Chapter 3 - Exercises

Use the `simple-template` as basis for this exercises:
 * Copy the folder `web-media-centro/code/simple-template`, and paste where you choice.
 * Rename it.

Write the solutions in separated files.

Variables

1. Money exchange:
Write a JS script that:
a. Store in a variable with an amount in MXN.
b. Calculate it USD value.
c. Calculate it EUR value.
d. Display in console.

2. Sale with discounts:
Write a JS script that:
a. Store in multiple variables the cost of some products of a cloth store.
b. Store two discounts: dresses (10%) and socks (50%).
c. Calculate the amount of the sale applying the discounts and display in console.
Example:

Blue dress - $500
Red socks - $100
White t-shirt - $120
Yellow coat - $800

Discount of dress - 10%
Discount of socks - 50%

Total: ($500 - 10%) + ($100 - 50%) + $120 + $800 = $450 + $50 + $120 + $800 = $1420

Functions

1. Family Tree:
Create a function that receive your parents name and your name, and displays a message like that:
“Diego you are the son of Esther and Gilberto”

2. Distance between two points:
Create a function that calculate the distance between two points (every point have two coordinates: x, y).

Hint: Your function should receive four parameters.

3. Money exchange:
Create a function with the functionality of exercise 1 of variables.

Flow control (conditionals)

1. Sign of number:
Create a function that receive a number and display in console the sign of the number.
Example:
sign(-4) displays “-“

sign(10) displays “+”

2. Little sort:
Create a function that receive three numbers and display in console the numbers in increase order.
Example:
order(5, -1, 2) displays “-1, 2, 5”
order(-3, 43, 10) displays “-3, 10, 43”

3. Min number
Create a function that receive five numbers and return the minimum:
Example:
min(4, 1, 0, 300, 43.2) return 0
min(-13, -134, 394, 1, 4) return -134

Flow control (repetitions)

1. Average mark
Create a function that receive three marks of a student and return the average mark.

2. Padding with underscores
Create a function that receive two positive integers and a string. The first number is the amount of underscores to pad the string in left and the second is the amount of underscores to pad the string in right.
Return the result.
Example:
pad(4, 2, ‘Hello World’) returns ‘____Hello World__’
pad(1, 4, ‘My name is’) returns ‘_My name is____’

3. *Star triangle*
Write a script that display the following pattern (use for and while):
*
**
***
****
*****
******
****
****
****

Hint: Use a nested loop (a for inside a for, or a while inside another while)
Bonus: Create a function that receive the number of lines in your pattern, and change the symbol of the pattern.

Arrays

1. First element
Create a function to get the last n elements of an array and return in another array.
Example:
last([7, 9, 0, -2], 1) return [7]
last([2, -2.33, ‘Hola mundo’, 8, ‘Adios mundo’], 3) return [2, -2.33, ‘Hola mundo’]
last([10, 39, 0.22, [‘abc’]], 0) return []

2. Join all
Write a function to join all elements of the following array into a string separated with a specified symbol.
Example:
joinAll(['Red', 'Green', 'White', 'Black'], '-') return 'Red+Green+White+Black'
joinAll(['Red', 'Green', 'White', 'Black'], ',') return 'Red,Green,White,Black'
joinAll(['Red', 'Green', 'White', 'Black'], '3') return 'Red3Green3White3Black'

3. Reverse
Write a function to reverse an array.

Objects

1. The Recipe Card
a. Create an object to hold information on your favorite recipe. It should have properties for title (a string), ingredients an objects with value pairs (ingredient, amount), level of difficulty and the steps (an array of strings).
b. Display the information of the recipe in the console.

2. Contact database
a. Create an array of objects called contacts. Each object have the following info:
* Name
* Home telephone
* Work telephone
* Cell phone
* Address (must be an object with street, number, and country)

b. Create a function to filter the contacts which name begins with a specified letter.

3. Points
a. Create multiple objects representing points in plane:
Example:
var point1 = { x: 10, y: 10 };
var point2 = { x: 123, y: 0 };
var point3 = { x: -34.2, y: 90.0 };

b. Create a function to know if two vectors are equals.
c. Attach a function called `distanceTo(v)` to compute the distance between the owner function and another point that is the parameter.

Happy Hacking!