



CSS - ALGORITMO 3

Curso "Full Stack Python – Talento Digital 2020"

2020

Gunther Andrés Toro Guillón

Práctica de Algoritmos N°3

Noviembre, 2020

Contenido

1	PARTE I	3
1.1	Ejercicio 1	3
1.2	Ejercicio 2	3
1.3	Ejercicio 3	4
1.4	Ejercicio 4	4
1.5	Ejercicio 5	5
1.6	Ejercicio 6	5
1.7	Ejercicio 7	6
1.8	Ejercicio 8	6
2	PARTE II	7
2.1	Analiza los valores de un array y obtén el promedio (average) de esos valores.	7
2.2	Crea un array con todos los enteros impares (odd integers) entre 1 y 255 (inclusive) ...	8
2.3	Cuadra cada valor con un array dado, obteniendo el mismo array con valores cambiados.	8

1 PARTE I

Predice el output de los siguientes códigos. No ejecutes el código directo en la consola o JSbin, hazlo manualmente sin computador. Saber hacer esto en papel es muy importante. Una vez que hayas hecho lo anterior, ejecuta el código en tu consola/JSbin y compara este resultado con tu predicción manual. Si tu predicción es diferente a los resultados del computador, intenta descifrar por qué y entender cómo funciona el código.

Queremos que tengas el hábito de usar el diagrama T y hacer esto en papel. Esto no solo te ayudará en los ejercicios de pizarra, sino que a ser un mejor desarrollador y solucionador de problemas. No te saltes esta actividad.

1.1 Ejercicio 1

```
function a(x,y) {
  return 5;
}
console.log(a(5,5))
```

$a(x,y)$
 $a(5,5) = 5$

consola.
5

1.2 Ejercicio 2

```
function a(x,y) {
  z = [];
  z.push(x);
  z.push(y);
  z.push(5);
  console.log(z);
  return z;
}
b = a(2,2)
console.log(b);
console.log(a(6,8));
```

VAR.
 $b = a(2,2)$
 $b = [2, 2, 5]$

$a(x,y)$
 $z = []$
 $z = [2]$
 $z = [2, 2]$
 $z = [2, 2, 5]$
 $a(6,8)$
 $z = []$
 $z = [6]$
 $z = [6, 8]$
 $z = [6, 8, 5]$

consola.
[2, 2, 5]
[2, 2, 5]
[6, 8, 5]
[6, 8, 5]

1.3 Ejercicio 3

```
function a(x){
  z = [];
  z.push(x);
  z.pop();
  z.push(x);
  z.push(x);
  return z;
}
y = a(2);
y.push(5);
console.log(y);
```

VAR	a(x)	CONSOLA
y = a(2)	z = []	[2 2 5]
y = [2 2]	z = [2]	
y = [2 2 5]	z = []	
	z = [2]	
	z = [2 2]	

1.4 Ejercicio 4

```
function a(x){
  if(x[0] < x[1]) {
    return true;
  }
  else {
    return false;
  }
}
b = a([2,3,4,5])
console.log(b);
```

VAR	a(x)	CONSOLA
b = a([2,3,4,5])	a([2,3,4,5]) = True	TRUE
b = True		

1.5 Ejercicio 5

```
function a(x){
  for(var i=0; i<x.length; i++){
    if(x[i] > 0){
      x[i] = "Coding";
    }
  }
  return x;
}
console.log(a([1,2,3,4]))
```

VAR
a([1,2,3,4])

a(x)
i = 0
x[0] = Coding
i = 1
x[1] = Coding
i = 2
x[2] = Coding
i = 3
x[3] = Coding
i = 4

Console.
[Coding Coding Coding Coding]

1.6 Ejercicio 6

```
function a(x){
  for(var i=0; i<x.length; i++){
    if(x[i] > 5){
      x[i] = "Coding";
    }
    else if(x[i] < 0){
      x[i] = "Dojo";
    }
  }
  return x;
}
console.log(a([5,7,-1,4]))
```

VAR
a([5,7,-1,4])

a(x)
i = 0
x[0] = 5
i = 1
x[1] = 7
x[1] = Coding
i = 2
x[2] = -1
x[2] = Dojo
i = 3
x[3] = 4
i = 4

Console.
[~~Undefined~~ 5 Coding Dojo ~~Undefined~~ 4]

1.7 Ejercicio 7

```
function a(x){  
  if(x[0] > x[1]) {  
    return x[1];  
  }  
  return 10;  
}  
b = a([5,10])  
console.log(b);
```

Varz a(x) Console

b = a([5,10]) a([5,10]) 10

b = 10

1.8 Ejercicio 8

```
function sum(x){  
  sum = 0;  
  for(var i=0; i<x.length; i++){  
    sum = sum + x[i];  
    console.log(sum);  
  }  
  return sum;  
}
```

Console } No se ejecuta
- NADA.

2 PARTE II

2.1 Analiza los valores de un array y obtén el promedio (average) de esos valores.

```
function printAverage(x){  
  sum = 0;  
  // your code here  
}  
y = printAverage([1,2,3]);  
console.log(y); // should log 2  
  
y = printAverage([2,5,8]);  
console.log(y); // should log 5
```

```
function printAverage (x){  
  Sum = 0  
  for (var i=0; i < x.length; i++){  
    Sum = Sum + x[i];  
  }  
  Sum = Sum / x.length;  
  Return Sum.  
}  
  
Y = print Average ([1, 2, 3]);  
console.log (Y);  
  
Y = printAverage ([2, 5, 8]);  
console.log (Y);
```

2.2 Crea un array con todos los enteros impares (odd integers) entre 1 y 255 (inclusive)

```
function returnOddArray(){
  // your code here
}
y = returnOddArray();
console.log(y); // should log [1,3,5,...,253,255]
```

```
function returnOddArray(){
  x = [];
  var y = 0;
  for (var i = 0; i < 255; i = i + 2){
    x[y] = i + 1;
    y = y + 1;
  }
  return x
}
Y = returnOddArray();
console.log(Y);
```

2.3 Cuadra cada valor con un array dado, obteniendo el mismo array con valores cambiados.

```
function squareValue(x){
  // your code here
  return x;
}
y = squareValue([1,2,3]);
console.log(y); // should log [1,4,9]

y = squareValue([2,5,8]);
console.log(y); // should log [4,25,64]
```

```
function squareValue(x){
  for (var i = 0; i < x.length; i++){
    x[i] = x[i] * x[i];
  }
  return x
}
Y = squareValue([1,2,3]);
console.log(Y);
Y = squareValue([2,5,8]);
console.log(Y);
```