

CURSO: CC3S2

Tarea L-04 : Javascript.

Subir un archivo (CC3S2-LEC04-NOMBRE-APELLIDO.DOC) con los ejercicios resueltos.

Para cada pregunta: Presente la respuesta y muestre el resultado.

1. FILTRADO DE ARRAYS

Los arrays poseen métodos predefinidos que nos permiten manipularlos. Por ejemplo, los métodos `forEach`, `map`, `some` y `filter` son bastante utilizados. Algo muy común es filtrar arrays para que contengan sólo ciertos valores. Para esto podemos utilizar el método `.filter`.
Por ejemplo:

```
const pets = ['cat', 'dog', 'elephant']

const filtered = pets.filter(function (pet) {
  return (pet !== 'elephant')
})
```

La variable `filtered` será igual a un array que contiene solo `cat` y `dog`.

El ejercicio:

Crea un archivo llamado `filtrado-de-arrays.js`.

En ese archivo, define una variable llamada `numbers` que referencie al siguiente array:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17]
```

Luego, define una variable llamada `filtered` que referencie el resultado de `numbers.filter()`.

Las funciones que recibe `.filter()` serán

1. para filtrar los elementos pares.
2. para filtrar los primos.
3. Para filtrar los multiplos de 3

Utiliza `console.log()` para imprimir las 3 variantes del array filtrado a la terminal.

2. Objetos:

Los objetos son en cierta forma contenedores y se los puede pensar cómo diccionarios.

Tendrá ciertas llaves y cada una se verá referenciada a un valor.

Por ejemplo:

```
const foodPreferences = {
  pizza: 'yum',
  salad: 'gross'
}
```

En el ejemplo anterior podemos ver que las llaves del objeto `foodPreferences` son `pizza` y `salad`. Sus valores son `yum` y `gross` respectivamente.

El ejercicio:

Crea un archivo llamado `objects.js`.

En ese archivo, define una variable llamada pizza de la siguiente forma:

```
const pizza = {  
  name: 'Napolitana'  
  toppings: ['cheese', 'sauce', 'pepperoni'],  
  crust: 'deep dish',  
  serves: 2  
}
```

Utiliza console.log() para imprimir un arreglo de 5 pizzas la terminal.

3. Modifique el programa anterior para imprimir Solo las pizzas que tienen “pepperoni” (muestre al menos 2 tipos)

4. Cree una función constructor de pizzas y utilícela para crear 3 variantes de cada uno de los 5 tipos de pizza anteriores. Los tipos serán Regular, Large y extra large.

Imprima todos los tipos de pizzas.

5.

Functions map and reduce are standard functions from traditional functional programming that achieved broader recognition as a result of Google’s MapReduce method for processing and generating large data sets. While map, reduce, and a number of related functions are provided in many JavaScript implementations, map and reduce can also be defined relatively simply in JavaScript as follows:

```
function map (f, inarray) {  
  var out = [];  
  for(var i = 0; i < inarray.length; i++) {  
    out.push( f(inarray[i]) )  
  }  
  return out;  
}  
  
function reduce (f, inarray) {  
  if(inarray.length <= 1) return;  
  if(inarray.length == 2) return f(inarray[0], inarray[1]);  
  r = inarray[0];  
  for(var li = 1; li < inarray.length; li++){  
    r = f(r, inarray[li]);  
  }  
  return r;  
}
```

Function map(f, inarray) returns an array constructed by applying f to every element if inarray.

Function reduce(f, inarray) applies the function f of two arguments to elements in the list, from left to right, until it reduces the list to a single element.

For example:

```
js> map( function(x){return x+1}, [1,2,3,4,5])  
2,3,4,5,6  
js> reduce( function(x,y){return x+y}, [1,2,3,4,5])  
15  
js> reduce( function(x,y){return x*y}, [1,2,3,4,5])  
120
```

These functions can be combined in various ways. For each of the following questions, you may use a JavaScript implementation to test your answer yourself, but turn your solution in as part of a written description for manual grading.

(a) Explain how to use map and reduce to compute the sum of the first five squares, in one line. (The sum of the first three squares is $1^2 + 2^2 + 3^2$.)

(b) Explain how to use map and reduce to count the number of positive numbers in an array of numbers.

(c) Explain how to use map and/or reduce to “flatten” an array of arrays of numbers, such as `[[1,2],[3,4],[5,6],[7,8,9]]`, to an array of numbers.

(Hint: Look for built-in JavaScript concatenation functions.)