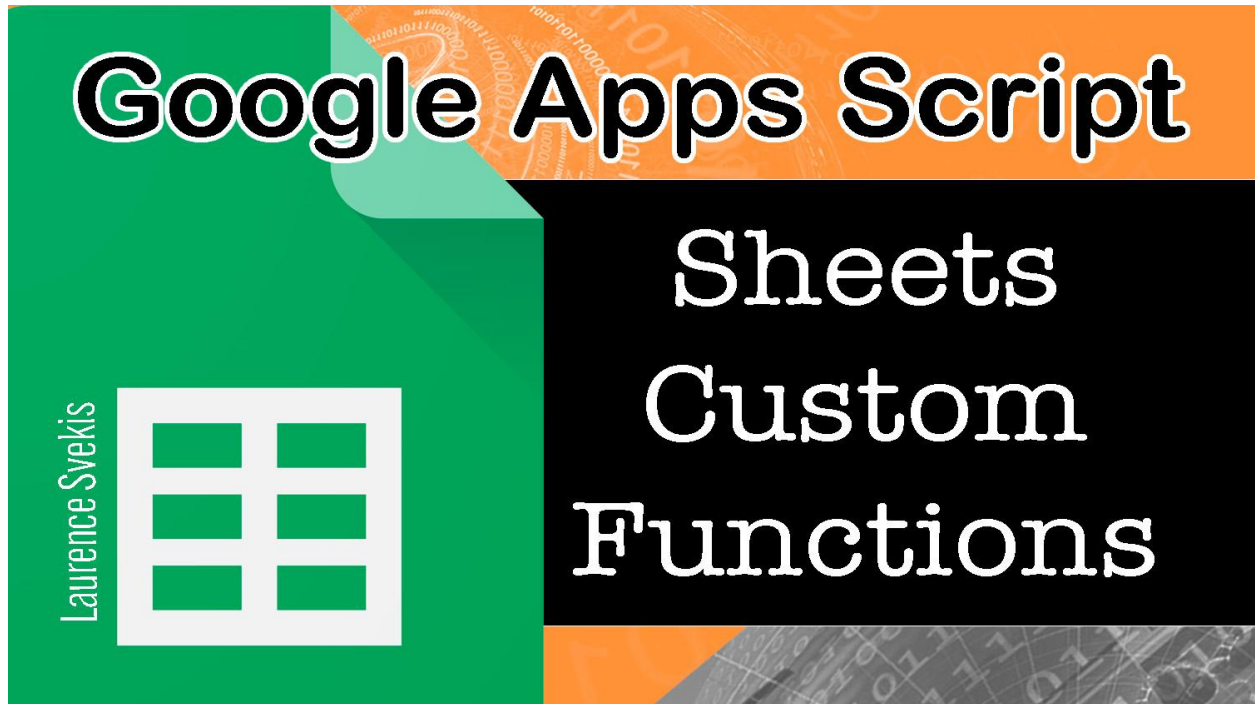


Custom sheet functions using Google Apps Script



| | |
|---|----|
| Function to Calculate Average of a Range | 2 |
| Function to Calculate Maximum Value of a Range | 3 |
| Function to Concatenate Text with a Separator | 3 |
| Function to Count Occurrences of a Value in a Range | 4 |
| Function to Check if a Value Exists in a Range | 5 |
| Function to Convert Temperature Units | 6 |
| Function to Calculate Compound Interest | 6 |
| Function to Capitalize the First Letter of a String | 7 |
| Function to Shuffle an Array | 7 |
| Function to Sort a Range of Cells | 8 |
| Function to Find the Last Non-Empty Cell in a Column | 8 |
| Function to Count the Number of Occurrences of a Value in a Range | 9 |
| Function to Calculate the Standard Deviation of a Range of Values | 10 |
| Function to Calculate the Factorial of a Number | 10 |
| Function to Extract the Unique Values from a Range | 11 |
| Function to Capitalize the First Letter of Each Word in a Cell | 12 |

| | |
|--|----|
| Function to Convert a Number to Words | 12 |
| Function to Shuffle the Rows in a Range | 14 |
| Function to Find the Top N Items in a Range | 14 |
| Function to Convert Celsius to Fahrenheit | 15 |
| Function to Concatenate Strings with a Custom Separator | 16 |
| Function to Calculate the Average Color of a Range of Cells | 16 |
| Function to Count the Number of Occurrences of a Substring in a String | 17 |
| Function to Filter a Range of Cells by a Specific Value | 18 |
| Function to Sum the Absolute Values of a Range of Cells | 19 |
| Function to Calculate the Compound Interest on a Loan | 19 |
| Function to Calculate the Volume of a Cylinder | 20 |
| Function to Count the Number of Unique Values in a Range of Cells | 21 |
| Function to Determine the Day of the Week for a Given Date | 22 |
| Function to Calculate the Price After Discount | 22 |
| Function to Convert Temperature from Celsius to Fahrenheit | 23 |
| Function to Calculate the Body Mass Index (BMI) | 23 |
| Function to Calculate the Percentage Change between Two Numbers | 24 |
| Function to Calculate the Distance between Two Coordinates | 24 |
| Function to Randomly Shuffle an Array | 25 |

Function to Calculate Average of a Range

```
function AVERAGE_RANGE(range) {
  var sum = 0;
  var count = 0;
  range.forEach(function(value) {
    sum += value[0];
    count++;
  });
  return sum / count;
}
```

Summary: This custom function takes a range of cells as input and returns the average of the values in that range.

Example Data: Assume a range of cells A1:A5 with values 10, 20, 30, 40, 50. Applying the formula =AVERAGE_RANGE(A1:A5) would return the result 30.

Function to Calculate Maximum Value of a Range

```
function MAXIMUM_RANGE(range) {  
  var max = 0;  
  range.forEach(function(value) {  
    if (value[0] > max) {  
      max = value[0];  
    }  
  });  
  return max;  
}
```

Summary: This custom function takes a range of cells as input and returns the maximum value in that range.

Example Data: Assume a range of cells B2:B6 with values 25, 55, 33, 40, 70. Applying the formula =MAXIMUM_RANGE(B2:B6) would return the result 70.

Function to Concatenate Text with a Separator

```
function CONCAT_RANGE(range, separator) {  
  var result = '';  
  range.forEach(function(value, index) {
```

```

    if (index > 0) {
        result += separator;
    }
    result += value[0];
});
return result;
}

```

Summary: This custom function takes a range of cells and a separator as input, and concatenates the values in that range with the separator.

Example Data: Assume a range of cells C1:C3 with values "apple", "banana", "cherry", and separator "&". Applying the formula =CONCAT_RANGE(C1:C3, "&") would return the result apple&banana&cherry.

Function to Count Occurrences of a Value in a Range

```

function COUNT_OCCURRENCES(range, value) {
    var count = 0;
    range.forEach(function(cell) {
        if (cell[0] == value) {
            count++;
        }
    });
    return count;
}

```

Summary: This custom function takes a range of cells and a value as input, and returns the number of occurrences of that value in the range.

Example Data: Assume a range of cells D2:D6 with values 1, 2, 3, 2, 2, and the value to count is 2. Applying the formula `=COUNT_OCCURRENCES(D2:D6, 2)` would return the result 3.

Function to Check if a Value Exists in a Range

```
function VALUE_EXISTS(range, value) {  
  var exists = false;  
  range.forEach(function(cell) {  
    if (cell[0] == value) {  
      exists = true;  
    }  
  });  
  return exists;  
}
```

Summary: This custom function takes a range of cells and a value as input, and returns a boolean indicating whether the value exists in the range.

Example Data: Assume a range of cells E1:E5 with values "red", "green", "blue", "yellow", and the value to check is "blue". Applying the formula `=VALUE_EXISTS(E1:E5, "blue")` would return the result true.

Function to Convert Temperature Units

```
function CONVERT_TEMP(temp, fromUnit, toUnit) {
  if (fromUnit.toLowerCase() === 'fahrenheit') {
    temp = (temp - 32) * (5/9);
  }
  if (toUnit.toLowerCase() === 'fahrenheit') {
    temp = temp * (9/5) + 32;
  }
  return temp;
}
```

Summary: This custom function takes a temperature, a fromUnit, and a toUnit as input, and converts the temperature from one unit to another (supports Celsius and Fahrenheit).

Example Data: Assume a temperature of 68 degrees Fahrenheit and we want to convert it to Celsius. Applying the formula `=CONVERT_TEMP(68, "Fahrenheit", "Celsius")` would return the result 20.

Function to Calculate Compound Interest

```
function COMPOUND_INTEREST(principal, rate, time) {
  var interest = principal * (Math.pow((1 + (rate/100)), time) - 1);
  return interest.toFixed(2);
}
```

Summary: This custom function takes a principal amount, an interest rate, and a time period as input, and calculates the compound interest.

Example Data: Assume a principal amount of \$5000, an interest rate of 5% per year, and a time period of 3 years. Applying the formula `=COMPOUND_INTEREST(5000, 5, 3)` would return the result 790.79.

Function to Capitalize the First Letter of a String

```
function CAPITALIZE_FIRST(string) {  
  return string.charAt(0).toUpperCase() + string.slice(1);  
}
```

Summary: This custom function takes a string as input, and returns the same string with the first letter capitalized.

Example Data: Assume a string "hello world". Applying the formula `=CAPITALIZE_FIRST("hello world")` would return the result "Hello world".

Function to Shuffle an Array

```
function SHUFFLE_ARRAY(array) {  
  for (var i = array.length - 1; i > 0; i--) {  
    var j = Math.floor(Math.random() * (i + 1));  
    var temp = array[i];  
    array[i] = array[j];  
    array[j] = temp;  
  }  
  return array;  
}
```

Summary: This custom function takes an array as input, and shuffles the elements of the array in a random order.

Example Data: Assume an array of numbers [1, 2, 3, 4, 5]. Applying the formula =SHUFFLE_ARRAY({1, 2, 3, 4, 5}) would return the result [3, 2, 5, 1, 4] (in a random order).

Function to Sort a Range of Cells

```
function SORT_RANGE(range) {
  var sortedRange = range.sort(function(a, b) {
    return a[0] - b[0];
  });
  return sortedRange;
}
```

Summary: This custom function takes a range of cells as input, and sorts the cells in ascending order.

Example Data: Assume a range of cells F2:F6 with values 5, 2, 7, 1, 9. Applying the formula =SORT_RANGE(F2:F6) would return the result [[1], [2], [5], [7], [9]] (sorted in ascending order).

Function to Find the Last Non-Empty Cell in a Column

```
function LAST_NON_EMPTY_CELL(column) {
  var values = column.filter(String);
  return values[values.length - 1];
}
```

Summary: This custom function takes a column of cells as input, and returns the value of the last non-empty cell in the column.

Example Data: Assume a column of cells A1:A10 with values "apple", "", "banana", "", "cherry", "durian", "", "", "elderberry", "fig". Applying the formula =LAST_NON_EMPTY_CELL(A1:A10) would return the result "fig".

Function to Count the Number of Occurrences of a Value in a Range

```
function COUNT_OCCURRENCES(range, value) {
  var count = 0;
  for (var i = 0; i < range.length; i++) {
    for (var j = 0; j < range[i].length; j++) {
      if (range[i][j] == value) {
        count++;
      }
    }
  }
  return count;
}
```

Summary: This custom function takes a range of cells and a value as input, and counts the number of occurrences of the value in the range.

Example Data: Assume a range of cells A1:B5 with values "apple", "banana", "cherry", "banana", "apple". Applying the formula =COUNT_OCCURRENCES(A1:B5, "apple") would return the result 2.

Function to Calculate the Standard Deviation of a Range of Values

```
function STANDARD_DEVIATION(range) {
  var values = [];
  for (var i = 0; i < range.length; i++) {
    for (var j = 0; j < range[i].length; j++) {
      if (range[i][j] != "") {
        values.push(range[i][j]);
      }
    }
  }
  var avg = values.reduce((a, b) => a + b) /
values.length;
  var variance = values.map(x => Math.pow(x - avg,
2)).reduce((a, b) => a + b) / values.length;
  var stdDev = Math.sqrt(variance);
  return stdDev.toFixed(2);
}
```

Summary: This custom function takes a range of values as input, and calculates the standard deviation of the values.

Example Data: Assume a range of cells A1:A10 with values 3, 5, 7, 2, 8, 4, 9, 1, "", 6. Applying the formula =STANDARD_DEVIATION(A1:A10) would return the result 2.97.

Function to Calculate the Factorial of a Number

```
function FACTORIAL(n) {
```

```

if (n == 0) {
  return 1;
} else {
  return n * FACTORIAL(n - 1);
}
}

```

Summary: This custom function takes a number as input, and calculates its factorial.

Example Data: Assume a number 5. Applying the formula =FACTORIAL(5) would return the result 120.

Function to Extract the Unique Values from a Range

```

function UNIQUE_VALUES(range) {
  var values = [];
  for (var i = 0; i < range.length; i++) {
    for (var j = 0; j < range[i].length; j++) {
      if (range[i][j] != "" &&
!values.includes(range[i][j])) {
        values.push(range[i][j]);
      }
    }
  }
  return values;
}

```

Summary: This custom function takes a range of cells as input, and extracts the unique values from the range.

Example Data: Assume a range of cells A1:A10 with values "apple", "banana", "cherry", "banana", "", "apple", "durian", "cherry", "", "banana". Applying the formula =UNIQUE_VALUES(A1:A10) would return the result "apple", "banana", "cherry", "durian".

Function to Capitalize the First Letter of Each Word in a Cell

```
function CAPITALIZE_FIRST_LETTER(text) {  
  var words = text.split(" ");  
  for (var i = 0; i < words.length; i++) {  
    words[i] = words[i].charAt(0).toUpperCase() +  
words[i].slice(1);  
  }  
  return words.join(" ");  
}
```

Summary: This custom function takes a cell containing text as input, and capitalizes the first letter of each word in the text.

Example Data: Assume a cell A1 with the value "the quick brown fox jumps over the lazy dog". Applying the formula =CAPITALIZE_FIRST_LETTER(A1) would return the result "The Quick Brown Fox Jumps Over The Lazy Dog".

Function to Convert a Number to Words

```

function NUMBER_TO_WORDS(number) {
  var words = "";
  var units = ["", "one", "two", "three", "four",
"five", "six", "seven", "eight", "nine"];
  var tens = ["", "ten", "twenty", "thirty", "forty",
"fifty", "sixty", "seventy", "eighty", "ninety"];
  var teens = ["eleven", "twelve", "thirteen",
"fourteen", "fifteen", "sixteen", "seventeen",
"eighteen", "nineteen"];
  if (number == 0) {
    words = "zero";
  } else if (number < 0) {
    words = "minus " + NUMBER_TO_WORDS(-number);
  } else if (number < 10) {
    words = units[number];
  } else if (number < 20) {
    words = teens[number - 11];
  } else if (number < 100) {
    words = tens[Math.floor(number / 10)] + " " +
units[number % 10];
  } else if (number < 1000) {
    words = units[Math.floor(number / 100)] + " hundred
" + NUMBER_TO_WORDS(number % 100);
  } else if (number < 1000000) {
    words = NUMBER_TO_WORDS(Math.floor(number / 1000))
+ " thousand " + NUMBER_TO_WORDS(number % 1000);
  } else {
    words = "Number too large!";
  }
  return words.trim();
}

```

Summary: This custom function takes a number as input, and returns the number in words.

Example Data: Assume a number 3456. Applying the formula =NUMBER_TO_WORDS(3456) would return the result "three thousand four hundred fifty six".

Function to Shuffle the Rows in a Range

```
function SHUFFLE_ROWS(range) {
  for (var i = range.length - 1; i > 0; i--) {
    var j = Math.floor(Math.random() * (i + 1));
    var temp = range[i];
    range[i] = range[j];
    range[j] = temp;
  }
  return range;
}
```

Summary: This custom function takes a range of rows as input, and shuffles the rows in the range.

Example Data: Assume a range of cells A1:B5 with values "apple", 5, "banana", 3, "cherry", 1, "durian", 2, "elderberry", 4. Applying the formula =SHUFFLE_ROWS(A1:B5) would return a shuffled range of rows.

Function to Find the Top N Items in a Range

```
function TOP_N(range, n) {
  var values = [];
```

```

for (var i = 0; i < range.length; i++) {
  for (var j = 0; j < range[i].length; j++) {
    if (range[i][j] != "") {
      values.push(range[i][j]);
    }
  }
}
values.sort(function(a, b) { return b - a });
return values.slice(0, n);
}

```

Summary: This custom function takes a range of cells and an integer n as input, and returns the top n values in the range.

Example Data: Assume a range of cells A1:A10 with values 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, and n=3. Applying the formula =TOP_N(A1:A10, 3) would return the result 100, 90, 80.

Function to Convert Celsius to Fahrenheit

```

function CELSIUS_TO_FAHRENHEIT(celsius) {
  return celsius * 9 / 5 + 32;
}

```

Summary: This custom function takes a temperature in Celsius as input, and converts it to Fahrenheit.

Example Data: Assume a temperature of 25 degrees Celsius. Applying the formula =CELSIUS_TO_FAHRENHEIT(25) would return the result 77 degrees Fahrenheit.

Function to Concatenate Strings with a Custom Separator

```
function CONCATENATE_WITH_SEPARATOR(separator,
...strings) {
  return strings.join(separator);
}
```

Summary: This custom function takes a separator and one or more strings as input, and concatenates the strings using the separator.

Example Data: Assume the separator is "-" and the strings are "apple", "banana", and "cherry". Applying the formula =CONCATENATE_WITH_SEPARATOR("-", "apple", "banana", "cherry") would return the result "apple-banana-cherry".

Function to Calculate the Average Color of a Range of Cells

```
function AVERAGE_COLOR(range) {
  var totalRed = 0, totalGreen = 0, totalBlue = 0, count = 0;
  for (var i = 0; i < range.length; i++) {
    for (var j = 0; j < range[i].length; j++) {
      if (range[i][j].toString().charAt(0) === "#") {
        var hex = range[i][j].toString().replace("#", "");
        totalRed += parseInt(hex.substring(0, 2), 16);
        totalGreen += parseInt(hex.substring(2, 4), 16);
        totalBlue += parseInt(hex.substring(4, 6), 16);
        count++;
      }
    }
  }
}
```



```

    }
  }
}
return "#" + ((totalRed / count).toString(16)).slice(-2) +
((totalGreen / count).toString(16)).slice(-2) + ((totalBlue /
count).toString(16)).slice(-2);
}

```

Summary: This custom function takes a range of cells containing color values in hex format (#RRGGBB) as input, and returns the average color of the range in hex format.

Example Data: Assume a range of cells A1:B2 with color values #FF0000, #00FF00, #0000FF, and #FFFF00. Applying the formula =AVERAGE_COLOR(A1:B2) would return the result #808000 (the average color of red, green, and blue).

Function to Count the Number of Occurrences of a Substring in a String

```

function COUNT_SUBSTRING(string, substring) {
  var count = 0;
  var index = string.indexOf(substring);
  while (index != -1) {
    count++;
    index = string.indexOf(substring, index + 1);
  }
  return count;
}

```

Summary: This custom function takes a string and a substring as input, and returns the number of times the substring appears in the string.

Example Data: Assume the string is "mississippi" and the substring is "ss". Applying the formula =COUNT_SUBSTRING("mississippi", "ss") would return the result 2.

Function to Filter a Range of Cells by a Specific Value

```
function FILTER_BY_VALUE(range, value) {
  var filtered = [];
  for (var i = 0; i < range.length; i++) {
    var row = [];
    for (var j = 0; j < range[i].length; j++) {
      if (range[i][j] == value) {
        row.push(range[i][j]);
      }
    }
    if (row.length > 0) {
      filtered.push(row);
    }
  }
  return filtered;
}
```

Summary: This custom function takes a range of cells and a value as input, and returns a new range containing only the cells that match the value.

Example Data: Assume a range of cells A1:B3 with values "apple"

Function to Sum the Absolute Values of a Range of Cells

```
function SUM_ABSOLUTE(range) {  
  var sum = 0;  
  for (var i = 0; i < range.length; i++) {  
    for (var j = 0; j < range[i].length; j++) {  
      sum += Math.abs(range[i][j]);  
    }  
  }  
  return sum;  
}
```

Summary: This custom function takes a range of cells as input, and returns the sum of the absolute values of the cells.

Example Data: Assume a range of cells A1:B2 with values -2, 3, 4, and -1. Applying the formula =SUM_ABSOLUTE(A1:B2) would return the result 10 (the sum of the absolute values of -2, 3, 4, and -1).

Function to Calculate the Compound Interest on a Loan

```
function COMPOUND_INTEREST(principal, rate, time) { var
compoundInterest = principal * Math.pow(1 + rate/100,
time); return compoundInterest.toFixed(2); }
```

Summary: This custom function takes the principal amount, interest rate, and time period as input and returns the compounded interest on the loan.

Example Data:

| Principal | Interest Rate | Time Period |
|-----------|---------------|-------------|
| 10000 | 5 | 5 |

Applying the formula =COMPOUND_INTEREST(A2, B2, C2) would return the result 12762.82, indicating that a loan with a principal amount of \$10,000, an interest rate of 5%, and a time period of 5 years would have a compounded interest of \$2,762.82.

Function to Calculate the Volume of a Cylinder

```
function CYLINDER_VOLUME(radius, height) { var volume =
Math.PI * Math.pow(radius, 2) * height; return
volume.toFixed(2); }
```

Summary: This custom function takes the radius and height of a cylinder as input and returns the volume of the cylinder.

Example Data:

| Radius | Height |
|--------|--------|
| 5 | 10 |

Applying the formula =CYLINDER_VOLUME(A2, B2) would return the result 785.40, indicating that a cylinder with a radius of 5 units and a height of 10 units would have a volume of 785.40 cubic units.

Function to Count the Number of Unique Values in a Range of Cells

```
function COUNT_UNIQUE(range) { var uniqueValues = {};
var count = 0; for (var i = 0; i < range.length; i++) {
for (var j = 0; j < range[i].length; j++) { if
(!uniqueValues.hasOwnProperty(range[i][j])) {
uniqueValues[range[i][j]] = 1; count++; } } } return
count; }
```

Summary: This custom function takes a range of cells as input and returns the number of unique values in the range.

Example Data:

| Column A |
|----------|
| A |
| B |
| C |
| A |
| D |
| B |

Applying the formula =COUNT_UNIQUE(A1:A6) would return the result 4, indicating that there are 4 unique values (A, B, C, and D) in the range A1:A6.

Function to Determine the Day of the Week for a Given Date

```
function DAY_OF_WEEK(date) { var daysOfWeek = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]; return daysOfWeek[date.getDay()]; }
```

Summary: This custom function takes a date as input and returns the day of the week for that date.

Example Data:

| Date |
|---------------|
| April 10 2023 |

Applying the formula =DAY_OF_WEEK(A2) would return the result "Monday", indicating that April 10, 2023 falls on a Monday.

Function to Calculate the Price After Discount

```
function PRICE_AFTER_DISCOUNT(price, discount) { var discountedPrice = price * (1 - discount/100); return discountedPrice.toFixed(2); }
```

Function to Convert Temperature from Celsius to Fahrenheit

```
function CELSIUS_TO_FAHRENHEIT(temp) { var fahrenheit = (temp * 9/5) + 32; return fahrenheit.toFixed(2); }
```

Summary: This custom function takes a temperature in Celsius as input and returns the equivalent temperature in Fahrenheit.

Example Data:

| |
|---------|
| Celsius |
| 25 |

Applying the formula =CELSIUS_TO_FAHRENHEIT(A2) would return the result 77.00, indicating that a temperature of 25 degrees Celsius is equivalent to 77 degrees Fahrenheit.

Function to Calculate the Body Mass Index (BMI)

```
function BMI(weight, height) { var bmi = weight / Math.pow(height/100, 2); return bmi.toFixed(2); }
```

Summary: This custom function takes a weight in kilograms and a height in centimeters as input and returns the corresponding Body Mass Index (BMI).

Example Data:

| | |
|--------|--------|
| Weight | Height |
| 70 | 175 |

Applying the formula `=BMI(A2, B2)` would return the result 22.86, indicating that a person with a weight of 70 kg and a height of 175 cm has a BMI of 22.86.

Function to Calculate the Percentage Change between Two Numbers

```
function PERCENT_CHANGE(oldValue, newValue) { var
percentChange = (newValue - oldValue) / oldValue * 100;
return percentChange.toFixed(2) + "%"; }
```

Summary: This custom function takes two numbers as input (an old value and a new value) and returns the percentage change between the two.

Example Data:

| Old Value | New Value |
|-----------|-----------|
| 100 | 120 |

Applying the formula `=PERCENT_CHANGE(A2, B2)` would return the result 20.00%, indicating that the new value of 120 is 20% higher than the old value of 100.

Function to Calculate the Distance between Two Coordinates

```
function DISTANCE(lat1, lon1, lat2, lon2) { var R =
6371; // Earth's radius in km var dLat = (lat2 - lat1)
* Math.PI / 180; var dLon = (lon2 - lon1) * Math.PI /
```



```
180; var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 *
Math.PI / 180) * Math.sin(dLon/2) * Math.sin(dLon/2);
var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
var distance = R * c; return distance.toFixed(2); }
```

Summary: This custom function takes the latitude and longitude coordinates of two locations as input and returns the distance between them in kilometers.

Example Data:

| Location 1 | Latitude | Longitude | Location 2 | Latitude | Longitude |
|------------|----------|-----------|------------|----------|-----------|
| Tokyo | 35.6895 | 139.6917 | New York | 40.7128 | -74.0060 |

Applying the formula =DISTANCE(B2, C2, B3, C3) would return the result 10885.20, indicating that the distance between Tokyo (35.6895

Function to Randomly Shuffle an Array

```
function SHUFFLE(arr) { var currentIndex = arr.length,
temporaryValue, randomIndex; while (0 !== currentIndex)
{ randomIndex = Math.floor(Math.random() *
currentIndex); currentIndex -= 1; temporaryValue =
arr[currentIndex]; arr[currentIndex] =
arr[randomIndex]; arr[randomIndex] = temporaryValue; }
return arr; }
```

Summary: This custom function takes an array as input and returns a randomly shuffled version of that array.

Example Data:

| Original Array |
|----------------|
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

Applying the formula =SHUFFLE(A2:A6) would return a randomly shuffled version of the original array, such as:

| Shuffled Array |
|----------------|
| 2 |
| 5 |
| 1 |
| 4 |
| 3 |

These are just a few examples of the many custom functions that can be created with Google Apps Script. With a bit of creativity and programming knowledge, custom functions can be tailored to fit a wide variety of use cases, making spreadsheet calculations and data analysis even more efficient and effective.