

Arrays y Funciones

Título	Descripción	Enlace
Por qué la existencia de las funciones	Introducción a las funciones simple	Sección 1 de este documento
Principios de desarrollo importantes: SOLID, DRY, KISS	Introducción y explicación de principios básicos de programación cuando se empiezan a usar funciones dentro de código más grande.	<ul style="list-style-type: none">- https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898- https://keepcoding.io/blog/principio-dry-en-python/https://www.consuunt.es/principio-kiss/
Funciones: teoría inicial	Introducción a cómo se realiza una función.	https://www.aprendejavascrpt.dev/clase/funciones/tu-primera-funcion
Funciones: argumentos y parámetros	Explicación sobre qué son los argumentos y qué son los parámetros en las funciones. Y sí, el orden importa.	https://www.aprendejavascrpt.dev/clase/funciones/parametros
Funciones: como expresión	Introducción a la declaración de funciones y explicación del <i>hoisting</i>	https://www.aprendejavascrpt.dev/clase/funciones/function-expression
Funciones: recursividad	OPCIONAL: que nos suene qué es la recursión y el que quiera que lo revise. ¡Suele ser ejercicio de entrevista!	https://www.aprendejavascrpt.dev/clase/funciones/recursividad
Funciones flecha	OPCIONAL PERO RECOMENDADO: se habla sobre una manera de definir las funciones que está muy extendida en el ámbito de javascript.	https://www.aprendejavascrpt.dev/clase/funciones/flecha
Ejercicio funciones	OPCIONAL: ejercicio con pistas sobre funciones	https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/intermediate-algorithm-scripting/arguments-optional
Apuntes: sección 2 y 3	Datos interesantes para tener en cuenta cuando trabajamos con funciones	-

Introducción a los arrays	Introducción a los arrays con vídeo explicativo	https://www.aprendejavascrip.dev/clase/arrays/arrays-que-son
Introducción a los arrays (II)	Introducción a los arrays y primeras operaciones básicas	https://www.javascripttutorial.net/javascript-array/
Recorrer arrays	Cómo recorrer arrays para hacer operaciones con ellos de distintas maneras.	https://www.aprendejavascrip.dev/clase/arrays/iteracion
Búsqueda en arrays	Otras operaciones interesantes con arrays, sobre todo de búsqueda.	https://www.aprendejavascrip.dev/clase/arrays/buscar
Ordena arrays	Operaciones de ordenación de arrays.	https://www.aprendejavascrip.dev/clase/arrays/ordenar
Ejercicio teórico I	OPCIONAL: ejercicio para practicar sobre arrays	https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/iterate-through-an-array-with-a-for-loop
Ejercicio teórico II	OPCIONAL: ejercicio para practicar sobre arrays	https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-data-structures/add-items-to-an-array-with-push-and-unshift

1. Por qué las funciones

Muchas veces, cuando trabajamos la programación, nos ocurre que un mismo conjunto de código podemos usarlo varias veces. Considerando que la programación se utiliza de manera secuencial, esto significaría que tendríamos que repetir ese mismo código todas las veces que lo usemos.

Aquí entra en valor la idea de utilizar funciones. Esto son pequeños conjuntos de código que realizan una serie de acciones y que siempre se van a repetir de igual manera.

Ventajas de usar funciones:

- **Código más legible y limpio**
- Comienza a poderse utilizar el concepto de **test unitario**
- **Ahorramos tiempo de picar código**

- Nos permite **definir un conjunto de acciones con una abstracción** (el nombre de la función), lo que hace y los argumentos que necesita

2. El tipo de los argumentos

En javascript los argumentos de la función no quedan especificados en el código, de tal manera que cuando se van a usar javascript decide qué es lo que realmente quieres usar. En este caso, si por ejemplo metes una string en un argumento que inicialmente era un int, lo más posible es que cuando se ejecute te dé un error de parámetro o de conversión.

Por eso puede ser interesante documentar las funciones.

Veamos un ejemplo:

```
> function que_haces(a, b){ return a+b };  
undefined  
> que_haces(1, 2)  
3  
> que_haces("suerte", "vaya")  
'suertevaya'  
> que_haces(1, "y ahora que")  
'1y ahora que'  
> que_haces("y ahora que", 2)  
'y ahora que2'  
>
```

3. Alcance o “scope” de las variables

Este es un tema importante que hay que tener en cuenta siempre en programación, y es el alcance de las variables. Generalmente, el alcance de una variable suele estar siempre al mismo nivel en el que se define y en los niveles inferiores, pero no suele estar en los superiores.

Veamos un ejemplo con las funciones:

```

JS ejemplo.js > ...
1   let scope = "global";
2
3   function check_scope(){
4       let scope = "function_scope";
5       function inside() { return scope;}
6       return inside;
7   }
8
9   console.log(check_scope());
10  console.log(scope);

```

¿Qué devolverá cada console log?

Ejercicios

- Arrays

1. Cuando defines un array copiándolo con el método spread en otro array, ¿modificar el segundo modifica el primero? ¿modificar el primero modifica el segundo?
2. Cuando defines un array de tamaño 4, ¿puedes asignar un valor al índice 26?
3. Realiza todos los ejercicios que quieras de la página:
<https://www.w3resource.com/javascript-exercises/javascript-array-exercises.php>
4. Realiza la función reverse de un array sin utilizar `reverse`.
5. Crea una función/programa que borre todos los elementos no deseados de un array (undefined, false, strings vacías y nulls)
 - Al menos prueba con estas tres listas:


```

let list_test_1 = [5, 4, 3, null, 2, "", 1];
let list_test_2 = ["", false, undefined];
let list_test_3 = [5, 4, 1, undefined];
          
```
6. Crea una función que compruebe si dos arrays son idénticos. Uno de los argumentos de la función será si debe ser idéntica en cuanto a contenido o también en cuanto a orden.
7. Crea una función llamada `create_multiples` que tenga dos argumentos: por un lado, un valor de tamaño de array y por otro el número del cuál se van a obtener múltiplos. La función debe devolver un array con el tamaño puesto de múltiplos de ese número. Por ejemplo: (2, 4) => [4, 8].
8. Crea una función `convert_arrays_to_strings` a la cual le introduzcas dos arrays: uno con nombres y otro con edades. La función debe sacar por pantalla las frases con el nombre más los años que tiene, de tal manera que para [Perico, Juan], [22, 33] se pinte por pantalla:

Perico tiene 22 años.

Juan tiene 33 años.

9. Crea un programa (en una página web o con un archivo .js) que te vaya preguntando por input valores hasta que pongas un cero, después, tiene que printear todos los valores por pantalla. *Input: 2, 3, 4, hola, 0. Output: 2, 3, 4, hola.*
10. Crea una función a la que le metas un array y borre los elementos duplicados del array. La función debe devolver otro array.
11. Crea una función que calcule la media de un array numérico. Al menos algunos de los test cases deberían ser:

```
let list_test_1 = [1, 1, 1];  
let list_test_2 = [1, 2, 3];  
let list_test_3 = [];  
let list_test_4 = [undefined, false];
```

12. Crea una función que calcule la desviación típica de un array numérico que introduzcas como entrada.
13. Crea una función que borre todos los elementos de un array que estén entre el 10 y el 20.
14. Sigue los ejercicios de esta página:

<https://www.discoduroderoer.es/ejercicios-propuestos-y-resueltos-arrays-de-java>

- Funciones

15. Crea una función que convierta de Fahrenheit a Celsius. Define otra que convierta de Celsius a Fahrenheit. Define una tercera que haga las dos cosas dependiendo el input, es decir, se añadirá un argumento para decidir cómo tiene que ser la conversión.
16. Crear una función o un programa con página web que pida al usuario que ingrese un dígito y un número, después debe informar el número de veces que aparece el dígito en el número. Por ejemplo, si el número es 22334324 y el dígito 2, debería decir que aparece 3 veces.
17. Pedir que se introduzca un número de dni, después, calcular si el dni es verdadero o no en función de su formato y el número de dígitos. Es decir, habría que comprobar el número de dígitos y la letra.
18. Este programa puede estar escrito en otro lenguaje o en pseudocódigo, pero la sintaxis en es la misma. Después de pensarlo, ¿serías capaz de pasarlo a

javascript?:

9

Sin ejecutar el siguiente programa, determinar cuál es la salida en pantalla si se ingresan los valores $x=6$, $y=7$:

```
def coordenadaZ(x,y):  
    x=x+10  
    y=y+15  
    return x+y  
  
#programa principal  
x=int(input("Coordenada eje x: "))  
y=int(input("Coordenada eje y: "))  
for i in range(3):  
    z=coordenadaZ(x,y)  
    x=x+1  
    y=y+1  
    print(x, " . ",y)
```

19. Escribir una función que, dado un string, retorne la longitud de la última palabra. Se considera que las palabras están separadas por uno o más espacios. También podría haber espacios al principio o al final del string pasado por parámetro y no se deberían tener en cuenta.
20. Escribir la función titulo(string_data), la cual recibe un string y lo retorna convirtiendo la primera letra de cada palabra a mayúscula y las demás letras a minúscula, dejando inalterados los demás caracteres. Precondición: el separador de palabras es el espacio: " ". Un ejemplo sencillo sería: maría tiene un coche gris -> María Tiene Un Coche Gris
21. Crea una función area_circle que te devuelva el área de un círculo.
22. Crea una función que te devuelva el área de un rectángulo y otra que te calcule el área de un triángulo.
23. Crea una función que reciba cuatro esquinas ordenadas (puntos x, y) y devuelva el tamaño en píxeles de la imagen.
24. Crea una función que reciba cuatro esquinas ordenadas y calcule el área del rectángulo.
25. Crea una función que tenga como entrada un string y una vocal y te cambie todas las vocales del string por esa vocal. Después, imprime por pantalla "Cuando Fernando Séptimo llevaba pantalón" con las diferentes vocales.
26. Crea una función que reciba 2 parámetros, precio e iva, y devuelva el precio con iva incluido. Si no recibe el iva, aplicará el 21 por ciento por defecto.
27. Define con una función recursiva cómo calcularías la serie de Fibonacci.
28. Define una función en una página web que al clicar el botón del ratón la llames y te escriba en la página un saludo. Para ello, tendrás que indagar cómo ejecutar funciones con eventos del ratón en Javascript.
29. Una función que pida una dirección de email y valide si la dirección es correcta o no. Documentate en:
https://help.xmatters.com/ondemand/trial/valid_email_format.htm#:~:text=A%20valid%20email%20address%20consists,com%22%20is%20the%20email%20domain.

BIBLIOGRAFÍA

Aprende JavaScript - Curso de JavaScript desde cero y paso a paso. (s. f.-d).

<https://www.aprendejavascript.dev/>

Casero, A. (2024, 24 mayo). Principio DRY en Python: Qué es y cómo aplicarlo

[2024]. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/principio-dry-en-python/>

FreeCodeCamp.org. (s. f.-c). [https://www.freecodecamp.org/learn/javascript-](https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/intermediate-algorithm-scripting/arguments-optional)

[algorithms-and-data-structures/intermediate-algorithm-scripting/arguments-optional](https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/intermediate-algorithm-scripting/arguments-optional)

Principio KISS explicado de manera Práctica con Ejemplos. (s. f.).

<https://www.consuunt.es/principio-kiss/>

Thelma, U. (2024, 30 mayo). The S.O.L.I.D Principles in Pictures - Backticks &

Tildes - Medium. *Medium*. <https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898>