purch__c,
    "Printer", "Extra toner cartridges", "Camera", "Memory cards", "Special of the day")
```

For details about using this function, see CASE on page 42.
**Suggested Offers**

This formula suggests a product based on the support history for a computer reseller. When the Problem custom field matches a field, the formula field returns a suggestion.

```plaintext
CASE(Problem__c, 
  "Memory", "Suggest new memory cards", 
  "Hard Drive failure", "Suggest new hard drive with tape backup", 
  "")
```

For details about using this function, see CASE on page 42.

---

**Commission Calculations**

**Commission Amounts for Opportunities**

The following is a simple formula where commission is based on a flat 2% of the opportunity Amount.

```plaintext
IF(ISPICKVAL(StageName, "Closed Won"), 
  ROUND(Amount *0.02, 2), 0)
```

This example calculates the commission amount for any opportunity that has a “Closed Won” stage. The value of this field will be the amount times 0.02 for any closed/won opportunity. Open or lost opportunities will have a zero commission value.

For details about using these functions, see IF on page 53, ISPICKVAL on page 64, and ROUND on page 80.

**Commission Deal Size**

This formula calculates a commission rate based on deal size, returning a 9% commission rate for deals over 100,000 and an 8% commission rate for smaller deals.

```plaintext
IF(Amount > 100000, 0.09, 0.08 )
```

For details about using this function, see IF on page 53.

**Commission Greater Than or Equal To**

This formula assigns the “YES” value to opportunities with a commission greater than or equal to one million. Note, this is a text formula field on opportunities that uses a custom currency field called Commission.

```plaintext
IF(Commission__c >= 1000000, "YES", "NO")
```

For details about using this function, see IF on page 53.
**Commission Maximum**

This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

\[
\text{MAX}($\text{User.Commission\_Percent\_c} \times \text{Price}, \text{Price} \times \text{Account\_Discount\_c}, 100)
\]

For details about using this function, see MAX on page 70.

**Contact Management**

**Contact's Account Creation Date**

This date formula displays the account's Created Date field on the contacts page.

\[\text{Account.CreatedDate}\]

**Contact's Account Discount Percent**

This percent formula displays the account's Discount Percent field on the contacts page.

\[\text{Account.Discount\_Percent\_c}\]

**Contact's Account Name**

This formula displays the standard Account Name field on the contacts page.

\[\text{Account.Name}\]

**Contact's Account Phone**

This formula displays the standard Account Phone field on the contacts page.

\[\text{Account.Phone}\]

**Contact's Account Rating**

Use this formula to display the Account Rating field on the contacts page.


For details about using this function, see CASE on page 42.

**Contact's Account Website**

This formula displays the standard Account Website field on the contacts page.

\[\text{Account.Website}\]
If the account website URL is long, use the HYPERLINK function to display a label such as “Click Here” instead of the URL. For example:

```plaintext
IF(Account.Website='', '',
    IF(
        OR(LEFT(Account.Website, 7) = "http://", LEFT(Account.Website, 8) = "https://"),
        HYPERLINK(Account.Website, "Click Here"),
        HYPERLINK("http://" & Account.Website, "Click Here")
    )
)
```

This formula also adds the necessary "http://" or "https://" before a URL if neither were included in the URL field.

For details about using this function, see HYPERLINK on page 51.

**Contact's Age**

Use this formula to calculate a person’s age based on a standard field called Birthdate. The person’s Birthdate is subtracted from today’s date, which returns the number of days since the person’s Birthdate. This number is divided by the number of days in a year and rounded down to the nearest integer.

```
FLOOR((TODAY()-Birthdate)/365.2425)
```

For details about using these functions, see FLOOR on page 49 and TODAY on page 84.

**Contact's Birthday**

This formula displays the value “Yes” if the contact’s birthday falls in the current calendar month.

```
IF (MONTH(Birthdate) = MONTH(TODAY()), "Yes", "")
```

For details about using these functions, see IF on page 53, MONTH on page 73, and TODAY on page 84.

**Contact Identification Numbering**

This formula displays the first five characters of the contact’s last name and the last four characters of the contact’s social security number separated by a dash. Note that this example uses a text custom field called SSN on contacts.

```
TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN__c, 4))
```

For details about using these functions, see TRIM on page 85, LEFT on page 65, and RIGHT on page 79.

**Contact Preferred Phone**

This formula displays the contact’s preferred contact method in a contact related list—work phone, home phone, or mobile phone—based on a selected option in a Preferred Phone custom picklist.

```
CASE(Preferred_Phone__c, "Work", "w. " & Phone, "Home", "h. " & HomePhone, "Mobile", "m. " & MobilePhone, "No Preferred Phone")
``
For details about using this function, see **CASE** on page 42.

**Contact Priority**

This formula assesses the importance of a contact based on the account rating and the contact's title. If the account rating is **Hot** or the title starts with **Executive**, then the priority is high (P1). If the account rating is **Warm** or the title starts with **VP**, then the priority is medium (P2), and if the account rating is **Cold** then the priority is low (P3).

```
IF(OR(ISPICKVAL(Account.Rating, "Hot"), CONTAINS(Title, "Executive")), "P1",
    IF(OR(ISPICKVAL(Account.Rating, "Warm"), CONTAINS(Title, "VP")), "P2",
        IF(ISPICKVAL(Account.Rating, "Cold"), "P3",
            "P3")
    )
)
```

**Contact Yahoo! ID**

This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called **Yahoo Name** on contacts where you can store the contact’s Yahoo! Messenger ID.

```
HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,
    IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m=g&t=0", "Yahoo"))
```

For details about using these functions, see **HYPERLINK** on page 51 and **IMAGE** on page 55.

**Dynamic Address Formatting**

This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country for the account.

```
CASE(ShippingCountry,
    "USA",
    ShippingStreet & BR() &
    ShippingCity & "",
    " & ShippingState & ", &
    ShippingPostalCode & BR()
& ShippingCountry,
    "France",
    ShippingStreet & BR() &
    ShippingPostalCode & " " &
    ShippingCity & BR() &
    ShippingCountry, "etc")
```

For details about using this function, see **CASE** on page 42 and **BR** on page 41.
**Telephone Country Code**

This formula determines the telephone country code of a contact based on the Mailing Country of the mailing address.

```plaintext
CASE(MailingCountry, 
    "USA", "1", 
    "Canada", "1", 
    "France", "33", 
    "UK", "44", 
    "Australia", "61", 
    "Japan", "81", 
    "?"
)
```

For details about using this function, see `CASE` on page 42.

**Unformatted Phone Number**

This formula removes the parentheses and dash characters from North American phone numbers. This is necessary for some auto-dialer software.

```plaintext
IF(Country_Code__c = "1", MID( Phone ,2, 3) & MID(Phone,7,3) & MID(Phone,11,4), Phone)
```

For details about using these functions, see `IF` on page 53 and `MID` on page 71.

**Data Categorization**

**Deal Size Large and Small**

This formula displays “Large Deal” for deals over one million dollars or “Small Deal” for deals under one million dollars.

```plaintext
IF(Sales_Price__c > 1000000, 
    "Large Deal", 
    "Small Deal")
```

For details about using this function, see `IF` on page 53.

**Deal Size Small**

This formula displays “Small” if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

```plaintext
IF(AND(Price<1,Quantity<1),"Small", null)
```

For details about using these functions, see `IF` on page 53 and `AND` on page 40.

**Product Categorization**

This formula checks the content of a custom text field named `Product_Type` and returns “Parts” for any product with the word “part” in it. Otherwise, it returns “Service.” Note that the values are
case sensitive, so if a Product_Type field contains the text “Part” or “PART,” this formula returns “Services.”

```
IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")
```

For details about using these functions, see IF on page 53 and CONTAINS on page 45.

## Date Calculations

### Birthday in Current Year Accounting for Leap Years

This formula returns the date of a person’s birthday in the current year, even if the person’s birthday is on February 29th in a leap year.

```
IF(AND(MONTH(Birthdate) = 2, DAY(Birthdate) = 29),
   IF(OR(MOD(YEAR(DATEVALUE(NOW())), 400) = 0,
      AND(MOD(YEAR(DATEVALUE(NOW())), 4) = 0, MOD(YEAR(DATEVALUE(NOW())), 100) <> 0)),
      DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate), DAY(Birthdate)),
      DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate + 1), 28)),
   DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate), DAY(Birthdate))))
```

### Day of Week (number)

This formula calculates today’s day of the week as a number (0 = Sunday, 1 = Monday, 2 = Tuesday, and so on).

```
MOD(TODAY() - DATE(1900, 1, 7), 7)
```

Similarly, this formula substitutes the TODAY() function shown in the previous example with a custom date field called Sign Up Date. It returns the day of the week as a number for that field.

```
MOD(Sign_Up_Date__c - DATE(1900, 1, 7), 7)
```

For details about using these functions, see MOD on page 72, TODAY on page 84, and DATE on page 46.

### Day of Week (text)

This formula calculates today’s day of the week and displays it as text. To determine the day of the week for a date field, use the formula below and replace “TODAY()” with that date field.

```
CASE(
   MOD(TODAY() - DATE(1900, 1, 7), 7),
   0, "Sunday",
   1, "Monday",
   2, "Tuesday",
   3, "Wednesday",
   4, "Thursday",
   5, "Friday",
   6, "Saturday", "Error")
```
For details about using these functions, see **CASE** on page 42, **MOD** on page 72, **TODAY** on page 84, and **DATE** on page 46.

### Day of Year

This formula calculates today’s numeric day of the year (a number between 1 and 365).

\[
\text{TODAY()} - \text{DATE(YEAR(TODAY())), 1, 1) + 1}
\]

For details about using these functions, see **TODAY** on page 84, **DATE** on page 46, and **YEAR** on page 91.

### Days Until End of Month

This formula displays the number of days between a specific date and the end of the month in which the date occurs.

\[
\text{IF(MONTH(CloseDate)=12, DATE(YEAR(CloseDate),12,31) - CloseDate, DATE(YEAR(CloseDate), MONTH(CloseDate)+1,1) - CloseDate-1)}
\]

For details about using these functions, see **IF** on page 53, **MONTH** on page 73, **DATE** on page 46, and **YEAR** on page 91.

### Time of Day

This formula returns the time of day in Greenwich Mean Time (GMT), for example: “20:04 PM”.

\[
\text{MID (TEXT (Due_Date_Time__c), 12, 5)}
\]

For details about using these functions, see **MID** on page 71 and **TEXT** on page 82.

### Discounting

#### Maintenance and Services Discount

This formula field uses two custom currency fields: **Maintenance Amount** and **Services Amount**. It displays “Discounted” on an opportunity if its maintenance amount and services amount do not equal the opportunity **Amount** standard field value. Otherwise, it displays "Full Price."

\[
\text{IF(Maintenance_Amount__c + Services_Amount__c <> Amount, "Discounted", "Full Price")}
\]

For details about using this function, see **IF** on page 53.
Opportunity Discount Amount

This formula calculates the difference of the opportunity Amount less the Discount Amount. Note that Discount Amount is a custom currency field on opportunities.

\[ \text{Amount} - \text{Discount Amount} \_\_\_c \]

For details about using this operator, see - (Subtract) on page 33.

Opportunity Discount Rounded

Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called Discount Percent.

\[ \text{ROUND}(\text{Amount} - \text{Amount} \times \text{Discount Percent} \_\_\_c, 2) \]

For details about using this function, see ROUND on page 80.

Opportunity Discount with Approval

This formula adds a “Discount Approved” checkbox to an opportunity. It uses conditional logic to check the value of the approval flag before calculating the commission.

\[
\text{IF}(\text{Discount Approved}\_\_\_c, \text{ROUND}(\text{Amount} - \text{Amount} \times \text{Discount Percent} \_\_\_c, 2), \text{Amount})
\]

For details about using these functions, see IF on page 53 and ROUND on page 80.

Employee Services

Bonus Calculation

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have custom number field for Number of Employees, a custom percent field for Bonus Percent, and currency custom fields for the employee's Gross and company's Performance.

\[
\text{MIN}(\text{Gross} \_\_\_c \times \text{Bonus Percent} \_\_\_c, \text{Performance} \_\_\_c / \text{Number of Employees} \_\_\_c)
\]

For details about using this function, see MIN on page 72.

Employee 401K

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or $250, whichever is less. It assumes you have custom currency field for Contribution.

\[
\text{MIN}(250, \text{Contribution} \_\_\_c / 2)
\]

For details about using this function, see MIN on page 72.
**Hours Worked Per Week**
This formula uses a custom tab to enable time tracking of hours worked per day. It uses a formula field to sum the hours per week.

\[ \text{MonHours\_c + TuesHours\_c + WedsHours\_c + ThursHours\_c + FriHours\_c} \]

For details about using this operator, see + (Add) on page 33.

**Total Pay Amount**
This formula determines total pay by calculating regular hours multiplied by a regular pay rate, plus overtime hours multiplied by an overtime pay rate.

\[
\begin{align*}
\text{Total Pay} &= \text{IF(Total_Hours\_c <= 40, Total_Hours\_c * Hourly_Rate\_c,} \\
&\phantom{= \text{IF(Total_Hours\_c <= 40,} } 40 * \text{Hourly_Rate}\_c + \\
&\phantom{= \text{IF(Total_Hours\_c <= 40,} } (\text{Total_Hours}\_c - 40) * \text{Overtime_Rate}\_c) \\
\end{align*}
\]

For details about using this function, see IF on page 53.

**Expense Tracking**

**Expense Identifier**
This formula displays the text “Expense-” followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

\"Expense-\" & Trip_Name\_c & "-" & ExpenseNum\_c

For details about using this operator, see - (Subtract) on page 33.

**Mileage Calculation**
This formula calculates mileage expenses for visiting a customer site at 35 cents a mile.

\[ \text{Miles_Driven}\_c * 0.35 \]

For details about using this operator, see * (Multiply) on page 34.

**Financial Calculations**

**Compound Interest**
This formula calculates the interest, you will have after T years, compounded M times per year.

\[ \text{Principal}\_c * \left( 1 + \text{Rate}\_c / M \right)^{T \times M} \]
For details about using these operators, see * (Multiply) on page 34, / (Divide) on page 34, and ^ (Exponentiation) on page 35.

**Compound Interest Continuous**

This formula calculates the interest that will have accumulated after T years, if continuously compounded.

\[ \text{Principal} \times \text{EXP}(\text{Rate} \times T) \]

For details about using this function, see EXP on page 47.

**Consultant Cost**

This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of $1200 per day. Note that Consulting Days is a custom field on opportunities.

\[ \text{Consulting Days} \times 1200 \]

For details about using this operator, see * (Multiply) on page 34.

**Gross Margin**

This formula provides a simple calculation of gross margin. In this formula example, Total Sales and Cost of Goods Sold are custom currency fields.

\[ \text{Total Sales} - \text{Cost of Goods Sold} \]

For details about using this operator, see - (Subtract) on page 33.

**Gross Margin Percent**

This formula calculates the gross margin based on a margin percent.

\[ \text{Margin percent} \times \text{Items Sold} \times \text{Price item} \]

For details about using this operator, see * (Multiply) on page 34.

**Payment Due Indicator**

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field on contracts.

\[ \text{BLANKVALUE(Payment Due Date, Start Date +5)} \]

For details about using this function, see BLANKVALUE on page 41.

**Payment Status**

This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, it returns the text “PAYMENT OVERDUE” and if not, it leaves the field blank. This example
uses a custom date field called Payment Due Date and a text custom field called Payment Status on contracts.

```plaintext
IF(
    AND(Payment_Due_Date__c < TODAY(),
    ISPICKVAL(Payment_Status__c, "UNPAID"),
    "PAYMENT OVERDUE",
    null )
)
```

For details about using these functions, see IF on page 53, AND on page 40, TODAY on page 84, and ISPICKVAL on page 64.

**Image Links**

**Yahoo! Instant Messenger™ Image**

This formula displays an image that indicates whether a contact or user is currently logged in to Yahoo! Instant Messenger. Clicking the image launches the Yahoo! Instant Messenger window. This formula uses a custom text field called Yahoo Name to store the contact or user's Yahoo! ID.

```plaintext
IF(ISBLANK(Yahoo_Name__c),"",HYPERLINK("ymsgr:sendIM?"& Yahoo_Name__c,
IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m=g&t=0", "")))
```

For details about using these functions, see IF on page 53, LEN on page 66, HYPERLINK on page 51, and IMAGE on page 55.

**“Skype Me”™ Auto Dialer Button**

This formula displays an image that looks like a push button. Clicking the button automatically dials the specified phone number.

```plaintext
HYPERLINK("callto://" & "+1" & Phone,
IMAGE("http://goodies.skype.com/graphics/skypeme_btn_small_blue.gif",
"Click to Skype"))
```

For details about using these functions, see HYPERLINK on page 51 and IMAGE on page 55.

**Flags for Case Priority**

This formula displays a green, yellow, or red flag image to indicate case priority.

```plaintext
IMAGE(
    CASE( Priority,
        "Low", "/img/samples/flag_green.gif",
        "Medium", "/img/samples/flag_yellow.gif",
        "High", "/img/samples/flag_red.gif",
        "/s.gif"),
    "Priority Flag")
```

For details about using this function, see IMAGE on page 55.
**Color Squares for Case Age**

This formula displays a 30 x 30 pixel image of a red, yellow, or green, depending on the value of a `Case Age` custom text field.

```plaintext
IF( Case_Age__c > 20, IMAGE("/img/samples/color_red.gif", "red", 30, 30), IF( Case_Age__c > 10, IMAGE("/img/samples/color_yellow.gif", "yellow", 30, 30), IMAGE("/img/samples/color_green.gif", "green", 30, 30) ) )
```

For details about using these functions, see `IF` on page 53 and `IMAGE` on page 55.

**Traffic Lights for Status**

This formula displays a green, yellow, or red traffic light images to indicate status, using a custom picklist field called `Project Status`. Use this formula in list views and reports to create a “Status Summary” dashboard view.

```plaintext
IMAGE( CASE(Project_Status__c, "Green", "/img/samples/light_green.gif", "Yellow", "/img/samples/light_yellow.gif", "Red", "/img/samples/light_red.gif", "/s.gif"), "status color")
```

For details about using these functions, see `IMAGE` on page 55 and `CASE` on page 42.

**Stars for Ratings**

This formula displays a set of one to five stars to indicate a rating or score.

```plaintext
IMAGE( CASE(Rating__c, "1", "/img/samples/stars_100.gif", "2", "/img/samples/stars_200.gif", "3", "/img/samples/stars_300.gif", "4", "/img/samples/stars_400.gif", "5", "/img/samples/stars_500.gif", "/img/samples/stars_000.gif"), "rating")
```

For details about using these functions, see `IMAGE` on page 55 and `CASE` on page 42.

**Consumer Reports™-Style Colored Circles for Ratings**

This formula displays a colored circle to indicate a rating on a scale of one to five, where solid red is one, half red is two, black outline is three, half black is four, and solid black is five.

```plaintext
IMAGE( CASE(Rating__c, "1", "/img/samples/rating1.gif", "2", "/img/samples/rating2.gif", "3", "/img/samples/rating3.gif", "4", "/img/samples/rating4.gif", "5", "/img/samples/rating5.gif", "/s.gif"), "rating")
```
For details about using these functions, see IMAGE on page 55 and CASE on page 42.

**Horizontal Bars to Indicate Scoring**

This formula displays a horizontal color bar (green on a white background) of a length that is proportional to a numeric score. In this example, the maximum length of the bar is 200 pixels.

```
IMAGE("/img/samples/color_green.gif", "green", 15, Industry_Score__c * 2) &
IMAGE("/s.gif", "white", 15, 200 - (Industry_Score__c * 2))
```

For details about using this function, see IMAGE on page 55.

**Integration Links**

**Application API Link**

This formula creates a link to an application outside Salesforce.com, passing the parameters so that it can connect to Salesforce.com via the Force.com Web Services API and create the necessary event.

```
HYPERLINK ("https://www.myintegration.com?sId=" & GETSESSIONID() & 
"&rowID=" & Name & "&action=CreateTask","Create a Meeting Request")
```

For details about using these functions, see HYPERLINK on page 51 and GETSESSIONID on page 51.

**Shipment Tracking Integration**

This formula creates a link to FedEx, UPS, or DHL shipment tracking websites, depending on the value of a **Shipping Method** custom picklist field. Note that the parameters shown in this example for FedEx, UPS, and DHL websites are illustrative and do not represent the correct parameters for all situations.

```
CASE(Shipping_Method__c, 
"Fedex", 
"UPS",
HYPERLINK("http://wwwapps.ups.com/WebTracking/processInputRequest?HTMLVersion=5.0&sort_by=status&loc=en_US&InquiryNumber1=" & tracking_id__c & 
"&track.x=32&track.y=7", "Track"), 
"DHL",
HYPERLINK("http://track.dhl-usa.com/TrackByNbr.asp?Shipmen\tNumber=" & tracking_id__c,"Track"), "")
```

For details about using these functions, see CASE on page 42 and HYPERLINK on page 51.
**Skype Auto Dialer Integration**

This formula creates a linkable phone number field that automatically dials the phone number via the Skype VOIP phone application. It requires installation of the Skype application (a third-party product not provided by salesforce.com) on your desktop.

\[
\text{HYPERLINK}("\text{callto://}+" \text{Country Code}_\text{c} \text{ & Phone Unformatted}_\text{c}, \text{Phone})
\]

For details about using this function, see HYPERLINK on page 51.

---

**Lead Management**

**Lead Aging (for open leads)**

This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead is not open, this field is blank.

\[
\text{IF(ISPICKVAL(Status, "Open"), ROUND(NOW()-CreatedDate, 0), null)}
\]

For details about using these functions, see IF on page 53, ISPICKVAL on page 64, ROUND on page 80, and NOW on page 74.

**Lead Data Completeness**

This formula calculates the percent of certain lead fields that your sales personnel enter. The formula field checks the values of two custom number fields: Phone and Email. If the fields are empty, the formula returns the value “0.” The formula returns a value of “1” for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

\[
(IF(\text{Phone} = "", 0, 1) + IF(\text{Email} = "", 0, 1)) * 50
\]

For details about using this function, see IF on page 53.

**Lead Numbering**

This formula returns a number value for the text value in the auto-number field Lead Number. This can be useful if you want to use the Lead Number field in a calculation, such as round-robin or other routing purposes. Note that auto-number fields are text fields and must be converted to a number for numeric calculations.

\[
\text{VALUE(Lead\_Number\_c)}
\]

For details about using this function, see VALUE on page 89.

**Round Robin Assignment of Cases or Leads**

The following formula example for leads assumes you have three lead queues and you want to assign an equal number of incoming leads to each queue. You can also assign cases using a similar formula.

\[
\text{MOD(\text{VALUE(Lead\_Number\_c)}, 3)}
\]
This formula is for a custom formula field named `Round_Robin_ID` that assigns each lead a value of 0, 1, or 2. This formula uses a custom auto-number field called `Lead Number` that assigns each lead a sequential number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example:

- `Round_Robin_ID = 0` is assigned to Queue A
- `Round_Robin_ID = 1` is assigned to Queue B
- `Round_Robin_ID = 2` is assigned to Queue C

For details about using these functions, see `MOD` on page 72 and `VALUE` on page 89.

---

**Metrics**

**Temperature Conversion**
This formula converts Celsius degrees to Fahrenheit.

\[ 1.8 \times \text{degrees}_\text{celsius}_\text{c} + 32 \]

For details about using these operators, see `*` (Multiply) on page 34 and `+` (Add) on page 33.

**Unit of Measure Conversion**
This formula converts kilometers to miles.

\[ \text{Miles}_\text{c} / 0.621371192 \]

For details about using this operator, see `/` (Divide) on page 34.

---

**Opportunity Management**

**Days Left to Close**
This formula returns the expected number of days left to the close date of an opportunity.

\[ \text{Expected}_\text{close}_\text{date}_\text{c} - \text{TODAY()} \]

For details about using this function, see `TODAY` on page 84.
Display Close Month for Reporting Purposes

This formula returns the month in text for the close date of an opportunity. Use this example when building a custom report that groups opportunities by the month of the **Close Date**.

```
CASE(
    MONTH(CloseDate),
    1, "January",
    2, "February",
    3, "March",
    4, "April",
    5, "May",
    6, "June",
    7, "July",
    8, "August",
    9, "September",
    10, "October",
    11, "November",
    12, "December",
    "Invalid month")
```

For details about using these functions, see **CASE** on page 42 and **MONTH** on page 73.

Expected Product Revenue

This formula calculates total revenue from multiple products, each with a different probability of closing.

```
ProductA_probability__c * ProductA_revenue__c + ProductB_probability__c * ProductB_revenue__c
```

For details about using these operators, see *(Multiply)* on page 34 and *(Add)* on page 33.

Maintenance Calculation

This formula calculates maintenance fees as 20% of license fees per year. **Maintenance Years** is a custom field on opportunities.

```
Amount * Maint_Years__c * 0.2
```

For details about using this operator, see *(Multiply)* on page 34.

Monthly Subscription-Based Calculated Amounts

This formula calculates an opportunity amount based on a monthly subscription rate multiplied by the subscription period.

```
Monthly_Amount__c * Subscription_Months__c
```

For details about using this operator, see *(Multiply)* on page 34.

Monthly Value

This formula divides total yearly value by 12 months.

```
Total_value__c / 12
```
Opportunity Additional Costs

This formula calculates the sum of the opportunity Amount, maintenance amount, and services fees. Note that Maint amount and Service Fees are custom currency fields on opportunities.

\[ \text{Amount} + \text{Maint\_Amount\_c} + \text{Services\_Amount\_c} \]

For details about using this operator, see + (Add) on page 33.

Opportunity Categorization

This formula uses conditional logic to populate an Opportunity category text field, based on the value of the Amount standard field. Opportunities with amounts less than $1500 are “Category 1,” those between $1500 and $10000 are “Category 2,” and the rest are “Category 3.” This example uses nested IF statements.

\[
\text{IF}(\text{Amount} < 1500, "\text{Category 1}", \text{IF}(\text{Amount} > 10000, "\text{Category 3}", "\text{Category 2}") )
\]

For details about using this function, see IF on page 53.

Opportunity Data Completeness

This formula takes a group of opportunity fields and calculates what percent of them are being used by your sales personnel. This formula checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the Treat blank fields as blanks option under Blank Field Handling while the Advanced Formula subtab is showing.

\[
(\text{IF}(\text{ISBLANK(Maint\_Amount\_c)}, 0, 1) + \text{IF}(\text{ISBLANK(Services\_Amount\_c)}, 0, 1) + \text{IF}(\text{ISBLANK(Discount\_Percent\_c)}, 0, 1) + \text{IF}(\text{ISBLANK(Amount)}, 0, 1) + \text{IF}(\text{ISBLANK(Timeline\_c)}, 0, 1)) / 5
\]

For details about using this function, see ISBLANK on page 58.

Opportunity Expected License Revenue

This formula calculates expected revenue for licenses based on probability of closing.

\[ \text{Expected\_rev\_licenses\_c} * \text{Probability} \]

For details about using this operator, see * (Multiply) on page 34.

Opportunity Reminder Date

This formula creates reminder date based on seven days before the close date of an opportunity. Use this formula field in a workflow rule to create an event for the appropriate user to take action.

\[ \text{Reminder Date} = \text{CloseDate} - 7 \]
For details about using these operators, see = and == (Equal) on page 36 and - (Subtract) on page 33.

**Opportunity Revenue Text Display**

This formula returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the Expected Revenue of a campaign is “$200,000,” this formula field displays “200000.”

```plaintext
TEXT(ExpectedRevenue)
```

For details about using this function, see TEXT on page 82.

**Opportunity Split Credit for Sales Representatives**

This formula splits opportunity amount between multiple sales representatives. The total reps custom field indicates the total number of representatives on the deal.

```plaintext
Amount / total_reps__c
```

For details about using this operator, see / (Divide) on page 34.

**Opportunity Total Deal Size**

This formula calculates the sum of maintenance and services amounts.

```plaintext
Amount + Maint_Amount__c + Services_Amount__c
```

For details about using this operator, see + (Add) on page 33.

**Opportunity Total Price Based on Units**

This formula generates proposal pricing based on unit price and total volume.

```plaintext
Unit_price__c * Volume__c * 20
```

For details about using this operator, see * (Multiply) on page 34.

**Professional Services Calculation**

This formula estimates professional service fees at an average loaded rate of $1200 per day. Consulting Days is a custom field on opportunities.

```plaintext
Consulting_Days__c * 1200
```

For details about using this operator, see * (Multiply) on page 34.

**Stage-Based Sales Document Selection**

This formula identifies a relevant document in the Documents tab based on opportunity Stage. Use document IDs in the form of “00l30000000j7AO.”

```plaintext
CASE(StageName,
  "Prospecting", "Insert 1st Document ID",
  "Qualification", "Insert 2nd Document ID",
  "...
```

24
For details about using this function, see \texttt{CASE} on page 42.

\textbf{Sales Coach}

This formula creates a hyperlink that opens a stage-specific document stored in the Documents tab. It uses the previously defined custom formula field that identifies a document based on opportunity \texttt{Stage}. See \textbf{Stage-Based Sales Document Selection} on page 24.

\texttt{HYPERLINK("/servlet/servlet.FileDownload?file=\& Relevant\_Document\_c, "View Document in New Window")}

For details about using this function, see \texttt{HYPERLINK} on page 51.

\textbf{Shipping Cost by Weight}

This formula calculates postal charges based on weight.

\texttt{package\_weight\_c \ast cost\_lb\_c}

For details about using this operator, see \texttt{\ast (Multiply)} on page 34.

\textbf{Shipping Cost Percentage}

This formula calculates shipping cost as a fraction of \texttt{total amount}.

\texttt{Ship\_cost\_c \div total\_amount\_c}

For details about using this operator, see \texttt{/ (Divide)} on page 34.

\textbf{Tiered Commission Rates}

This formula calculates the 2\% commission amount of an opportunity that has a probability of 100\%. All other opportunities will have a commission value of zero.

\texttt{IF(Probability = 1, ROUND(Amount \ast 0.02, 2), 0)}

For details about using these functions, see \texttt{IF} on page 53 and \texttt{ROUND} on page 80.

\textbf{Total Contract Value from Recurring and Non-Recurring Revenue}

This formula calculates both recurring and non-recurring revenue streams over the lifetime of a contract.

\texttt{Non\_Recurring\_Revenue\_c + Contract\_Length\_Months\_c \ast Recurring\_Revenue\_c}

For details about using these operators, see \texttt{+ (Add)} on page 33 and \texttt{\ast (Multiply)} on page 34.
Pricing

**Total Amount**
This formula calculates a total amount based on unit pricing and total units.

\[ \text{Unit\_price\_c} \times \text{Total\_units\_c} \]

For details about using this operator, see *(Multiply)* on page 34.

**User Pricing**
This formula calculates a price per user license.

\[ \frac{\text{Total\_license\_rev\_c}}{\text{Number\_user\_licenses\_c}} \]

For details about using this operator, see *(Divide)* on page 34.

Project Management

**Calculate Intermediate Milestone from End Date**
This formula calculates intermediate milestone dates by subtracting days from the end date (for projects that are planned based on end date).

\[ \text{Release\_Date\_c} - 7 \times \text{Phase\_duration\_in\_weeks\_c} \]

For details about using this operator, see *(Multiply)* on page 34.

Scoring Calculations

**Lead Scoring**
This formula scores leads, providing a higher score for phone calls than website requests.

\[ \text{CASE(LeadSource, "Phone", 2, "Web", 1, 0)} \]

For details about using this function, see **CASE** on page 42.
Customer Success Scoring

This formula uses a simple scoring algorithm to rank customers a high score for positive survey results in Salesforce.com.

\[ \text{Survey Question 1} \times 5 + \text{Survey Question 2} \times 2 \]

For details about using these operators, see * (Multiply) on page 34 and + (Add) on page 33.

Operators and Functions

Use the following operators and functions when building formulas. Click on the name or description below to view more details. All functions are available everywhere that you can include a formula such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Note: Extraneous spaces in the samples below are ignored by Salesforce.com.

Math Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Calculates the sum of two values.</td>
</tr>
<tr>
<td>-</td>
<td>Calculates the difference of two values.</td>
</tr>
<tr>
<td>*</td>
<td>Multiplies its values.</td>
</tr>
<tr>
<td>/</td>
<td>Divides its values.</td>
</tr>
<tr>
<td>^</td>
<td>Raises a number to a power of a specified number.</td>
</tr>
<tr>
<td>()</td>
<td>Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.</td>
</tr>
</tbody>
</table>

Logical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>= and ==</td>
<td>Evaluates if two values are equivalent.</td>
</tr>
<tr>
<td>&lt; and !=</td>
<td>Evaluates if two values are not equivalent.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Evaluates if a value is less than the value that follows this symbol.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Evaluates if a value is greater than the value that follows this symbol.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Evaluates if a value is less than or equal to the value that follows this symbol.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Evaluates if a value is greater than or equal to the value that follows this symbol.</td>
</tr>
</tbody>
</table>
### Text Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;</code></td>
<td>Connects two or more strings.</td>
</tr>
</tbody>
</table>

### Date and Time Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DATE</code></td>
<td>Returns a date value from year, month, and day values you enter. Salesforce.com displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.</td>
</tr>
<tr>
<td><code>DATEVALUE</code></td>
<td>Returns a date value for a date/time or text expression.</td>
</tr>
<tr>
<td><code>DAY</code></td>
<td>Returns a day of the month in the form of a number between 1 and 31.</td>
</tr>
<tr>
<td><code>MONTH</code></td>
<td>Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.</td>
</tr>
<tr>
<td><code>NOW</code></td>
<td>Returns a date/time representing the current moment.</td>
</tr>
<tr>
<td><code>TODAY</code></td>
<td>Returns the current date as a date data type.</td>
</tr>
<tr>
<td><code>YEAR</code></td>
<td>Returns the four-digit year in number format of a given date.</td>
</tr>
</tbody>
</table>

### Informational Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BLANKVALUE</code></td>
<td>Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.</td>
</tr>
<tr>
<td><code>ISBLANK</code></td>
<td>Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.</td>
</tr>
<tr>
<td><code>ISNULL</code></td>
<td>Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.</td>
</tr>
</tbody>
</table>
**Nullfunctions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULLVALUE</td>
<td>Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression.</td>
</tr>
<tr>
<td>PRIORVALUE</td>
<td>Returns the previous value of a field.</td>
</tr>
</tbody>
</table>

**Logical Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false.</td>
</tr>
<tr>
<td>CASE</td>
<td>Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the else_result.</td>
</tr>
<tr>
<td>IF</td>
<td>Determines if expressions are true or false. Returns a given value if true and another value if false.</td>
</tr>
<tr>
<td>ISCHANGED</td>
<td>Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.</td>
</tr>
<tr>
<td>ISNEW</td>
<td>Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.</td>
</tr>
<tr>
<td>ISNUMBER</td>
<td>Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.</td>
</tr>
<tr>
<td>NOT</td>
<td>Returns FALSE for TRUE and TRUE for FALSE.</td>
</tr>
<tr>
<td>OR</td>
<td>Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false.</td>
</tr>
</tbody>
</table>

**Math Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.</td>
</tr>
<tr>
<td>CEILING</td>
<td>Rounds a number up to the nearest integer.</td>
</tr>
<tr>
<td>EXP</td>
<td>Returns a value for $e$ raised to the power of a number you specify.</td>
</tr>
<tr>
<td>FLOOR</td>
<td>Returns a number rounded down to the nearest integer.</td>
</tr>
</tbody>
</table>
### Useful Formula Fields

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LN</td>
<td>Returns the natural logarithm of a specified number. Natural logarithms are based on the constant ( e ) value of 2.71828182845904.</td>
</tr>
<tr>
<td>LOG</td>
<td>Returns the base 10 logarithm of a number.</td>
</tr>
<tr>
<td>MAX</td>
<td>Returns the highest number from a list of numbers.</td>
</tr>
<tr>
<td>MIN</td>
<td>Returns the lowest number from a list of numbers.</td>
</tr>
<tr>
<td>MOD</td>
<td>Returns a remainder after a number is divided by a specified divisor.</td>
</tr>
<tr>
<td>ROUND</td>
<td>Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.</td>
</tr>
<tr>
<td>SQRT</td>
<td>Returns the positive square root of a given number.</td>
</tr>
</tbody>
</table>

### Text Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGINS</td>
<td>Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not.</td>
</tr>
<tr>
<td>BR</td>
<td>Inserts a line break in a string of text.</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.</td>
</tr>
<tr>
<td>FIND</td>
<td>Returns the position of a string within a string of text represented as a number.</td>
</tr>
<tr>
<td>GETSESSIONID</td>
<td>Returns the user’s session ID.</td>
</tr>
<tr>
<td>HYPERLINK</td>
<td>Creates a link to a URL specified that is linkable from the text specified.</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Inserts an image with alternate text and height/width specifications.</td>
</tr>
<tr>
<td>INCLUDES</td>
<td>Determines if any value selected in a multi-select picklist field equals a text literal you specify.</td>
</tr>
<tr>
<td>ISPICKVAL</td>
<td>Determines if the value of a picklist field is equal to a text literal you specify.</td>
</tr>
<tr>
<td>LEFT</td>
<td>Returns the specified number of characters from the beginning of a text string.</td>
</tr>
<tr>
<td>LEN</td>
<td>Returns the number of characters in a specified text string.</td>
</tr>
<tr>
<td>LOWER</td>
<td>Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.</td>
</tr>
</tbody>
</table>
**Function** | **Description**
---|---
LPAD | Inserts spaces or characters you specify to the left-side of a text string.
MID | Returns the specified number of characters from the middle of a text string given the starting position.
RIGHT | Returns the specified number of characters from the end of a text string.
RPAD | Inserts blank spaces or characters that you specify to the right-side of a text string.
SUBSTITUTE | Substitutes new text for old text in a text string.
TEXT | Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in validation rules, formula fields, and field updates.
TRIM | Removes the spaces and tabs from the beginning and end of a text string.
UPPER | Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
VALUE | Converts a text string to a number.

**Summary Functions**
The following functions can only be used in the context of creating custom summary formulas for summary reports.

**Function** | **Description**
---|---
PARENTGROUPVAL | A summary function that returns the value of the selected summary field from the parent grouping level that you specify. Parent grouping levels are those above the grouping level where you chose to display the formula. For matrix reports, choose both row and column grouping levels.
PREVGROUPVAL | A summary function that returns the value of the selected summary field from the previous summary row at the grouping level that you specify. For matrix reports, the `summary_field` you choose controls whether a previous row or column is returned. The increment determines how many steps previous to the current summary field to return. The default increment is 1. The maximum increment is 12.
Advanced Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETRECORDIDS</td>
<td>Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.</td>
</tr>
<tr>
<td>LINKTO</td>
<td>Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce.com page.</td>
</tr>
<tr>
<td>REGEX</td>
<td>Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, it returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.</td>
</tr>
<tr>
<td>REQUIRESRC</td>
<td>Returns a script tag with source for a URL you specify. Use this function when referencing the Force.com AJAX Toolkit or other JavaScript toolkits.</td>
</tr>
<tr>
<td>URLFOR</td>
<td>Returns a relative URL for an action, s-control, or a file in a static resource archive in a Visualforce page.</td>
</tr>
<tr>
<td>VLOOKUP</td>
<td>Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.</td>
</tr>
</tbody>
</table>

Encoding Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML_ENCODE</td>
<td>Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (&gt;), with HTML entity equivalents, such as &gt;.</td>
</tr>
<tr>
<td>JS_ENCODE</td>
<td>Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (), before unsafe JavaScript characters, such as the apostrophe (‘).</td>
</tr>
<tr>
<td>JS_IN_HTML_ENCODE</td>
<td>Encodes text and merge field values for use in JavaScript within HTML tags by inserting escape characters before unsafe JavaScript characters and replacing characters that are reserved in HTML with HTML entity equivalents.</td>
</tr>
<tr>
<td>URL_ENCODE</td>
<td>Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. For example, blank spaces are replaced with %20, and exclamation points are replaced with %21.</td>
</tr>
</tbody>
</table>
### + (Add)

| Description: | Calculates the sum of two values. |
| Use: | value1 + value2 and replace each value with merge fields, expressions, or other numeric values. |
| Formula Field Example: | Amount + Maint_Amount__c + Services_Amount__c |
| This formula calculates the sum of the opportunity Amount, maintenance amount, and services fees. Note that Maint amount and Service Fees are custom currency fields on opportunities. |
| Report Example: | EMAIL_OPT_OUT:SUM + DO_NOT_CALL:SUM |
| calculates all Email Opt Out fields plus all Do Not Call fields on the leads in your report. |
| This formula is a number data type that returns a positive integer. |
| Validation Rule Example: | Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40 |
| Use a formula like this one in a validation rule to display the following error message when the total number of hours entered for each work day is greater than 40: “Your total hours cannot exceed 40.” This example requires five custom fields on your custom object, one for each day of work. |

### - (Subtract)

| Description: | Calculates the difference of two values. |
| Use: | value1 - value2 and replace each value with merge fields, expressions, or other numeric values. |
| Example: | Amount - Discount_Amount__c |
| This formula calculates the difference of the opportunity Amount less the Discount |
**Useful Formula Fields**

<table>
<thead>
<tr>
<th>Description</th>
<th>Use</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMOUNT:SUM - Opportunity.Discount_Amount__c:SUM</td>
<td>Calculates the difference of all Amount fields and all Discount Amount custom fields on the opportunities in your report. This formula is a currency data type that returns a currency sign and decimal places.</td>
<td>Report Example: AMOUNT:SUM - Opportunity.Discount_Amount__c:SUM calculates the difference of all Amount fields and all Discounted Amount custom fields on the opportunities in your report. This formula is a currency data type that returns a currency sign and decimal places.</td>
</tr>
</tbody>
</table>

### * (Multiply)

<table>
<thead>
<tr>
<th>Description</th>
<th>Multiplies its values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use</td>
<td>value1 * value2 and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example</td>
<td>Consulting_Days__c * 1200</td>
</tr>
<tr>
<td></td>
<td>This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of $1200 per day. Note that Consulting Days is a custom field on opportunities.</td>
</tr>
<tr>
<td>Report Example</td>
<td>RowCount * AGE:AVG calculates the record count times the average age value of all opportunities in your report. This formula is a number data type that returns a positive or negative integer or decimal.</td>
</tr>
</tbody>
</table>

### / (Divide)

<table>
<thead>
<tr>
<th>Description</th>
<th>Divides its values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use</td>
<td>value1 / value2 and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example</td>
<td>AnnualRevenue/ NumberOfEmployees</td>
</tr>
<tr>
<td></td>
<td>This formula calculates the revenue amount per employee using a currency field on accounts.</td>
</tr>
<tr>
<td></td>
<td>IF(NumberOfOpportunities &gt; 0, NumberOfWonOpportunities / NumberOfOpportunities, null)</td>
</tr>
<tr>
<td></td>
<td>This formula calculates the win rate of opportunities on a campaign.</td>
</tr>
</tbody>
</table>
% Won Opportunities

\[
\frac{\text{WON:SUM}}{\text{RowCount}} \text{ calculates the percent of won opportunities using a record count representing the number of all opportunities in your report. This formula is a number data type that returns a positive or negative integer.}
\]

% Difference between Cost and Sales Price

\[
\frac{(\text{TOTAL\_PRICE:SUM} - \text{QUANTITY:SUM} \times \text{Product2\_Cost__c:SUM})}{(\text{QUANTITY:SUM} \times \text{Product2\_Cost__c:SUM})} \text{ calculates the average percent difference between what a product costs and its selling price on a product-by-product level across many opportunities. Note that Product2\_Cost__c:SUM is a custom currency field named Cost on products, which includes the cost of each product. This formula is a percent data type that returns a positive or negative integer. For best results, use this on a summary Opportunities with Products report that is summarized by Product Name and includes summary totals for Quantity, Total Price, and Cost.}
\]

^ (Exponentiation)

<table>
<thead>
<tr>
<th>Description</th>
<th>Raises a number to a power of a specified number.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>number(^\text{integer}) and replace number with a merge field, expression, or another numeric value; replace integer with a merge field that contains an integer, expression, or any integer.</td>
</tr>
<tr>
<td>Example:</td>
<td>NumberOfEmployees(^4) calculates the number of employees to the 4th power.</td>
</tr>
<tr>
<td>Report Example:</td>
<td>ACTIVE:SUM (^2) calculates the number of active Salesforce.com users to the 2nd power for administration. This formula is a number data type that returns a positive integer.</td>
</tr>
<tr>
<td>Tips:</td>
<td>Avoid replacing integer with a negative number.</td>
</tr>
</tbody>
</table>

() (Open Parenthesis and Close Parenthesis)

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies that the expressions within the open parenthesis and close parenthesis are evaluated</th>
</tr>
</thead>
</table>
first. All other expressions are evaluated using standard operator precedence.

Use:

(expression1) expression2... and replace each expression with merge fields, expressions, or other numeric values.

Example:

(expression1) expression2... and replace each expression with merge fields, expressions, or other numeric values.

Report Example:

(expression1) expression2... and replace each expression with merge fields, expressions, or other numeric values.

= and == (Equal)

Description: Evaluates if two values are equivalent.

Use:

(expression1) expression2... and replace each expression with merge fields, expressions, or other numeric values.

Example:

(expression1) expression2... and replace each expression with merge fields, expressions, or other numeric values.

Due Date

Due Date = CreatedDate + 5 assigns a due date that is five days past the create date.

Commission Amount

IF(Probability =1, ROUND(Amount*0.02, 2), 0)

This formula calculates the 2% commission amount of an opportunity that has a probability of 100%. All other opportunities will have a commission value of 0.

Possible results:

- An opportunity with a Probability of 90% will have a commission of 0.
- An opportunity with a Probability of 100% and an Amount of $100,000 will have a commission of $2,000.
### <> and != (Not Equal)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if two values are not equivalent.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>expression1 &lt;&gt; expression2 or expression1 != expression2, and replace each expression with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example:</td>
<td>IF(Maint_Amount__c + Services_Amount__c &lt;&gt; Amount, &quot;DISCOUNTED&quot;, &quot;FULL PRICE&quot;)</td>
</tr>
</tbody>
</table>

This formula displays “DISCOUNTED” on an opportunity if its maintenance amount and services amount do not equal the opportunity amount. Otherwise, displays “FULL PRICE.” Note that this example uses two custom currency fields for Maint Amount and Services Amount.

### < (Less Than)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if a value is less than the value that follows this symbol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>value1 &lt; value2 and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example:</td>
<td>IF(AnnualRevenue &lt; 1000000, 1, 2) assigns the value “1” to accounts with revenues less than one million and the value “2” to accounts with revenues greater than one million.</td>
</tr>
</tbody>
</table>

### > (Greater Than)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if a value is greater than the value that follows this symbol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>value1 &gt; value2 and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example:</td>
<td>IF(commission__c &gt; 1000000, &quot;High Net Worth&quot;, &quot;General&quot;) assigns the “High Net Worth” value to opportunities with a commission greater than one million. Note, this is a text formula field on opportunities that uses a commission custom field.</td>
</tr>
</tbody>
</table>
### <= (Less Than or Equal)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if a value is less than or equal to the value that follows this symbol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>value1 &lt;= value2</code> and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>IF(AnnualRevenue &lt;= 1000000, 1, 2)</code> assigns the value “1” to accounts with revenues less than or equal to one million and the value “2” to accounts with revenues greater than one million.</td>
</tr>
</tbody>
</table>

### >= (Greater Than or Equal)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if a value is greater than or equal to the value that follows this symbol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>value1 &gt;= value2</code> and replace each value with merge fields, expressions, or other numeric values.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>IF(Commission__c &gt;= 1000000, &quot;YES&quot;, &quot;NO&quot;)</code> assigns the “YES” value to opportunities with a commission greater than or equal to one million. Note, this is a text formula field on opportunities that uses a custom currency field called Commission.</td>
</tr>
</tbody>
</table>

### && (AND)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>(logical1) &amp;&amp; (logical2)</code> and replace <code>logical1</code> and <code>logical2</code> with the values or expressions that you want evaluated.</td>
</tr>
</tbody>
</table>
| Example:    | `IF((Price<100 && Quantity<5),"Small", null)`  
This formula displays “Small” if the price is less than 100 and quantity is less than five. Otherwise, this field is blank. |
### || (OR)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>(logical1)</td>
</tr>
<tr>
<td>Example:</td>
<td>IF((ISPICKVAL(Priority, &quot;High&quot;))</td>
</tr>
<tr>
<td></td>
<td>This formula returns the number of days a case has been open if the Status is new or the Priority is high. If the case was opened today, this field displays a zero.</td>
</tr>
<tr>
<td>Validation Rule Example:</td>
<td>(Discount_Rate__c &lt; 0)</td>
</tr>
<tr>
<td></td>
<td>This validation rule formula displays the following error message when the Discount Rate custom field is not between 0 and 40%: &quot;Discount Rate cannot exceed 40%.&quot;</td>
</tr>
</tbody>
</table>

### & (Concatenate)

<table>
<thead>
<tr>
<th>Description:</th>
<th>Connects two or more strings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>string1&amp;string2 and replace each string with merge fields, expressions, or other values.</td>
</tr>
</tbody>
</table>
| Example:    | "Expense-" & Trip_Name__c & 
"-" & ExpenseNum__c |
|             | This formula displays the text “Expense-” followed by trip name and the expense number. This is a text formula field that uses an expense number custom field. |

### ABS

<table>
<thead>
<tr>
<th>Description:</th>
<th>Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>ABS(number) and replace number with a merge field, expression, or other numeric value that has the sign you want removed.</td>
</tr>
<tr>
<td>Example:</td>
<td>ABS(ExpectedRevenue) calculates the positive value of the Expected Revenue</td>
</tr>
</tbody>
</table>

Useful Formula Fields
amount regardless of whether it is positive or negative.

**AND**

| **Description:** | Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false. Use this function as an alternative to the operator && (AND). |
| **Use:** | AND(logical1,logical2,...) and replace logical1,logical2,... with the values that you want evaluated. |
| **Formula Field Example:** | IF(AND(Price<1,Quantity<1),"Small",null) This formula displays “Small” if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one. |

**BEGINS**

| **Description:** | Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not. |
| **Use:** | BEGINS(text, compare_text) and replace text, compare_text with the characters or fields you want to compare. |
| **Example:** | IF(BEGINS (Product_type__c, "ICU"), "Medical", "Technical") This example returns the text “Medical” if the text in any Product Type custom text field begins with “ICU.” For all other products, it displays “Technical.” |
| **Tips:** | • This function is case sensitive so be sure your compare_text value has the correct capitalization.  
• When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset begins with “3,” all assets that have a blank serial number are considered valid. |
**BLANKVALUE**

**Description:** Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.

**Use:**

\[
\text{BLANKVALUE(expression, substitute_expression)}
\]

and replace `expression` with the expression you want evaluated; replace `substitute_expression` with the value you want to replace any blank values.

**Example:**

**Example 1**

\[
\text{BLANKVALUE(Department, "Undesignated")}
\]

This formula returns the value of the Department field if the Department field contains a value. If the Department field is empty, this formula returns the word Undesignated.

**Example 2**

\[
\text{(BLANKVALUE(Payment Due Date__c, StartDate +5)}
\]

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field on contracts.

**Tips:**

- Use `BLANKVALUE` instead of `NULLVALUE` in new formulas. `BLANKVALUE` has the same functionality as `NULLVALUE`, but also supports text fields. Salesforce.com will continue to support `NULLVALUE`, so you do not need to change existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the `BLANKVALUE` function to return a specified string if the field does not have a value; use the `ISBLANK` function if you only want to check if the field has a value.
- If you use this function with a numeric field, the function only returns the specified string if the field does not have a value and is not configured to treat blank fields as zeroes.

---

**BR**

**Description:** Inserts a line break in a string of text.

**Use:**

\[
\text{BR()}
\]
**Example:**

```plaintext
CASE(ShippingCountry, "USA",
    ShippingStreet & BR() &
    ShippingCity & ",
    " & ShippingState & " " &
    ShippingPostalCode & BR()
& ShippingCountry,
    "France",
    ShippingStreet & BR() &
    ShippingPostalCode & " " &
    ShippingCity & BR() &
    ShippingCountry, "etc"
)
```

This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country for the account.

**Tips:**

- Do not remove the parentheses after the function name.
- Keep the parentheses empty. They do not need to contain a value.
- Remember to surround the BR() with concatenation operators: &.
- Avoid using this function in mail merge templates.
- This function is not available in s-controls.

---

**CASE**

**Description:** Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the `else_result`.

**Use:**

```plaintext
CASE(expression, value1, result1, value2, result2, ... , else_result)
```

and replace `expression` with the field or value you want compared to each specified value. Replace each value and result with the value that must be equivalent to return the result entry. Replace `else_result` with the value you want returned when the expression does not equal any values.
**Formula Field Example:**

**Days Open for Cases**

Use this example of a custom formula field called `Days Open` to display different text depending on the number of days a case has been open:

```plaintext
CASE(Days_Open__c, 3, "Reassign", 2, "Assign Task", "Maintain")
```

The following text is displayed:

- “Reassign” for any case open three days.
- “Assign Task” for any case open two days.
- “Maintain” for all other cases.

**Last Activity Month**

This formula field displays the month of the last account activity or “None” if there are no activities for the account.

```plaintext
```

**Default Value Example:**

**Discount Rate**

Use the following default value formula to insert a different discount rate on an opportunity based on the department of the person creating the opportunity:

```plaintext
CASE(User.Department, "IT", 0.25, "Field", 0.15, 0)
```

In this example, the formula inserts a discount rate of 25% on any opportunity created by a user in the “IT” department or 15% on any opportunity created by someone in the “Field” department. A zero is applied if the creator does not belong to either of these departments. This is a custom percent field on opportunities that uses the standard user field `Department`.

**Product Language**
You may want to associate a product with its language so that your users know the type of documentation or adapter to include. Use the following default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is “Japanese” if the user’s country is “Japan” and “English” if the user’s country is “US.” If neither is true, the default value “unknown” is inserted into the Product Language field.

```
```

**Tips:**

- Be sure your `value1, value2...` expressions are the same data type.
- Be sure your `result1, result2...` expressions are the same data type.
- CASE functions cannot contain functions that return true or false. Instead, make true or false expressions return numbers such as:

```
CASE(1, IF(ISPICKVAL (Term__c, "12"), 1, 0),
    12 * Monthly_Commit__c,
    IF(ISPICKVAL(Term__c, "24"), 1, 0),
    24 * Monthly_Commit__c, 0)
```

In this formula, `Term` is a picklist field that is multiplied by the Monthly Commit whenever it contains the value 1 for true.

- The `else_result` value is required.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one should be returned. For example,

```
CASE(Field__c,"Partner", "P", "Customer", "C", LEFT(Field__c, -5))
```

returns an error even if the value of the field is “Partner” or “Customer” because the last statement is illogical.

- If the field in your CASE function is blank, it returns your `else_result` value. For example, this formula:

```
CASE(Days_Open__c, 3, "Reassign", 2, "Assign Task", "Maintain")
```

displays “Maintain” if the `Days Open` field is blank, 0, or any value other than 2 or 3.
- Use CASE functions to determine if a picklist value is equal to a particular value. For example, the formula `CASE(Term__c, "12", 12 * Monthly_Commit__c, "24", 24 * Monthly_Commit__c, 0)` multiplies the Monthly Commit amount by 12 whenever the Term is 12 or multiplies the Monthly Commit amount by 24 whenever the Term is 24. Otherwise, the result is zero.

**CEILING**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Rounds a number up to the nearest integer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>CEILING(number)</code> and replace <code>number</code> with the field or expression you want rounded.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Rounding Up (literal value)</strong></td>
</tr>
<tr>
<td></td>
<td><code>CEILING(2.5)</code></td>
</tr>
<tr>
<td></td>
<td>This formula returns 3, which is 2.5 rounded up to the nearest number.</td>
</tr>
</tbody>
</table>

**Earthquake Magnitude**

`CEILING(Magnitude__c)` returns the value of a formula number field that calculates the magnitude of an earthquake up to the nearest integer.

**CONTAINS**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>CONTAINS(text, compare_text)</code> and replace <code>text</code> with the text that contains the value of <code>compare_text</code>.</td>
</tr>
</tbody>
</table>


Example:

```
IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")
```

This formula checks the content of a custom text field named `Product_Type` and returns “Parts” for any product with the word “part” in it. Otherwise, it returns “Service.” Note that the values are case sensitive, so if a `Product_Type` field contains the text “Part” or “PART,” this formula returns “Services.”

Tips:

- This function is case sensitive so be sure your `compare_text` value has the correct capitalization.
- When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset contains “A,” all assets that have a blank serial number are considered valid.
- The CONTAINS function does not support multi-select picklists. Use INCLUDES to see if a multi-select picklist has a specific value.

DATE

Description:

Returns a date value from year, month, and day values you enter. Salesforce.com displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.

Use:

```
DATE(year, month, day)
```

and replace `year` with a four-digit year, `month` with a two-digit month, and `day` with a two-digit day.

Example:

```
DATE(2005, 01, 02)
```

creates a date field of January 2, 2005.

DATEVALUE

Description:

Returns a date value for a date/time or text expression.

Use:

```
DATEVALUE(expression)
```

and replace `expression` with a date/time or text value, merge field, or expression.
Example:  

**Closed Date**

`DATEVALUE(ClosedDate)` displays a date field based on the value of the Date/Time Closed field.

**Literal Date Value**

`DATEVALUE("2005-11-15")` returns November 15, 2005 as a date value.

**Tips:**

- If the field referenced in the function is not a valid text or date/time field, the formula field displays #ERROR!
- When entering a date as a literal value, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day.
- If the expression does not match valid date ranges, such as the MM is not between 01 and 12, the formula field displays #ERROR!
- Dates and times are always calculated using the user’s time zone.

---

**DAY**

**Description:**

Returns a day of the month in the form of a number between 1 and 31.

**Use:**

`DAY(date)` and replace `date` with a date field or value such as `TODAY()`.

**Example:**

`DAY(Code_Freeze__c)` returns the day in your custom code freeze date. Note this does not work on date/time fields.

---

**EXP**

**Description:**

Returns a value for e raised to the power of a number you specify.

**Use:**

`EXP(number)` and replace `number` with a number field or value such as 5.

**Example:**

**Exponent of a Literal Value**

`EXP(3)`

This formula returns the value of e to the third power.

**Compound Interest**
Principal__c * EXP(Rate__c * Years__c)

This formula calculates the compound interest based on a custom currency field for principal, custom percent field for rate, and custom number field for years.

**FIND**

**Description:**

Returns the position of a string within a string of text represented as a number.

**Use:**

FIND(search_text, text[, start_num]) and replace search_text with the string you want to find, replace text with the field or expression you want to search, and replace start_num with the number of the character from which to start searching from left to right.

**Example:**

**Street Address**

FIND(" ", Street) returns the character position of the first space in the Street field. You can use this number to find out the length of the street address as a means of separating a street address from street name in an address field.

**Deriving Website Addresses**

SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.") finds the location of the @ sign in a person’s email address to determine the length of text to replace with a “www.” as a means of deriving their website address.

**Tips:**

- Be sure to remove the brackets, [ and ], from your formula before validating it.
- If the field referenced in your text parameter is blank, the formula field displays 0.
- Your search_text parameter is case sensitive and cannot contain any wildcard characters.
- If your search does not return any results, a 0 displays in the field.
- The start_num parameter is optional. If you do not enter a start_num value, the formula uses the value one, or the first character in the string.
- If your `start_num` is not greater than zero, a 0 displays in the field.
- If your `start_num` is greater than the length of the text, a 0 displays in the field.
- When entering your `start_num` parameter, remember that some fields like the `Website` field are unique because a “http://” is automatically appended to the beginning of the text you enter.
- Note that the first character in a string is designated as one rather than zero.

### FLOOR

**Description:**

Returns a number rounded down to the nearest integer.

**Use:**

```plaintext
FLOOR(number) and replace `number` with a number field or value such as 5.245.
```

**Example:**

**Commission Amounts**

FLOOR(commission__c) rounds commission down to the nearest integer.

**Contact's Age**

```
FLOOR((TODAY() - Birthdate) / 365.2425)
```

Use this formula to calculate a person's age based on a standard field called `Birthdate`. The person's `Birthdate` is subtracted from today's date, which returns the number of days since the person's `Birthdate`. This number is divided by the number of days in a year and rounded down to the nearest integer.

### GETRECORDIDS

**Description:**

Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.

**Use:**

```plaintext
={!GETRECORDIDS(object_type)} and replace `object_type` with a reference to the custom or standard object for the records you want to retrieve.
```
Custom Button Example:

```javascript
var records = {!GETRECORDIDS($ObjectType.Case)};
var newRecords = [];
if (records[0] == null) {
    alert("Please select at least one row")
} else {
    for (var n=0; n<records.length; n++) {
        var c = new sforce.SObject("Case");
        c.id = records[n];
        c.Status = "New";
        newRecords.push(c);
    }
    result = sforce.connection.update(newRecords);
    window.location.reload();
}
```

In this example, all selected case records are updated with a Status of “New.” To set this up in your organization, create a custom list button for cases with the following attributes:

- Display Type is “List Button”
- Behavior is “Execute JavaScript”
- Content Source is “OnClick JavaScript”

Paste the sample code above into the content of your custom button. Finally, add the list button to the a page layout that contains the Cases related list, such as accounts or opportunities. Users can select any number of cases in the related list and click the list button to change the status of those cases at once. Notice the check for `records[0] == null`, which displays a message to users when they do not select at least one record in the list.

Tips:

- Use global variables to access special merge fields for s-controls, custom buttons, and links.
- Activities are special types of objects. Use `{!GETRECORDIDS($ObjectType.Task)}` when creating a task list button. Use
when creating an event list button.
- This function is only available in custom buttons, links, and s-controls.

### GETSESSIONID

| Description: | Returns the user’s session ID. |
| Use: | GETSESSIONID() |
| Example: | HYPERLINK("https://www.myintegration.com?sId="& GETSESSIONID() & "?&rowID="&Name & "action=CreateTask","Create a Meeting Request") creates a link to an application outside of Salesforce.com, passing the parameters so that it can connect to Salesforce.com via the API and create the necessary event. |

### HTMLENCODE

| Description: | Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (>), with HTML entity equivalents, such as &gt;. |
| Use: | {%HTMLENCODE(text)%} and replace text with the merge field or text string that contains the reserved characters. |
| Example: | If the merge field foo__c contains <B>Enter the user’s name</b>, {%HTMLENCODE(foo__c)%} results in: &lt;B&gt;Enter the user’s name&lt;/b&gt; |

### HYPERLINK

| Description: | Creates a link to a URL specified that is linkable from the text specified. |
| Use: | HYPERLINK(url, friendly_name [, target]) and replace url with the Web address, replace friendly_name with the link text, and, optionally, replace target with the window or frame in which to display the content. |
### Creating Events

 HYPERLINK("00U/e?retURL=%20006&0000001180&what_id=崆 & Id, "Create Event") adds a link called “Create Event” that, when clicked, creates a new event that is associated with the current opportunity.

### Phone Dialer

 HYPERLINK("http://servername/call?id=崆 & Id & &phone=" & Phone, Phone) creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace "servername" and "call" with the name of your dialing tool and the command it uses to dial. The merge field, Id, inserts the identifier for the contact, lead, or account record. The first Phone merge field tells the dialing tool what number to call and the last Phone merge field uses the value of the Phone field as the linkable text the user clicks to dial.

### Tips:

- Hyperlink formula fields are of type text.
- Include the protocol and URL in quotes as in
  
- Avoid using text functions such as LEN, LEFT, or RIGHT on HYPERLINK function results.
- When linking to Salesforce.com pages, use a relative link, such as "00U/e?retURL=%...", for hyperlink formulas unless you want to add the formula field to a search layout. Use the complete URL, including the server name and https://, in a hyperlink formula to add it to a search layout. Note that formula fields are not available in search result layouts.
- Use the $Api variable to reference API URLs.
- Be sure to remove the brackets, [ and ], from your formula before validating it.
- The target parameter is optional. If you do not specify a target, the link opens in a new browser window. Some common target parameters are:

  **_blank**
  
  Displays link in a new unnamed window.
Displays link in the same frame or window as the element that refers to it.

Displays link in the immediate frameset parent of the current frame. This value is the same as _self if the current frame has no parent.

Displays link in the full original window, cancelling any other frames. This value is the same as _self if the current frame has no parent.

For more information on basic HTML tags, consult an HTML reference on the Internet.

- The HYPERLINK function is available everywhere that you can define a formula except default values, field updates, s-controls, validation rules, approval processes, custom buttons and links, and workflow rules.

**IF**

**Description:** Determines if expressions are true or false. Returns a given value if true and another value if false.

**Use:**

\[
\text{IF(} \text{logical\_test}, \text{value\_if\_true}, \text{value\_if\_false} \text{)} \quad \text{and replace} \quad \text{logical\_test} \quad \text{with the expression you want evaluated; replace} \quad \text{value\_if\_true} \quad \text{with the value you want returned if the expression is true; replace} \quad \text{value\_if\_false} \quad \text{with the value you want returned if the expression is false.}
\]

**Formula Field Example:**

**Overdue Payments**

\[
\text{IF(AND(Payment\_Due\_Date\_c < TODAY(), Payment\_Status\_c ="UNPAID"), "PAYMENT OVERDUE", null)}
\]

This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, returns the text “PAYMENT OVERDUE” and if not, leaves the field blank. This example uses a custom date field called Payment Due Date and a text custom field called Payment Status on contracts.
### Insert Tax Rate

Use this default value formula to set the tax rate of an asset based on the user’s city. Create a custom percent field with the following default value:

```
IF($User.City = "Napa", 0.0750, 
    IF($User.City = "Paso Robles", 
        0.0725, 
        IF($User.City = "Sutter Creek", 
            0.0725, 
            IF($User.City = "Los Olivos", 
                0.0750, 
                IF($User.City = "Livermore", 
                    0.0875, null 
                ) 
            ) 
        ) 
    ) 
) 
```

### Custom Button Example:

```
{!
        "http://maps.google.com/maps?q=
            
```

This example uses the IF function to determine if an address is in the United States or United Kingdom so that it can use the appropriate type of Google map to display the address.

### Tips:

- Make sure your `value_if_true` and `value_if_false` expressions are the same data type.
- When using an IF function with the `$Profile.UserType` variable to determine the type of Salesforce.com user license the logged in user has, use the following values:
  - Standard for Salesforce
  - PowerPartner for PRM User
  - CustomerSuccess for Customer Portal User
  - PowerCustomerSuccess for Customer Portal Manager
For example, use the following formulas to determine if the logged in user has the license type in quotes:

\[
\text{IF(ISPICKVAL($Profile.UserType, "Standard"), 100, 0.1)}
\]

\[
\text{IF(ISPICKVAL($Profile.UserType, "PowerPartner"), 100, 0.1)}
\]

\[
\text{IF(ISPICKVAL($Profile.UserType, "CustomerSuccess"), 100, 0.1)}
\]

**Note:** $Profile merge fields are only available in Enterprise, Unlimited, and Developer Editions.

### IMAGE

**Description:** Inserts an image with alternate text and height/width specifications.

**Use:**

\[
\text{IMAGE(image_url, alternate_text, height, width)}
\]

and replace `image_url` with the full path to the image; replace `alternate_text` with the string of text you want displayed when you hover your mouse over the image; replace `height` with the vertical size of the image in pixels; replace `width` with the horizontal size of the image in pixels.

**Example:**

\[
\text{HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,}
\]

\[
\text{IMAGE("http://opi.yahoo.com/online?ue= & Yahoo_Name__c & "&m=g&n=0", "Yahoo"))}
\]

This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called `Yahoo Name` on contacts where you can store the contact’s Yahoo! Messenger ID.

**Tips:**

- The `height` and `width` parameters are optional.
- Use a text string to replace the `image_url` and `alternate_text` parameters. Surround each text string in quotes.
• Use numbers to replace the `height` and `width` parameters.
• Add images to your Documents tab if you want to display them elsewhere. For example, store the image of a product in a document folder, copy the URL to the document, and paste that URL in the `image_url` parameter of a formula field on the Products tab.
• If you use Internet Explorer, you may need to change your security settings so that it does not display a warning prompt when images use HTTP protocol. See the online help for Internet Explorer for instructions on changing your security settings.
• The IMAGE function cannot include the `GETSESSIONID` function as one of its arguments.
• The IMAGE function is available everywhere that you can define a formula except default values, field updates, s-controls, validation rules, approval processes, and workflow rules.

**INCLUDE**

**Description:**
Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.

**Use:**
```{}
{{INCLUDE(source, [inputs])}}
```
replace `source` with the s-control snippet you want to reference. Replace `inputs` with any information you need to pass to the snippet.

**S-Control Example:**

**Including Header Snippet**

```html
<html>
<body>
{{INCLUDE($SControl.Header_Snippet, [title = "My Title", theme = "modern"])}}
</body>
</html>
```

This example references a snippet that provides a header for a page that you created to display in a Web tab. It displays the page title "My Title." Use the `$SControl` global variable to reference a custom s-control.

**Including Input Parameters**
Use the following two examples to see how you can create a reusable snippet and include it in an s-control.

```html
<h2 class="{$Request.titleTheme}.title"> {$Request.titleText}</h2>
```

This snippet requires two input parameters: `titleTheme` and `titleText`. It is a reusable HTML tag that presents a page title and theme based on input parameters. Next, create an s-control that includes this snippet:

```html
<html>
<head>
</head>
<body>
{! INCLUDE($SControl.Title_Snippet, [titleTheme = "modern", titleText = "My Sample Title"]) }

... Insert your page specific content here ...

</body>
</html>
```

This s-control uses the snippet titled `Title_Snippet` to display the title of the page “My Sample Title” and modern theme. Replace `Insert your page specific content here` with your own HTML content and use the s-control as the source of a Web tab to create your own pages in Salesforce.com.

**Tips:**

- Because this function references an s-control snippet and does not copy it, it always runs the latest content of the s-control snippet. Remember when making a change to your s-control snippet that it affects all `INCLUDE` functions that refer to it.
- Use the `$Request` global variable to access any information inside the snippet.
- This function is only available in custom buttons, links, and s-controls.

**INCLUDES**

| Description: | Determines if any value selected in a multi-select picklist field equals a text literal you specify. |
Use:

| INCLUDES( | multiselect_picklist_field, text_literal) and replace multiselect_picklist_field with the merge field name for the multi-select picklist; and replace text_literal with the multi-select picklist value you want to match in quotes. |

Examples:

| INCLUDES(Hobbies__c, "Golf") returns TRUE if one of the selected values in the Hobbies custom multi-select picklist field is Golf. |

Tips:

- The text_literal expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.
- Salesforce.com returns an error if any of the following occurs:
  - You do not provide a text_literal expression.
  - You provide an empty text_literal expression, such as "" or " ".
- Use ISNULL to determine if a multi-select picklist field is empty.
- Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example:

```plaintext
INCLUDES(PRIORVALUE(multiselect_picklist_field), text_literal)
```

ISBLANK

| Description: | Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE. |
| Use: | ISBLANK(expression) and replace expression with the expression you want evaluated. |
Example:

(IF(ISBLANK(Maint_Amount__c), 0, 1) +
  IF(ISBLANK(Services_Amount__c),
    0,1) +
  IF(ISBLANK(Discount_Percent__c),
    0, 1) +
  IF(ISBLANK(Amount),
    0, 1) +
  IF(ISBLANK(Timeline__c), 0,
    1)) / 5

This formula takes a group of opportunity fields and calculates what percent of them are being used by your sales personnel. This formula checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the Treat blank fields as blanks option under Blank Field Handling while the Advanced Formula subtab is showing.

Tips:

- Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce.com will continue to support ISNULL, so you do not need to change any existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the BLANKVALUE function to return a specified string if the field does not have a value; use the ISBLANK function if you only want to check if the field has a value.
- If you use this function with a numeric field, the function only returns TRUE if the field has no value and is not configured to treat blank fields as zeroes.

ISCHANGED

Description: Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.

Use: ISCHANGED(field) and replace field with the name of the field you want to compare.
The following validation rule prevents users from changing an opportunity name after it has been created:

\[ \text{NOT(ISCHANGED(Name))} \]

\[ \text{NOT(AND(ISCHANGED(Priority), ISPICKVAL(Priority, "Low")))} \]

is a validation rule that ensures if a user changes the Priority of a case, the new priority cannot be "Low."

\[ \text{NOT(AND(ISCHANGED(CloseDate), OR(MONTH(CloseDate) \<> MONTH(TODAY()), YEAR(CloseDate) \<> YEAR(TODAY()),$Profile.Name \<> "Sales Manager"))} \]

is a validation rule that prevents a user from changing the Close Date of an opportunity to a date outside of the current month and year unless that user has the “Sales Manager” profile.

**Note:** $Profile merge fields are only available in Enterprise, Unlimited, and Developer Editions.

**Tips:**

- This function is available only in:
  - Assignment rules
  - Validation rules
  - Field updates
  - Workflow rules if the trigger type is set to Every time a record is created or edited.

- Use the NOT function to reverse the return values of TRUE and FALSE.

- This function returns FALSE when evaluating any field on a newly created record.

- If a text field was previously blank, this function returns TRUE when it contains any value.

- For number, percent, or currency fields, this function returns TRUE when:
  - The field was blank and now contains any value
  - The field was zero and now is blank
  - The field was zero and now contains any other value
### ISNEW

<table>
<thead>
<tr>
<th><strong>Description:</strong></th>
<th>Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use:</strong></td>
<td>ISNEW()</td>
</tr>
<tr>
<td><strong>Validation Rule Example:</strong></td>
<td>Use the following validation rule to prevent users from creating opportunities with a close date in the past. AND (ISNEW(), CloseDate &lt; TODAY()) checks if the user is creating a new opportunity and, if so, ensures that the Close Date is today or after today. Use this validation rule to ensure users add at least one product to an opportunity after they have created it. NOT(OR(ISNEW(),HasOpportunityLineItem)) In this example, the validation rule formula displays the following error message when an existing opportunity does not have any products: “You must add products to this opportunity before saving.” This does not display an error on the initial save because they cannot add products until after saving the record initially; but it prevents them from resaving or closing an opportunity that does not contain products.</td>
</tr>
<tr>
<td><strong>Tips:</strong></td>
<td>• This function is available only in validation rules, field updates, and workflow rules. • Use the NOT function to reverse the return values of TRUE and FALSE. • This function always returns FALSE when used in a workflow rule with a time-based trigger. • This function always returns FALSE when used in a field update for an approval action.</td>
</tr>
</tbody>
</table>

### ISNULL

<table>
<thead>
<tr>
<th><strong>Description:</strong></th>
<th>Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note:</strong></td>
<td>Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields.</td>
</tr>
</tbody>
</table>
Salesforce.com will continue to support ISNULL, so you do not need to change any existing formulas.

**Use:**

ISNULL(expression) and replace expression with the expression you want evaluated.

**Example:**

```plaintext
(IF(ISNULL(Maint_Amount__c), 0, 1) +
 IF(ISNULL(Services_Amount__c), 0, 1) +
 IF(ISNULL(Discount_Percent__c), 0, 1) +
 IF(ISNULL(Amount), 0, 1) +
 IF(ISNULL(Timeline__c), 0, 1)) / 5
```

This formula takes a group of opportunity fields and calculates what percent of them are being used by your sales personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the Treat blank fields as blanks option under Blank Field Handling while the Advanced Formula subtab is showing.

**Validation Rule Example:**

```plaintext
AND(ISPICKVAL(StageName, "Closed Won"),
 ISNULL(Project_Start_Date__c))
```

This validation rule makes the Project Start Date custom date field conditionally required whenever the opportunity stage is “Closed Won.”

**Tips:**

- Text fields are never null, so using this function with a text field always returns false. For example, the formula field IF(ISNULL(new__c) 1, 0) is always zero regardless of the value in the New field. For text fields, use the ISBLANK function instead.
- Multi-select picklist fields are never null in s-controls, buttons, and email templates, so using this function with a multi-select picklist field in those contexts always returns false.
- Empty date and date/time fields always return true when referenced in ISNULL functions.
• Choose Treat blank fields as blanks for your formula when referencing a number, percent, or currency field in an ISNULL function. Choosing Treat blank fields as zeroes gives blank fields the value of zero so none of them will be null.

• Merge fields can be handled as blanks, which can affect the results of components like s-controls because they can call this function.

• When using a validation rule to ensure that a number field contains a specific value, use the ISNULL function to include fields that do not contain any value. For example, to validate that a custom field contains a value of ‘1,’ use the following validation rule to display an error if the field is blank or any other number:

\[
\text{OR(} \text{ISNULL(field__c)}, \text{field__c}<>1)\]

**ISNUMBER**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>ISNUMBER(text) and replace text with the merge field name for the text field.</td>
</tr>
<tr>
<td>Validation Rule Example:</td>
<td>OR(LEN(Bank_Account_Number__c) &lt;&gt; 10, NOT(ISNUMBER(Bank_Account_Number__c)))</td>
</tr>
<tr>
<td>This validation rule ensures a custom text field called Bank Account Number is a number of 10 digits and is not blank.</td>
<td></td>
</tr>
<tr>
<td>Tips:</td>
<td>• This function returns FALSE for blank values.</td>
</tr>
<tr>
<td></td>
<td>• The ISNUMBER function is not aware of your locale. For example, ISNUMBER(&quot;123,12&quot;) and ISNUMBER(&quot;1 000&quot;) return FALSE even if the user’s locale is “French.”</td>
</tr>
<tr>
<td></td>
<td>• Chinese, Japanese, Korean, and special characters including a space return FALSE.</td>
</tr>
<tr>
<td></td>
<td>• The ISNUMBER function returns TRUE for scientific formatting such as “2E2” or “123.123.”</td>
</tr>
</tbody>
</table>
**ISPICKVAL**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Determines if the value of a picklist field is equal to a text literal you specify.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>ISPICKVAL(picklist_field, text_literal)</code> and replace <code>picklist_field</code> with the merge field name for the picklist; replace <code>text_literal</code> with the picklist value in quotes. <code>text_literal</code> cannot be a merge field or the result of a function.</td>
</tr>
</tbody>
</table>
| Examples:   | **Contract Activation**<br>IF(ISPICKVAL(Status, "Activated"), NOW()-ActivatedDate, null) calculates the number of days since the contract was activated. If the contract status is not "Activated," this field is blank.  
**Commission Amounts**<br>IF(ISPICKVAL(StageName, "Closed Won"), ROUND(Amount *0.02, 2), 0)<br>This example calculates the commission amount for any opportunity that has a "Closed Won" stage. The value of this field will be the amount times 0.02 for any closed/won opportunity. Open or lost opportunities will have a zero commission value.  
**Competitor-Triggered Workflow**<br>ISPICKVAL(Stage, "Closed Lost") && INCLUDES(Competitor__c, "Acme")<br>This formula in a workflow rule configures Salesforce.com to trigger the associated workflow actions if the Competitor multi-select picklist field on a lost opportunity is Acme. |
| Tips:       |  
- Replace `picklist_field` with a custom or standard field of type picklist.  
- Your `text_literal` expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.  
- Use `CASE` functions to determine if a picklist value is equal to a particular value.  
- When using the ISPICKVAL function to return the previous value of a picklist field, |
include the PRIORVALUE function inside the ISPICKVAL function as in this example:

\[
\text{ISPICKVAL(}\text{PRIORVALUE(}\text{picklist_field}\text{), text_literal)}
\]

**JSENCODE**

**Description:**
Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (’).

**Use:**
\[
\text{\{!JSENCODE(text)}\}\text{ and replace text with the merge field or text string that contains the unsafe JavaScript characters.}
\]

**Example:**
If the merge field foo__c contains \text{<B>Enter the user's name</b>}, \text{\{!JSENCODE(foo__c)}\ results in:
\text{\textbackslash u003CB\textbackslash u003EEnter the user's name\textbackslash u003C/\textbackslash u003E}}

**JSINHTMLENCODE**

**Description:**
Encodes text and merge field values for use in JavaScript within HTML tags by inserting escape characters before unsafe JavaScript characters and replacing characters that are reserved in HTML with HTML entity equivalents.

**Use:**
\[
\text{\{!JSINHTMLENCODE(text)}\}\text{ and replace text with the merge field or text string that contains the unsafe JavaScript characters.}
\]

**Example:**
If the merge field foo__c contains \text{<B>Enter the user's name</b>}, \text{\{!JSINHTMLENCODE(foo__c)}\ results in:
\text{\&lt;B&gt;Enter the user's name&lt;/b&gt;}

**LEFT**

**Description:**
Returns the specified number of characters from the beginning of a text string.

**Use:**
\[
\text{LEFT(text, num_chars)}\text{ and replace text with the field or expression you want returned;}
\]
replace *num_chars* with the number of characters from the left you want returned.

**Example:**

```
TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN__c, 4))
```

This formula displays the first five characters of the contact’s last name and the last four characters of the contact’s social security number separated by a dash. Note that this example uses a text custom field called SSN on contacts.

**Tips:**

- Reference auto-number fields as text fields in formulas.
- If the *num_chars* value is less than zero, Salesforce.com replaces the value with zero.

**LEN**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the number of characters in a specified text string.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>LEN(text) and replace text with the field or expression whose length you want returned.</td>
</tr>
<tr>
<td>Example:</td>
<td>LEN(PartNumber__c)</td>
</tr>
</tbody>
</table>

This formula returns the number of characters in a Product Code field.

**LINKTO**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce.com page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>{$LINKTO(label, target, id, [inputs], [no override])} and replace label with the text for the link, target with the URL, and id with a reference to the record. Inputs are optional and can include any additional parameters you want to add to the link. The no override argument is also optional and defaults to “false.” It applies to targets for standard Salesforce.com pages such as $Action.Account.New. Replace no override with “true” when you want to display a standard Salesforce.com page regardless of whether you have defined an override for it elsewhere.</td>
</tr>
</tbody>
</table>
### S-Control Example:

```html
<html>
<body>
{!LINKTO("Create a New Account", $Action.Account.New, $ObjectType.Account)}
</body>
</html>
```

This example allows users to click a link to create a new account. It is useful in account list views or Web tabs where you want users to create an account directly from that page. Use the $Action global variable to access the new account page in Salesforce.com.

### New Email Window S-Control

```html
<html>
<body>
{!LINKTO("Email link", "mailto:support@yourcompany.com?subject=Please%20Help")};
</body>
</html>
```

This example launches a new email window addressed to support@yourcompany.com with the subject “Please Help” whenever a user clicks “Mail link.”

### Link to Another S-Control

```html
<html>
<body>
{!LINKTO("Check for duplicates", $Scontrol.dedup_account, Account.Id)}
</body>
</html>
```

Use this example to generate a page containing a hyperlink labeled “Check for duplicates.” When users click this link, Salesforce.com runs your custom s-control. This example assumes you have already created a custom s-control to find duplicate accounts and merge their information.

### Tips:

- Avoid using this function in an inline s-control if you want it to open in a new window.
• Enclose multiple *inputs* in brackets to indicate they are together:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, [parm1="A", parm2="B"])}
```

• Set *inputs* to null if you do not have any to pass yet you want to set the *no override* argument:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, null, true)}
```

• When you override the tab home page for a standard or custom tab, set *target* to the “Tab” $Action global variable and *id* to the object type. For example,

```
LINKTO("Accounts Tab", $Action.Account.Tab, $ObjectType.Account)
```

• This function is only available in custom buttons, links, and s-controls.

---

**LN**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>LN(number) and replace number with the field or expression for which you want the natural logarithm. Note: the LN function is the inverse of the EXP function.</td>
</tr>
<tr>
<td>Example:</td>
<td>LN(10) returns the natural logarithm of 10, which is 2.30.</td>
</tr>
<tr>
<td></td>
<td>LN(Value__c) returns the natural logarithm of a custom number field called Value.</td>
</tr>
</tbody>
</table>

**LOG**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the base 10 logarithm of a number.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>LOG(number) and replace number with the field or expression from which you want the base 10 logarithm calculated.</td>
</tr>
</tbody>
</table>
### Example:

<table>
<thead>
<tr>
<th>Salary</th>
<th>LOG(Salary__c) calculates the logarithm of a person's salary. In this example, Salary is a custom currency field.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hydrogen</td>
</tr>
<tr>
<td></td>
<td>-LOG(Hydrogen__c) calculates the pH and acidity using the LOG function and a custom number field called Hydrogen, which represents the concentration of Hydrogen ions in the liquid measured in moles per liter.</td>
</tr>
</tbody>
</table>

### LOWER

**Description:** Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.

**Use:** LOWER(text, [locale]) and replace text with the field or text you wish to convert to lowercase, and locale with the optional two-character ISO language code or five-character locale code, if available. For information on supported languages, see What languages does Salesforce.com support?.

**Example:**

<table>
<thead>
<tr>
<th>SALESFORCE.COM</th>
<th>LOWER(&quot;SALESFORCE.COM&quot;) returns “salesforce.com.”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticker Symbol</td>
<td>LOWER(TickerSymbol) returns the text in Ticker Symbol in lower case characters.</td>
</tr>
</tbody>
</table>

**Applying Turkish Language Locale Rules**

The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the LOWER() function with the Turkish language locale, use the Turkish locale code tr in the LOWER() function as follows:

LOWER(text, "tr")

This ensures that Salesforce.com does not transform any dotted i in the text to a dotless I.
### LPAD

**Description:** Inserts spaces or characters you specify to the left-side of a text string.

**Use:**

\[
\text{LPAD}(\text{text}, \text{padded}_\text{length}[,\ \text{pad}_\text{string}])
\]

and replace the variables:

- **text** is the field or expression you want to insert spaces or characters to the left of.
- **padded_length** is the number of total characters in the text that will be returned.
- **pad_string** is the character or characters that should be inserted. pad_string is optional and defaults to a blank space.

If the value in text is longer than pad_string, text is truncated to the size of padded_length.

**Example:**

**Account Name: Padding**

\[
\text{LPAD}(\text{Name}, 20)
\]

truncates the Name field after 20 characters or inserts spaces to the left of the name until the entire value is 20 characters long. For example, if the name is salesforce.com, the value returned is "......salesforce.com."

**My_Company: No Change**

\[
\text{LPAD('my\_company\_com', 14, 'z')}
\]

returns "my\_company\_com" without change because it has 14 characters.

**Account Name Padded with Z**

\[
\text{LPAD(\text{Name}, 15, 'z')}
\]

returns the name "zsalesforce.com."

**Account Name: Truncating**

\[
\text{LPAD(\text{Name}, 2)}
\]

truncates the name after the second character. For example, if the name is salesforce.com, the value returned is "sa."

### MAX

**Description:** Returns the highest number from a list of numbers.

**Use:**

\[
\text{MAX}(\text{number, number,...})
\]

and replace number with the fields or expressions from which you want to retrieve the highest number.
Service Charge

```
MAX(0.06 * Total_Cost__c,
   Min_Service_Charge__c)
```

In this example, the formula field calculates a service charge of 6% of the total cost or a minimum service charge, whichever is greater. Note that Min Service Charge is a custom currency field with a default value of $15. However, you could make it a formula field if your minimum service charge is always the same amount.

Book Royalties

```
MAX(0.10 * Pages__c,
    (Retail_Price__c * 0.07) * 
    Total_Sold__c)
```

This formula determines which amount to pay in royalties for a book. It displays the greater of two amounts: $0.07 for each book sold or $0.10 per page. It assumes you have custom number fields for Pages and Total Sold and a custom currency field for Retail Price.

Commissions

```
MAX($User.Commission_Percent__c * 
    Price,
    Price * Account_Discount__c, 
    100)
```

This formula determines what commission to log for an asset based on which is greater: the user’s commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

MID

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the specified number of characters from the middle of a text string given the starting position.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>MID(text, start_num, num_chars) and replace text with the field or expression to use when returning characters; replace start_num with the number of characters from the left to use as a starting position; replace num_chars with the total number of characters to return.</td>
</tr>
</tbody>
</table>
Example: MID(Division, 3, 4) returns four characters of the Division name beginning with the third character from the left. On a user record, this represents the department code.

MIN

| Description: | Returns the lowest number from a list of numbers. |
| Use: | MIN(number, number,...) and replace number with the fields or expressions from which you want to retrieve the lowest number. |
| Example: | 401K Matching |
| | MIN(250, Contribution__c /2) |
| | This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or $250, whichever is less. It assumes you have custom currency field for Contribution. |
| Bonus | MIN(Gross__c * Bonus_Percent__c, Performance__c / Number_of_Employees__c) |
| | This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have custom number field for Number of Employees, a custom percent field for Bonus Percent, and currency custom fields for the employee's Gross and company's Performance. |

MOD

| Description: | Returns a remainder after a number is divided by a specified divisor. |
| Use: | MOD(number, divisor) and replace number with the field or expression you want divided; replace divisor with the number to use as the divisor. |
### Example:

```plaintext
MOD(3, 3) returns 0
MOD(4, 3) returns 1
MOD(123, 100) returns 23
```

You may want to prevent users from scheduling meetings on a Saturday or Sunday. Use the following example to apply a validation rule to a custom date field called `My Date`.

```plaintext
CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7),
     0, 0,
     6, 0,
     1) = 0
```

This example displays the following error message when the value of `My Date` is not Monday through Friday: “My Date is not a weekday.”

### MONTH

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>MONTH(date)</code> and replace <code>date</code> with the field or expression for the date containing the month you want returned.</td>
</tr>
</tbody>
</table>
| Example:     | **SLA Expiration**

```plaintext
MONTH(SLAExpirationDate__c) returns the month that your service-level agreement expires. This example uses a custom date field called `SLA Expiration Date`. |
```

**Current Month**

```plaintext
MONTH(TODAY()) returns the current month in a number format. For example, the month of February would be the value “2.” |
```

### NOT

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns FALSE for TRUE and TRUE for FALSE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>NOT(logical)</code> and replace <code>logical</code> with the expression that you want evaluated.</td>
</tr>
</tbody>
</table>
| Example:     | **IF(NOT(ISPICKVAL(Status, "Closed")),

```plaintext
ROUND(NOW()-CreatedDate, 0), null checks to see if a case is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the case is not open, this field is blank. |
```
### NOW

**Description:** Returns a date/time representing the current moment.

**Use:**

```
NOW()
```

**Example:**

```
IF(ISPICKVAL(Status, "Open"),
   ROUND(NOW()-CreatedDate, 0), null)
```

This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead is not open, this field is blank.

**Tips:**
- Do not remove the parentheses.
- Keep the parentheses empty. They do not need to contain a value.
- Use a date/time field in a NOW function instead of a date field. Created Date and Last Modified Date are date/time fields whereas Last Activity Date is a date field.
- Use TODAY if you prefer to use a date field.
- Dates and times are always calculated using the user's time zone.
- Use addition and subtraction operators with a NOW function and other date/time fields to return a number, representing number of days. For example NOW() - CreatedDate calculates the number of days since the created date of a record. In this example, the formula field data type is a number.
- Use addition and subtraction operators with a NOW function and numbers to return a date and time. For example NOW() +5 calculates the date and time five days ahead of now. In this example, the formula field data type is a date/time.

### NULLVALUE

**Description:** Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression.

**Note:** Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Salesforce.com will continue to support NULLVALUE, so you do not need to change existing formulas.

**Use:**

```
NULLVALUE(expression, substitute_expression)
```

and replace **expression** with the expression you want to evaluate; replace **substitute_expression** with the value you want to replace any blank values.
**Example:**

\[
\text{(NULLVALUE(Payment\_Due\_Date\_c, StartDate +5)}
\]

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field on contracts.

**Tips:**

- Avoid using this function with text fields because they are never null even when they are blank. Instead, use the BLANKVALUE function to determine if a text field is blank.
- Choose Treat blank fields as blanks for your formula when referencing a number, percent, or currency field in a NULLVALUE function. Choosing Treat blank fields as zeroes gives blank fields the value of zero so none of them will be null.
- Use the same data type for both the expression and substitute_expression.

**OR**

**Description:**

Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false. Use this function as an alternative to the operator || (OR).

**Use:**

\[
\text{OR(logical1, logical2...)}
\]

and replace any number of logical references with the expressions you want evaluated.

**Formula Field Example:**

\[
\text{IF(OR(ISPICKVAL(Priority, "High"), ISPICKVAL(Status, "New")), ROUND(NOW()-CreatedDate, 0), null)}
\]

This formula returns the number of days a case has been open if the Status is new or the Priority is high. If the case was opened today, this field displays a zero.

**Validation Rule Example:**

\[
\text{OR(Discount\_Rate\_c < 0, Discount\_Rate\_c > 0.40)}
\]

This validation rule formula displays the following error message when the Discount Rate custom field is not between 0 and 40%: “Discount Rate cannot exceed 40%.”

**PARENTGROUPVAL**

**Description:**

A summary function that returns the value of the selected summary field from the parent grouping level that you specify. Parent grouping levels are those above the grouping level where you chose to display the formula. For matrix reports, choose both row and column grouping levels.
**PARENTGROUPVAL**

You can use this function only in the context of creating custom summary formulas for summary reports.

**Use:**

PARENTGROUPVAL(summary_field, grouping_level) and replace summary_field and grouping_level with the expressions you want evaluated.

**Example:**

<table>
<thead>
<tr>
<th>TOTAL_PRICE:SUM/PARENTGROUPVAL(TOTAL_PRICE:SUM, GRAND_SUMMARY)</th>
</tr>
</thead>
</table>

This custom summary formula calculates the size of one product compared to other products for a report on opportunities and their products, grouped by product name.

**PREVGROUPVAL**

A summary function that returns the value of the selected summary field from the previous summary row at the grouping level that you specify. For matrix reports, the summary_field you choose controls whether a previous row or column is returned. The increment determines how many steps previous to the current summary field to return. The default increment is 1. The maximum increment is 12.

You can use this function only in the context of creating custom summary formulas for summary reports.

**Use:**

PREVGROUPVAL(summary_field, grouping_level [, increment]) and replace summary_field, grouping_level, and increment with the expressions you want evaluated.
Example:

AMOUNT:SUM - PREVGROUPVAL(AMOUNT:SUM, CLOSE_DATE)

This custom summary formula calculates the change (positive or negative) for a given month compared to the previous visible month in the report. In this example, the report is an opportunity matrix grouped by opportunity close date (in the columns) and stage (in the rows).

PRIORVALUE

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the previous value of a field.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>PRIORVALUE(field)</td>
</tr>
<tr>
<td>Validation Rule Example:</td>
<td>The following validation rule prevents users from changing the expected revenue of an opportunity after it is closed: AND(PRIORVALUE(Amount) &gt; Amount, IsClosed).</td>
</tr>
<tr>
<td>Tips:</td>
<td>- This function is available only in:</td>
</tr>
<tr>
<td></td>
<td>- Assignment rules</td>
</tr>
<tr>
<td></td>
<td>- Validation rules</td>
</tr>
<tr>
<td></td>
<td>- Field updates</td>
</tr>
<tr>
<td></td>
<td>- Workflow rules if the trigger type is set to Every time a record is created or edited.</td>
</tr>
<tr>
<td></td>
<td>- This function does not return default values.</td>
</tr>
<tr>
<td></td>
<td>- When users create a new record, this function returns the value of the field referenced rather than null. For example, if you create an account named “Acme,” PRIORVALUE(Account.Name) returns Acme.</td>
</tr>
<tr>
<td></td>
<td>- When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example:</td>
</tr>
<tr>
<td></td>
<td>ISPICKVAL(PRIORVALUE(picklist_field), text_literal)</td>
</tr>
<tr>
<td></td>
<td>- Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example:</td>
</tr>
<tr>
<td></td>
<td>INCLUDES(PRIORVALUE(multiselect_picklist_field), textLiteral)</td>
</tr>
</tbody>
</table>
**REGEX**

**Description:** Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, it returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.

**Use:** `REGEX(text, regex_text)` and replace `text` with the text field, and `regex_text` with the regular expression you want to match.

**Validation Rule Example:** This example ensures that a custom field called `SSN` matches a regular expression representing a valid social security number format of the form `999-99-9999`.

```sql
NOT(
  OR(
    LEN(SSN__c) = 0,
    REGEX(SSN__c, "[0-9]{3}-[0-9]{2}-[0-9]{4}"
  )
)
)
```

**Tips:**
- Regular expression syntax is based on Java Platform SE 6 syntax. However, backslash characters (\) must be changed to double backslashes (\\) because backslash is an escape character in Salesforce.com.
- The Salesforce.com regular expression engine matches an entire string as opposed to searching for a match within a string. For example, if you are searching for the name Marc Benioff, use the regular expression, `.*Marc Benioff.*`, to find a match in a string like the following:

  According to Marc Benioff, Salesforce increases customer success.

  If you use the regular expression, `Marc Benioff`, the only string that this regular expression will match is:

  Marc Benioff

- Capture groups and substitutions are ignored.
- This function is available everywhere formulas exist except formula fields and default values.

---

**REQUIRESCRIPT**

**Description:** Returns a script tag with source for a URL you specify. Use this function when referencing the Force.com AJAX Toolkit or other JavaScript toolkits.
Use:

{{!REQUIRESCRIPT(url)}} and replace url with the link for the script that is required.

For the AJAX Toolkit:

{{!requireScript("/soap/ajax/13.0/connection.js")}}

Returns:

<script src="/soap/ajax/13.0/connection.js"></script>

For Dojo:

{{!requireScript("/js/dojo/0.3.1/dojo.js")}}

Returns:

<script src="/js/dojo/0.3.1/dojo.js"></script>

Custom Button Example:

{{!REQUIRESCRIPT("/soap/ajax/13.0/connection.js")}}

var c = new sforce.SObject("Case");
c.id = "{{!Case.Id}}";
c.Status = "New";
result = sforce.connection.update([c]);
window.location.reload();

This example sets the Status of a case to “New” whenever a user clicks a custom button from the case detail page. To set this up in your organization, define a custom button for cases that has the following attributes:

• Display Type is “Detail Page Button”
• Behavior is “Execute JavaScript”
• Content Source is “OnClick JavaScript”

Next, paste the content above into your custom button definition and add it to your case page layouts.

Tips:

• Use global variables to access special merge fields for s-controls.
• Use this function when creating custom buttons or links where you have set the Behavior to “Execute JavaScript” and Content Source to “OnClick JavaScript” because the script tag should be outside the OnClick code.
• This function is only available for custom buttons and links that have Content Source set to “OnClick JavaScript.”
• When working in Visualforce, use INCLUDESCRIPT instead.

RIGHT

| Description: | Returns the specified number of characters from the end of a text string. |
Use: \( \text{RIGHT(text, num_chars)} \) and replace text with the field or expression you want returned; replace num_chars with the number of characters from the right you want returned.

Example: \( \text{TRIM(LEFT(LastName, 5))} \& "" \& \text{TRIM(RIGHT(SSN__c, 4))} \) displays the first five characters of the contact's last name and the last four characters of the contact's social security number separated by a dash. Note that this assumes you have a text custom field called SSN on contacts.

Tips: - Reference auto-number fields as text fields in formulas.
- If the num_chars value is less than zero, Salesforce.com replaces the value with zero.

**ROUND**

Description: Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.

Use: \( \text{ROUND(number, num_digits)} \) and replace number with the field or expression you want rounded; replace num_digits with the number of decimal places you want to consider when rounding.

Example: \( \text{ROUND (1.5, 0)} = 2 \)
\( \text{ROUND (1.2345, 0)} = 1 \)
\( \text{ROUND (-1.5, 0)} = -2 \)
\( \text{ROUND (225.49823, 2)} = 255.50 \)

**Simple Discounting**

\( \text{ROUND(Amount-Amount* Discount_Percent__c,2)} \)

Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called Discount Percent.

Tips: - Enter zero for num_digits to round a number to the nearest integer.
- Salesforce.com automatically rounds numbers based on the decimal places you specify. For example, a custom number field with two decimal places stores 1.50 when you enter 1.49999.
- The decimal numbers displayed depend on the decimal places you selected when defining the field in the custom field wizard. The num_digits represents the number of digits considered when rounding.
**RPAD**

**Description:** Inserts blank spaces or characters that you specify to the right-side of a text string.

**Use:**

```
RPAD(text, padded_length[, 'pad_string'])
```

- `text` is the field or expression after which you want to insert spaces or characters.
- `padded_length` is the number of total characters in the text string that will be returned.
- `pad_string` is the character or characters that should be inserted. `pad_string` is optional and defaults to a blank space.

If the value in `text` is longer than `pad_string`, `text` is truncated to the size of `padded_length`.

**Example:**

**Account Name: Padding Default**

`RPAD(Name, 20)` truncates the `Name` field after 20 characters if the name is longer than 20 characters or inserts spaces to the right of the existing name until the name is 20 characters long. For example, if the name is `salesforce.com`, the value returned is “salesforce.com”.

**My_Company: No Change**

`RPAD('my_company.com', 14, 'z')` returns “my_company.com” without change because it has 14 characters.

**Account Name: Padding with a Character**

`RPAD(Name, 15, 'z')` returns “salesforce.comz”.

**Account Name: Truncating**

`RPAD(Name, 2)` truncates the name after the second character. For example, if the name is `salesforce.com`, the value returned is “sa.”

---

**SQRT**

**Description:** Returns the positive square root of a given number.

**Use:**

```
SQRT(number)
```

- `number` is the field or expression you want computed into a square root.

**Example:**

`SQRT(25)` returns the square root of 25, which is 5.

**Amplitude**

`SQRT(Amplitude__c)` returns the square root of a custom number field representing the amplitude of an earthquake.

**Tips:**

- Calculating the square root of a negative number results in an error on the detail page.
- Avoid division by zero errors by including an IF function such as:
  
  \[
  \text{IF}(\text{Amplitude	extunderscore c} \geq 0, \text{SQRT}(\text{Amplitude	extunderscore c}), \text{null})
  \]

### SUBSTITUTE

**Description:** Substitutes new text for old text in a text string.

**Use:** SUBSTITUTE(text, old_text, new_text) and replace text with the field or value for which you want to substitute values, old_text with the text you want replaced, and new_text with the text you want to replace the old_text.

**Example:** SUBSTITUTE(Name, "Coupon", "Discount") returns the name of an opportunity that contains the term “Coupon” with the opportunity name plus “Discount” wherever the term “Coupon” existed.

SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.") finds the location of the @ sign in a person’s email address to determine the length of text to replace with a “www.” as a means of deriving their website address.

**Tips:**
- Each term provided in quotes is case sensitive.
- If the old_text appears more than once, each occurrence is replaced with the new_text value provided even when that results in duplicates.

### TEXT

**Description:** Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in validation rules, formula fields, and field updates.

**Use:** TEXT(value) and replace value with the field or expression you want to convert to text format. Avoid using any special characters besides a decimal point (period) or minus sign (dash) in this function.

**Example:**

**Expected Revenue in Text**

TEXT(ExpectedRevenue) returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the Expected Revenue of a campaign is "$200,000,” this formula field displays “200000.”

**Asset ID**

SerialNumber &"-"& TEXT(Quantity) returns an asset ID number starting with the serial number and ending with the quantity separated by a dash. The Serial Number field is already text but the Quantity field is a number, requiring the TEXT function before it.
### Use Picklist Values in Math Equations

\[
\text{VALUE(LEFT(TEXT(Quantity), 5))} \times \text{Unit}
\]

This formula multiplies the first five numbers of the `Quantity` picklist by the `Unit` numeric field.

### Compare Two Picklists

\[
\text{IF(TEXT(bug_status) = TEXT(case_status), "Match", "Out of Sync")}
\]

This formula compares the values of the `bug_status` picklist with values of the `case_status` picklist.

### Display Picklist Values From Parent Records

\[
\text{TEXT(Account.Industry)}
\]

This formula field on opportunities shows the industry of the associated account.

### Concatenate Picklist Values

\[
\text{TEXT(Account.Industry)} \& " - " \& \text{TEXT(Account.SubIndustry__c)}
\]

This formula field on opportunities shows the industry and subindustry of the associated account.

### Validation Rule Examples: Block the Save of a Closed Opportunity

`CONTAINS(TEXT(Status), "Closed")` returns TRUE if the `Status` picklist contains the value “Closed,” such as “Closed Won” and “Closed Lost.” This validation rule formula blocks users from saving changes to a closed opportunity.

### Use Numeric Functions on Numeric Picklist Values

\[
\text{VALUE(LEFT(TEXT(Quantity), 5))} \times \text{Unit} > 10000
\]

This formula multiplies the first five numbers of the `Quantity` picklist by the `Unit` numeric field, and returns TRUE if the result is greater than 10,000.

### Directly Compare Two Picklists

\[
\text{TEXT(bug_status) = TEXT(case_status)}
\]

This formula compares the values of the `bug_status` picklist with values of the `case_status` picklist, and returns TRUE if they are equal.

### Tips:
- The returned text is not formatted with any currency, percent symbols, or commas.
- Values are not sensitive to locale. For example, 24.42 EUR are converted into the number 24.42.
- Percents are returned in the form of a decimal.
• Dates are returned in the form of YYYY-MM-DD, that is, a four-digit year and two-digit month and day.
• Date/time values are returned in the form of YYYY-MM-DD HH:MM:SSZ where YYYY is a four-digit year, MM is a two-digit month, DD is a two-digit day, HH is the two-digit hour, MM are the minutes, SS are the seconds, and Z represents the zero meridian indicating the time is returned in UTC time zone.
• Picklist fields are only supported in TEXT functions used in validation rule formulas, formula fields, and field updates. In other formulas, use ISPICKVAL or CASE when referencing a picklist field.
• The TEXT function always returns picklist values in your organization’s master language, not the language of the current user.

---

TODAY

| Description: | Returns the current date as a date data type. |
| Use: | TODAY() |
| Example: | TODAY()-Date_in_approval__c calculates how many days a contract is in the approval process. This example is a number formula field on contracts that uses a custom date field called Date in approval. |
| Validation Rule Example: | CloseDate < TODAY() |
| This example ensures that users cannot change the Close Date of an opportunity to any date in the past. |

Tips:

• Do not remove the parentheses.
• Keep the parentheses empty. They do not need to contain a value.
• Use a date field with a TODAY function instead of a date/time field. Last Activity Date is a date field whereas Created Date and Last Modified Date are date/time fields.
• See NOW if you prefer to use a date/time field.
• Dates and times are always calculated using the user’s time zone.
• Use addition and subtraction operators with a TODAY function and other date fields to return a number, representing number of days. For example TODAY()-LastActivityDate calculates the number of days since the last activity date. In this example, the formula field data type is a number.
• Use addition and subtraction operators with a TODAY function and numbers to return a date. For example TODAY() +5 calculates the date five days ahead of today. In this example, the formula field data type is a date.
**TRIM**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Removes the spaces and tabs from the beginning and end of a text string.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>TRIM(text) and replace text with the field or expression you want to trim.</td>
</tr>
<tr>
<td>Example:</td>
<td>TRIM(LEFT(LastName,5)) &amp; &quot;-&quot; &amp; RIGHT(FirstName, 1) returns a network ID for users that contains the first five characters of their last name and first character of their first name separated by a dash.</td>
</tr>
</tbody>
</table>

**UPPER**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td>UPPER(text, [locale]) and replace text with the field or expression you wish to convert to uppercase, and locale with the optional two-character ISO language code or five-character locale code, if available. For information on supported languages, see What languages does Salesforce.com support?.</td>
</tr>
</tbody>
</table>
| Example:    | SALESFORCE.COM  
UPPER("salesforce.com") returns “SALESFORCE.COM.”  
SALESFORCE.COM 123  
UPPER("Salesforce.com 123") returns “SALESFORCE.COM 123.” |

**Applying Turkish Language Locale Rules**

The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the UPPER() function with the Turkish language locale, use the Turkish locale code tr in the UPPER() function as follows:

```plaintext
UPPER(text, “tr”)  
```

This ensures that Salesforce.com does not transform any dotted i in the text to a dotless I.

**URLENCODE**

<table>
<thead>
<tr>
<th>Description:</th>
<th>Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. For example, blank spaces are replaced</th>
</tr>
</thead>
</table>
with %20, and exclamation points are replaced with %21.

<table>
<thead>
<tr>
<th>Use:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{!URLENCODE(text)}</code> and replace text with the merge field or text string that you want to encode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the merge field foo__c contains <code>&lt;B&gt;Mark’s page&lt;/B&gt;</code>, <code>{!URLENCODE(foo__c)}</code> results in: <code>%3CB%3EMark%27s%20page%3C%2Fb%3E</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tips:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Custom buttons and links with URL content sources have separate encoding settings. If you use the URL ENCODE function to encode a custom button or link that has an encoding setting specified, Salesforce.com first encodes the URL according to the custom button or link setting, then encodes the result. For example, if the URL in a custom link contains a space and its encoding setting is UTF8, Salesforce.com first encodes the space to a plus sign (+), then the URL ENCODE function converts the plus sign to its character code, %2B.</td>
</tr>
<tr>
<td>• When you include the standard Account field on opportunities (Opportunity.Account) in the URL ENCODE function, the value of the field is the account ID, not the account name. To encode the account name, create a custom cross-object formula field on opportunities that spans to the account name, and use that field in the URL ENCODE function instead of Opportunity.Account. For example, if the cross-object formula is AccountNameFormula__c, use the following:</td>
</tr>
<tr>
<td><code>http://www.google.com/search?q={!URLENCODE(Opportunity.AccountNameFormula__c)}</code></td>
</tr>
</tbody>
</table>

**URLFOR**

<table>
<thead>
<tr>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a relative URL for an action, s-control, or a file in a static resource archive in a Visualforce page.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{!URLFOR(target, id, [inputs], [no override])}</code> and replace target with the URL or action, s-control, or static resource merge variable, id with a reference to the record, and inputs with any optional parameters. The no override argument is also</td>
</tr>
</tbody>
</table>
optional and defaults to “false.” It applies to targets for standard Salesforce.com pages such as $Action.Account.New. Replace no override with “true” when you want to display a standard Salesforce.com page regardless of whether you have defined an override for it elsewhere.
S-Control Example:

```html
<html>
<head>
<script src="/soap/ajax/13.0/connection.js"></script>
<script>
function init() {
    var queryResult = sforce.connection.query("Select count() from CaseSolution Where CaseId = '{!Case.Id}'");
    var size = queryResult.size;
    if (size > 0) {
        // go to the standard case close page without invoking the override
    } else {
        alert("Case must contain at least one solution before it can be closed.");
        // go to the standard case detail page
        this.parent.location.href = '{!URLFOR($Action.Case.View, Case.Id)}";
    }
}
</script>
</head>
<body onload="init()">
<p>&nbsp;</p>
</body>
</html>
```

In this example, the s-control displays a message to users when they click Close Case on a case with no associated solutions. If the case is associated with at least one solution, the close case page displays as usual. To implement this example, create an s-control with the content above. Then, override the close case option using the s-control.

This example shows how you can validate data using a standard button override. Use validation rules to validate data on a record and an s-control like the one in this example to validate associated data. This example does not apply to records updated using the API.

This example uses the URLFOR function to build Salesforce.com URLs rather than hard-coding them, which is more scalable. It also sets the no override argument to “true,” telling Salesforce.com to ignore the override and call a standard close case action within an override.

Visualforce Example:

```apex:image url="(!URLFOR($Resource.TestZip, 'images/Bluehills.jpg'))" width="50" height="50"
```

In this example, the <apex:image> component references a .jpg file contained within a .zip file that has been uploaded as a static resource. When uploaded, the name of the static resource was defined
as TestZip, and the path to the image within the resource is images/Bluehills.jpg.

Tips:
- Use global variables to access special merge fields for actions, s-controls, and static resources.
- If an input parameter name begins with any character other than a letter or dollar sign ($), enclose it in quotation marks.
- Enclose multiple inputs in brackets to indicate they are together:

  ```
  {!URLFOR($Action.Case.View, Case.Id, [parm1="A", parm2="B"])}
  ```

- Set inputs to null if you do not have any to pass yet you want to set the no override argument:

  ```
  {!URLFOR($Action.Case.View, Case.Id, null, true)}
  ```

- When you override a standard action, that action is no longer available in Salesforce.com. For example, if you override the new account action, that affects the New button on all pages such as the account detail page, account related lists on other detail pages, and the Create New drop down list in the sidebar. To override a standard action yet still access it, use the no override argument in your s-control to reference that action.

- When you override the tab home page for a standard or custom tab, set target to the "Tab" $Action global variable and id to the object type. For example,

  ```
  URLFOR($Action.Account.Tab, $ObjectType.Account)
  ```

- This function is only available in custom buttons, links, s-controls, and Visualforce pages.

### VALUE

**Description:** Converts a text string to a number.

**Use:** `VALUE(text)` and replace text with the field or expression you want converted into a number.

**Example:**

**Lead Number**

`VALUE(Lead_Number__c)` returns a number for the text value in the auto-number field Lead Number. This can be useful if you want to use the Lead Number field in a calculation. Note that auto-number fields are actually text fields and must be converted to a number for numeric calculations.

**Round Robin Lead Assignment**

`MOD(VALUE(Lead_Number__c), 3)`

This formula is for a custom formula field named Round_Robin_ID that assigns each lead a value of 0, 1, or 2. This formula uses a custom
auto-number field called Lead Number that assigns each lead a sequential number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example:

- Round_Robin_ID = 0 is assigned to Queue A
- Round_Robin_ID = 1 is assigned to Queue B
- Round_Robin_ID = 2 is assigned to Queue C

**Tips:**

Make sure the text in a VALUE function does not include any special characters other than a decimal point (period) or minus sign (dash). For example, the formula `VALUE(Text_field__c)` produces these results:

- If `Text field` is 123, the result is 123
- If `Text field` is blank, the result is #Error!
- If `Text field` is $123, the result is #Error!
- If `Text field` is EUR123, the result is #Error!

---

**VLOOKUP**

**Description:**

Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.

**Use:**

```
VLOOKUP(field_to_return, field_on_lookup_object, lookup_value)
```

and replace `field_to_return` with the field that contains the value you want returned, `field_on_lookup_object` with the field on the related object that contains the value you want to match, and `lookup_value` with the value you want to match.

**Validation Rule Example:**

This example checks that a billing postal code is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid zip code in the US. If the zip code is not found in the Zip_Code__c object or the billing state does not match the corresponding State_Code__c in the Zip_Code__c object, an error is displayed.

**Note:**

- Use this example when the billing country is US or USA.

```plaintext
AND(
    LEN(BillingPostalCode) > 0,
    OR(BillingCountry = "USA",
        BillingCountry = "US"),
    VLOOKUP(
        $ObjectType.Zip_Code__c.Fields.State_Code__c,
        $ObjectType.Zip_Code__c.Fields.Name,
        LEFT(BillingPostalCode,5))
)```
Tips:

- The `field_to_return` must be an auto number, roll-up summary, lookup relationship, master-detail relationship, checkbox, date, date/time, email, number, percent, phone, picklist, text, text area, or URL field type.
- The `field_on_lookup_object` must be the Record Name field on a custom object.
- The `field_on_lookup_object` and `lookup_value` must be the same data type.
- If more than one record matches, the value from the first record is returned.
- The value returned must be on a custom object.
- You cannot delete the custom field or custom object referenced in this function.
- This function is only available in validation rules.

YEAR

<table>
<thead>
<tr>
<th>Description:</th>
<th>Returns the four-digit year in number format of a given date.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use:</td>
<td><code>YEAR(date)</code> and replace <code>date</code> with the field or expression that contains the year you want returned.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>YEAR(TODAY()) - YEAR(Initial_Meeting__c)</code> returns the number of years since your initial meeting with a client. This example uses a custom date field called <code>Initial_Meeting</code>.</td>
</tr>
</tbody>
</table>