

CALC_AGE

	A	B	C
1	Name	Date of Birth	Age
2	John	5-15	33
3	Jane	1985-09-20	38
4	Alex	2002-11-15	20

In this example, we'll create a custom formula that calculates the age of a person based on their date of birth.

Scenario: You want to create a custom formula that calculates the age of a person based on their date of birth.

Data Table:

A	B	C
Name	Date of Birth	Age
John	1990-05-15	
Jane	1985-09-20	
Alex	2002-03-10	
Total		

Step 1: Setting up the Spreadsheet

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Create a new Google Sheets document.

Enter the names and date of birth in columns A and B starting from row 2.

Leave cells in column C empty for now.

Step 2: Writing the Google Apps Script Code

Click on "Extensions" in the top menu, then select "Apps Script".

Delete any default code and replace it with the following script:

```
function CALC_AGE(dateOfBirth) {
  var dob = new Date(dateOfBirth);
  var today = new Date();
  var age = calculateAge(dob, today);
  return age;
}

function calculateAge(dob, today) {
  var yearsDiff = today.getFullYear() - dob.getFullYear();
  var monthsDiff = today.getMonth() - dob.getMonth();
  var daysDiff = today.getDate() - dob.getDate();
  if (monthsDiff < 0 || (monthsDiff === 0 && daysDiff < 0)) {
    yearsDiff--;
  }
  return yearsDiff;
}
```

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Step 3: Using the Custom Formula in Google Sheets

Go back to your Google Sheets document.

In cell C2, enter the following formula:

```
=CALC_AGE(B2:B)
```

Step 3: Testing the Custom Formula

Enter names and dates of birth in columns A and B starting from row 2.

Use the custom formula in cell C2 to calculate the ages based on the provided dates of birth.

For example, if you have entered the names and dates of birth as shown in the data table, the calculated ages should appear in column C.

Remember to enable the "Google Apps Script" extension and use the exact function name (CALC_AGE) in your formula.

Code Explained

This Google Apps Script code defines two functions, CALC_AGE and calculateAge, that work together to calculate a person's age based on their date of birth.

CALC_AGE(dateOfBirth) Function:

This function takes one argument, dateOfBirth, which is expected to be a string representing a date in a specific format (e.g., "YYYY-MM-DD").

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Inside the function, it first converts the `dateOfBirth` string into a JavaScript Date object using the new `Date(dateOfBirth)` constructor. This allows for easier date manipulation and comparison.

It then creates another Date object called `today`, representing the current date.

Next, it calls the `calculateAge` function and passes both the `dob` (date of birth) and `today` as arguments.

Finally, it returns the calculated age, which is the result of the `calculateAge` function.

`calculateAge(dob, today)` Function:

1. This function is called by the `CALC_AGE` function and takes two Date objects as arguments: `dob` (date of birth) and `today` (current date).
2. It calculates the age by performing the following steps:
3. It calculates the difference in years between the current year (`today.getFullYear()`) and the year of birth (`dob.getFullYear()`).
4. It calculates the difference in months between the current month (`today.getMonth()`) and the month of birth (`dob.getMonth()`).
5. It calculates the difference in days between the current day (`today.getDate()`) and the day of birth (`dob.getDate()`).
6. It then checks if the months difference is negative (indicating the birthday hasn't occurred yet in the current year) or if the months difference is zero but the days difference is negative (indicating the birthday is yet to come in

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

the current month). If either of these conditions is true, it subtracts 1 from the calculated years difference.

7. Finally, it returns the calculated age, which represents the number of full years between the date of birth and the current date.

Here's an example of how you can use the CALC_AGE function:

```
var dateOfBirth = "1990-05-15"; // Replace with the
person's actual date of birth in the "YYYY-MM-DD"
format.
```

```
var age = CALC_AGE(dateOfBirth);
console.log("Age: " + age + " years");
```

In this example, dateOfBirth is set to a specific date, and then the CALC_AGE function is called to calculate the age. The result is displayed in the console.

```
function CALC_AGE(dobval){
const dob = new Date(dobval);
const today = new Date();
const age = calculateAge(dob, today);
return age;
}
```

```
function calculateAge(dob, today){
let yearsDiff = today.getFullYear() - dob.getFullYear();
const monthDiff = today.getMonth() - dob.getMonth();
const daysDiff = today.getDate() - dob.getDate();
```

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

```

if(monthDiff < 0 || (monthDiff === 0 && daysDiff <0)){
yearsDiff--;
}
return yearsDiff;
}

```

Custom formula to convert currency using real-time exchange rates

Amount	From	To Currency	Exchange	Total
100.00	USD	EUR		
50.00	USD	GBP		
2,000.00	USD	CAD		
300.00	USD	CNY		
136.00	CAD	USD		
50.00	CAD	GBP		
2,000.00	CAD	IDR		
300.00	CAD	INR		

=GoogleFinance(concat(B2,C2:C))

Let's break down the components of this formula:

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

`concat(B2, C2:C):`

`concat` is a Google Sheets function used for concatenating (joining) text strings together.

B2 is likely a reference to a cell containing a stock symbol or ticker symbol, such as "AAPL" for Apple Inc.

C2:C appears to be a reference to a range of cells that might contain additional information or modifiers for the stock, such as start and end dates for historical data.

`GoogleFinance():`

`GoogleFinance` is a built-in Google Sheets function that allows you to fetch various types of financial information for a specified stock or financial instrument.

When used alone, `GoogleFinance` typically requires a single argument, which is a reference to the stock symbol or a cell containing the symbol.

Putting it all together:

`concat(B2, C2:C)` concatenates the stock symbol in cell B2 with any additional information or modifiers from the range C2:C. This can be used to dynamically construct a query for `GoogleFinance`.

However, there are a few important considerations to keep in mind:

The exact functionality of this formula may depend on how your spreadsheet is set up, what's in cell B2, and what's in the range C2:C. It's possible that the formula has been customized for a specific purpose in your spreadsheet.

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

The usage of `concat` in combination with `GoogleFinance` may not be a standard or documented approach for using `GoogleFinance` in Google Sheets. Typically, you would directly provide the stock symbol as an argument to `GoogleFinance`, not concatenate it with other cell values.

In summary, the formula `=GoogleFinance(concat(B2, C2:C))` seems to be an attempt to dynamically construct a `GoogleFinance` query, but its specific behavior and effectiveness will depend on the content of the referenced cells and how your spreadsheet is structured.

```
// Custom formula to convert currency using real-time exchange rates  
function CONVERT_CUR(amount, xVal) {  
  return amount * xVal;  
}
```

Find out more about Google Finance

<https://www.google.com/finance/markets/currencies?hl=en>

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

D2 | `=GoogleFinance(concat(B2,C2:C))`

	A	B	C	D	E
1	Amount	From	To Currency	Exchange	Total
2	100.00	USD	EUR	0.954295	95.43
3	50.00	USD	GBP	0.827205	41.36
4	2,000.00	USD	CAD	1.36715	2,734.30
5	300.00	USD	CNY	7.1821	2,154.63
6	136.00	CAD	USD	0.731425	99.47
7	50.00	CAD	GBP	0.6050384171	30.25
8	2,000.00	CAD	IDR	11359.76167	22,719,523.34
9	300.00	CAD	INR	60.97045429	18,291.14

The above Google Apps Script code defines a simple JavaScript function named CONVERT_CUR that takes two parameters: amount and xVal. This function appears to be designed for currency conversion, where amount is the amount of money you want to convert, and xVal is the exchange rate or conversion factor.

Here's a detailed explanation of the code:

```
function CONVERT_CUR(amount, xVal) {
  return amount * xVal;
}
```

function CONVERT_CUR(amount, xVal) {: This line defines a JavaScript function named CONVERT_CUR with two parameters - amount and xVal. Functions in JavaScript are blocks of reusable code that can be called with specific inputs to perform a specific task.

Learn more about Google Apps Scripts with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

return amount * xVal; This line contains the code that will be executed when the CONVERT_CUR function is called. It calculates the result of multiplying the amount by xVal and then returns the result. The * operator is used for multiplication in JavaScript.

Here's how you can use this function:

```
// Example usage:
var amountInUSD = 100; // The amount in US Dollars
var exchangeRate = 1.23; // The exchange rate (e.g., 1
USD to 1.23 EUR)

var convertedAmount = CONVERT_CUR(amountInUSD,
exchangeRate);

console.log("Converted amount: " + convertedAmount);
// This will print "Converted amount: 123" to the
console.
```

In this example, the CONVERT_CUR function is called with amountInUSD (100) and exchangeRate (1.23) as arguments. It returns the converted amount, which is 123 in this case, and stores it in the convertedAmount variable. Finally, the converted amount is printed to the console.

This function is a simple currency conversion calculator that can be used to convert one currency to another by providing the amount and the exchange rate as input.