Fundamento1

|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <title>Función Básica 1</title> |
|  | <script type="text/javascript"> |
|  | //Ejercicio 1 |
|  | 1.- Obtén del 1 al 255 - Escribe una función que devuelve un array con todos los número del 1 al 255. |
|  | 2.- Consigue pares hasta 1000 - Escribe una función que entregue la suma de todos los número pares del 1 al 1000 - Puedes usar un operador de módulo para este ejercicio. |
|  | 3.- Suma impares hasta 5000 - Escribe una función que devuelva la suma de todos los números impares entre 1 y 5000 (ej: 1+3+5+...+4997+4999). |
|  | 4.- Itera un arrayEscribe una función que devuelva la suma de todos los valores dentro de un array (ej: [1,2,5] returns 8. [-5,2,5,12] returns 14). |
|  | 5.-Encuentra el mayor (max) Dado un array con múltiples valores, escribe una función que devuelva el número mayor (ej: para [-3,3,5,7] el número mayor (max) es 7). |
|  | 6.-Encuentra el promedio (avg) Dado un array con múltiples valores, escribe una función que devuelva el promedio de los valores (ej: para [1,3,5,7,20] el promedio es 7.2). |
|  | 7.- Array de impares Escribe una funci[on que devuelva un array de todos los números impares entre 1 y 50 (ej: [1,3,5, …, 47,49]). Pista: Usa el método ‘push’. |
|  | 8.- Mayor que Y - Dado un valor Y, escribe una función que toma un array y devuelve los valores mayores que Y. Por ejemplo, si arr = [1,3,5,7] y Y = 3, tu función devolverá 2 (hay 2 números en el array mayores que 3, esto son 5 y 7). |
|  | 9.- Cuadrados Dado un array con múltiples valores, escribe una función que reemplace cada valor por el cuadrado del mismo valor (ej: [1,5,10,-2] será [1,25,100,4]). |
|  | 10.- Negativos - Dado un array con múltiples valores, escribe una función que reemplace cualquier número negativo dentro del array por 0. Cuando el programa esté listo, el array no debiera contener números negativos (ej: [1,5,10,-2] se convertirá en [1,5,10,0]). |
|  | 11.- Max/Min/Avg - Dado un array con múltiples valores, escribe una función que devuelva un nuevo array que solo contenga el valor mayor (max), menor (min) y promedio (avg) del array original (ej: [1,5,10,-2] devolverá [10,-2,3.5]). |
|  | 12.-Intercambia Valores -Escribe una función que intercambie el primer y el último valor de cualquier array. La extensión mínima predeterminada del array es 2 (ej: [1,5,10,-2] será [-2,5,10,1]). |
|  | 13.-De Número a String Escribe una función que tome un array de números y reemplace cualquier valor negativo por el string ‘Dojo’. Por ejemplo, dado el array = [-1,-3,2], tu función devolverá [‘Dojo’,‘Dojo’,2]. |
|  |  |
|  | //Resultados |
|  |  |
|  | //Resultado Ejercicio 1 |
|  | function array() { |
|  | // body... |
|  | }() { |
|  | z=[]; |
|  | for (var i = 1; i < 256; i++) { |
|  | z.push[i] |
|  | } |
|  | return z; |
|  | } |
|  | console.log.(array()); |
|  |  |
|  | //Resultado Ejercicio 2 |
|  | function pares( ){ |
|  | var sum=0; |
|  | for (var i = 1; i <=1000; i+=2) { |
|  | sum=sum+i; |
|  | } |
|  | return sum; |
|  | } |
|  | console.log(pares()); |
|  |  |
|  | //Resultado Ejercicio 3 |
|  | function impares( ){ |
|  | var sum=0; |
|  | for (var i = 1; i <=5000; i+=2) { |
|  | sum=sum+i; |
|  | } |
|  | return sum; |
|  | } |
|  | console.log(impares()); |
|  |  |
|  | //Resultado Ejercicio 4 |
|  | function arrayEscribe(x) { |
|  | var sum=0; |
|  | for (var i = 0; i < x.length; i++) { |
|  | sum=sum+x[i]; |
|  | } |
|  | return sum; |
|  | } |
|  | a=[1,2,5]; |
|  | console.log(arrayEscribe(a)) |
|  |  |
|  | //Resultado Ejercicio 5 |
|  | function max(x){ |
|  | var mayor=x[0]; |
|  | for (var i = 0; i < a.length; i++) { |
|  | if(mayor<x[i]){ |
|  | mayor=x[i]; |
|  | } |
|  | } |
|  | return mayor; |
|  | } |
|  | a=[-3,3,5,7] |
|  | console.log(max(a)) |
|  |  |
|  | //Resultado Ejercicio 6 |
|  | function avg(x){ |
|  | sum=0; |
|  | for (var i = 0; i < x.length; i++) { |
|  | sum=sum+x[i]; |
|  | } |
|  | return(sum/x.length); |
|  | } |
|  | y=avg([1,3,5,7,20]) |
|  | console.log(y) |
|  |  |
|  | //Resultado Ejercicio 7 |
|  | function ttt(){ |
|  | x=[ ]; |
|  | for (var i = 1; i <=50; i+=2) { |
|  | x.push(i); |
|  | } |
|  | return(x); |
|  | } |
|  | y= ttt(); |
|  | console.log(y); |
|  |  |
|  | //Resultado Ejercicio 8 |
|  | function max(a,y){ |
|  | M=[]; |
|  |  |
|  | for (var i = 0; i < a.length; i++) { |
|  | if(y< a[i]){ |
|  | M.push (a[i]); |
|  | } |
|  | } |
|  | return (M); |
|  | } |
|  | a=[1,3,5,7]; |
|  | y=3; |
|  | console.log(max(a,y).length); |
|  |  |
|  | //Resultado Ejercicio 9 |
|  | function a(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(x[i]>0){ |
|  | x[i]=x[i]\*x[i] |
|  | } |
|  | } |
|  | return (x); |
|  | } |
|  | console.log (a([1,5,10,-2])) |
|  |  |
|  | //Resultado Ejercicio 10 |
|  | function a(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(x[i]<0){ |
|  | x[i]=0; |
|  | } |
|  | } |
|  | return (x); |
|  | } |
|  | console.log (a([1,5,10,-2])) |
|  |  |
|  | //Resultado Ejercicio 11 |
|  | function a (array){ |
|  | var max=array[0]; |
|  | var min=array[0]; |
|  | var prom=0; |
|  | for (var i=0; i<array.length; i++){ |
|  | if(array[i]>max){ |
|  | max=array[i]; |
|  | } |
|  | if(array[i]<min){ |
|  | min=array[i]; |
|  | } |
|  | prom=prom+array[i]; |
|  | } |
|  | prom=prom/array.length; |
|  | var res=[max,min,prom]; |
|  | return res |
|  | } |
|  | array=a([1,5,10,-2]); |
|  | console.log(array); |
|  |  |
|  | //Resultado Ejercicio 12 |
|  |  |
|  | function reverse(x){ |
|  | var aux=x[0]; |
|  | x[0]=x[x.length-1]; |
|  | x[x.length-1]=aux; |
|  | return x; |
|  | } |
|  | a=[1,5,10,-2] |
|  | console.log (reverse(a)) |
|  |  |
|  | //Resultado Ejercicio 13 |
|  | function neg(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(x[i]<0){ |
|  | x[i]="Dojo" |
|  | } |
|  | } |
|  | return (x); |
|  | } |
|  | z=neg([-1,-3,2]) |
| Funndamentos2 | console.log (z) |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <title>Función Básica 2</title> |
|  | <script type="text/javascript"> |
|  | //Ejercicios |
|  |  |
|  | //Tamaño Grande - Dado un array, escribe una función que cambie todos los números positivos en él, por el string “big”. Ejemplo: grande([-1,3,5,-5]) devuelve [-1, “grande”, “grande”, -5]. |
|  | //Resultado Ejercicio 1 |
|  | function reneg(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(x[i]>0){ |
|  | x[i]="Big" |
|  | } |
|  | } |
|  | return (x); |
|  | } |
|  | a=reneg([-1,3,5,-5]) |
|  | console.log (a) |
|  |  |
|  | //console |
|  | [-1, "Big", "Big", -5] |
|  |  |
|  | //Imprime (print) el menor, devuelve (return) el mayor - Crea una función que tome un array de números. La función debería imprimir (print) el valor menor del array, y devolver (return) el mayor. |
|  | //Resultado Ejercicio 2 |
|  | function b(x){ |
|  | var max=x[0]; |
|  | var min=x[0]; |
|  |  |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(max < x[i]){ |
|  | max=x[i]; |
|  | } |
|  | else if (min>x[i]){ |
|  | min=x[i]; |
|  | } |
|  | } |
|  | console.log (min) |
|  | return (max); |
|  | } |
|  | a=[-1,3,6,7]; |
|  |  |
|  | console.log (b(a)) |
|  |  |
|  | //Imprime (print) uno, devuelve (return) otro- Crea una función para un array de números. La función debería imprimir (print) el penúltimo valor y devolver (return) el primer valor impar. |
|  | //Resultado Ejercicio 3 |
|  | function returnPrint (x){ |
|  | console.log (x[x.length-2]); |
|  | for (var i = 0; i < x.length; i++) { |
|  | if (x[i]%2!==0){ |
|  | return x[i]; |
|  | } |
|  | } |
|  | } |
|  |  |
|  | a=[2,4,-3,8,-5,-10]; |
|  | console.log (returnPrint(a)) |
|  |  |
|  | //Doble Visión - Dado un array (similar a decir ‘tomado un array’), crea una función que devuelva un nuevo array donde cada valor se duplique. Entonces, doble([1,2,3]) debiera devolver [2, 4, 6] sin cambiar el array original. |
|  | //Resultado Ejercicio 4 |
|  | function doble(x){ |
|  | b=[]; |
|  | for (var i = 0; i < x.length; i++) { |
|  | b.push(x[i]\*2); |
|  | } |
|  | return b; |
|  | } |
|  |  |
|  | a=[1,2,3]; |
|  | console.log(doble (a)) |
|  |  |
|  | //Contar Positivos - Dado un array de números, crea una función para reemplazar el último valor con el número de valores positivos encontrados en el array. Ejemplo, contarPositivos([-1,1,1,1]) cambia el array original y devuelve [-1,1,1,3]. |
|  | //Resultado Ejercicio 5 |
|  | var cont=0; |
|  | function positivo(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if(x[i]>0){ |
|  | cont=cont+1; |
|  | } |
|  | } |
|  | x[x.length-1]=cont; |
|  | return x; |
|  | } |
|  | a=[-1,1,1,1]; |
|  | console.log(positivo(a)); |
|  |  |
|  |  |
|  | //Pares e Impares - Crea una función que acepte un array. Cada vez que ese array tenga 3 valores impares seguidos, imprime (print) “¡Qué imparcial!”, y cada vez que tenga 3 pares seguidos, imprime (print) “¡Es para bien!”. |
|  | //Resultado Ejercicio 6 |
|  |  |
|  | function par(x){ |
|  |  |
|  | for (var i = 0; i < x.length-2; i++) { |
|  |  |
|  | var valor1=x[i]; |
|  | var valor2=x[i+1]; |
|  | var valor3=x[i+2]; |
|  |  |
|  | if (valor1%2!=0 && valor2%2!=0 && valor3%2!=0){ |
|  | console.log ("¡Qué imparcial"); |
|  |  |
|  | } |
|  | if (valor1%2==0 && valor2%2==0 && valor3%2==0){ |
|  | console.log ("¡Es para bien!"); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | } |
|  | a=[2,-4,8,-11,43,3,-2,-5]; |
|  | par(a); |
|  |  |
|  | //Incrementa los Segundos - Dado un array de números arr, agrega 1 a cualquier otro elemento, específicamente a aquellos cuyo índice es impar (arr[1], arr[3], arr[5], etc). Luego, console.log cada valor del array y devuelve arr. |
|  | //Resultado Ejercicio 7 |
|  |  |
|  | function segundos(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | if (i%2!=0){ |
|  | x[i]= x[i]+1; |
|  | } |
|  | console.log (x[i]); |
|  | } |
|  | } |
|  | arr=[1,4,5,6,7]; |
|  | segundos(arr); |
|  |  |
|  |  |
|  | //Longitudes previas - Pasado un array (similar a decir ‘tomado un array’ o ‘dado un array’) que contiene strings, reemplaza cada string con un número de acuerdo cantidad de letras (longitud) del string anterior. Por ejemplo, longitudesPrevias([“programar”,“dojo”, “genial”]) debería devolver [“programar”,9, 4]. Pista: ¿For loops solo puede ir hacia adelante? |
|  | //Resultado Ejercicio 8 |
|  |  |
|  | function strings(x){ |
|  |  |
|  | for (var i =x.length-1; i>0; i--) { |
|  | x[i]=x[i-1].length; |
|  | } |
|  | return x; |
|  | } |
|  |  |
|  | a=["programar","dojo","genio"] |
|  | console.log (strings(a)) |
|  |  |
|  | //Agrega Siete - Construye una función que acepte un array. Devuelve un nuevo array con todos los valores del original, pero sumando 7 a cada uno. No alteres el array original. Por ejemplo, agregaSiete([1,2,3) debería devolver [8,9,10] en un nuevo array. |
|  | //Resultado Ejercicio 9 |
|  |  |
|  | function Siete(x){ |
|  | for (var i = 0; i < x.length; i++) { |
|  | x[i]=x[i]+7 |
|  | } |
|  | return x; |
|  | } |
|  |  |
|  | a=[1,2,3]; |
|  | console.log (Siete(a)) |
|  |  |
|  | //Array Inverso - Dado un array, escribe una función que invierte sus valores en el lugar. Ejemplo: invertir([3,1,6,4,2)) devuelve el mismo array pero con sus valores al revés, es decir [2,4,6,1,3]. Haz esto sin crear un array temporal vacío. (Pista: necesitarás intercambiar (swap) valores). |
|  | //Resultado Ejercicio 10 |
|  |  |
|  | function swap(x){ |
|  | var temp; |
|  | for (let i = 0; i < (x.length/2); i++) { |
|  | temp=x[i]; |
|  | x[i]=x[x.length-(i+1)] |
|  | x[x.length-(i+1)]=temp; |
|  | } |
|  | return x; |
|  | } |
|  | a=[3,1,6,4,2]; |
|  | console.log (swap(a)) |
|  |  |
|  |  |
|  | // Perspectiva: Negativa - Dado un array crear y devuelve uno nuevo que contenga todos los valores del array original, pero negativos (no simplemente multiplicando por -1). Dado [1,-3,5], devuelve [-1,-3,-5]. |
|  | //Resultado Ejercicio 11 |
|  |  |
|  | function neg(x){ |
|  | for (var i = 0; i < x.length; i++) |
|  | if (x[i]>0){ |
|  | x[i]=x[-i] |
|  | } |
|  | return x; |
|  | } |
|  |  |
|  | a=[1,-3,5]; |
|  | console.log (neg(a)) |
|  |  |
|  |  |
|  | //Siempre hambriento - Crea una función que acepte un array e imprima (print) “yummy” cada vez que alguno de los valores sea “comida”. Si ningún valor es “comida”, entonces imprime “tengo hambre” una vez. |
|  | //Resultado Ejercicio 12 |
|  |  |
|  | function hambre(x){ |
|  | var cont=0; |
|  | for (i = 0; i < x.length; i++) { |
|  | if (x[i]=="comida") { |
|  | console.log ("yummy"); |
|  | cont=cont+1; |
|  | } |
|  | } |
|  | if (cont===0){ |
|  | console.log ("tengo hambre"); |
|  | } |
|  | } |
|  |  |
|  | a=["lucho","pide","pizza","comida"] |
|  | hambre(a); |
|  |  |
|  |  |
|  | //Cambiar hacia el centro - Dado un array, cambia el primer y último valor, el tercero con el ante penútimo, etc. Ejemplo: cambiaHaciaElCentro([true, 42, “Ada”, 2, “pizza”]) cambia el array a [“pizza¨, 42, “Ada”, true]. cambiaHaciaElCentro([1,2,3,4,5,6]) cambia el array a [6,2,4,3,5,1]. No es necesario devolver (return) el array esta vez. |
|  | //Resultado Ejercicio 13 |
|  |  |
|  | function swap (x){ |
|  | var temp; |
|  | for (let i = 0; i < x.length/2; i++) { |
|  | temp=x[i]; |
|  | x[i]=x[x.length-(i+1)]; |
|  | x[x.length-(i+1)]=temp; |
|  | } |
|  | return x; |
|  | } |
|  | a=[true, 42, “Ada”, 2, “pizza”] |
|  | swap(a); |
|  |  |
|  |  |
|  |  |
|  |  |
|  | //Escala el Array - Dado un array arr y un número num, multiplica todos los valores en el array arr por el número num, y devuelve el array arr modificado. Por ejemplo, escalaArray([1,2,3], 3] debería devolver [3,6,9]. |
|  | //Resultado Ejercicio 14 |
|  |  |
|  | function |

Funsion básica

|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  |  |
|  |  |
|  | </head> |
|  | <body> |
|  | <!--Funcion 1--> |
|  | <script> |
|  | function a(){ |
|  | return 35 |
|  | } |
|  | console.log(a()) |
|  | </script> |
|  | 35 |
|  | <!--Funcion 2--> |
|  | <script> |
|  | function a(){ |
|  | return 4; |
|  | } |
|  | console.log(a()+a()); |
|  | </script> |
|  | 8 |
|  | <!--Funcion 3--> |
|  | <script> |
|  | function a(b){ |
|  | return b; |
|  | } |
|  | console.log(a(2)+a(4)) |
|  | </script> |
|  | 6 |
|  | <!--Funcion 4--> |
|  | <script> |
|  | function a(b){ |
|  | console.log(b); |
|  | return b\*3; |
|  | } |
|  | console.log(a(3)) |
|  | </script> |
|  | 3 |
|  | 9 |
|  | <!--Funcion 5--> |
|  | <script> |
|  | function a(b){ |
|  | return b\*4; |
|  | console.log(b); |
|  | } |
|  | console.log(a(10)) |
|  | </script> |
|  | 10 |
|  | 40 |
|  | <!--Funcion 6--> |
|  | <script> |
|  | function a(b){ |
|  | if(b<10) { |
|  | return 2; |
|  | } |
|  | else { |
|  | return 4; |
|  | } |
|  | console.log(b); |
|  | } |
|  | console.log(a(15)); |
|  | </script> |
|  | 15 |
|  | 4 |
|  | <!--Funcion 7--> |
|  | <script> |
|  | function a(b,c){ |
|  | return b\*c; |
|  | } |
|  | console.log(10,3); |
|  | console.log( a(3,10) ) |
|  | </script> |
|  | 10 3 |
|  | 30 |
|  | <!--Funcion 8--> |
|  | <script> |
|  | function a(b){ |
|  | for(var i=0; i<10;i++){ |
|  | console.log(i) |
|  | } |
|  | return i |
|  | } |
|  | console.log(3); |
|  | console.log(4); |
|  | </script> |
|  | 3 |
|  | 4 |
|  | <!--Funcion 9--> |
|  | <script> |
|  |  |
|  | function a(){ |
|  | for(var i=0; i<10; i++){ |
|  | i = i +2; |
|  | console.log(i) |
|  | } |
|  | } |
|  | a() |
|  | </script> |
|  | 2 |
|  | 5 |
|  | 8 |
|  | 11 |
|  | <!--Funcion 10--> |
|  | <script> |
|  | function a(b,c) { |
|  | for(var i=b; i<c; i++) { |
|  | console.log(i) |
|  | } |
|  | return b\*c |
|  | } |
|  | a(0,10); |
|  | console.log(a(0,10)); |
|  | </script> |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |
|  | 5 |
|  | 6 |
|  | 7 |
|  | 8 |
|  | 9 |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |
|  | 5 |
|  | 6 |
|  | 7 |
|  | 8 |
|  | 9 |
|  | 0 |
|  | <!--Funcion 11--> |
|  | <script> |
|  | function a(){ |
|  | for(var i=0; i<10; i++){ |
|  | for(var j=0; j<10; j++){ |
|  | console.log(j); |
|  | } |
|  | console.log(i); |
|  | } |
|  | } |
|  | </script> |
|  | <!--Funcion 12--> |
|  | <script> |
|  | function a(){ |
|  | for(var i=0; i<10; i++){ |
|  | for(var j=0; j<10; j++){ |
|  | console.log(i,j); |
|  | } |
|  | console.log(j,i); |
|  | } |
|  | } |
|  | </script> |
|  | <!--Funcion 13--> |
|  | <script> |
|  | var z = 10; |
|  | function a(){ |
|  | var z = 15; |
|  | console.log(z); |
|  | } |
|  | console.log(z); |
|  | </script> |
|  | 10 |
|  | <!--Funcion 14--> |
|  | <script> |
|  | var z = 10; |
|  | function a(){ |
|  | var z=15; |
|  | console.log(z); |
|  | } |
|  | a(); |
|  | console.log(z); |
|  | </script> |
|  | 15 |
|  | 10 |
|  | <!--funcion 15--> |
|  | <script> |
|  | var z = 10; |
|  | function a(){ |
|  | var z = 15; |
|  | console.log(z); |
|  | return z; |
|  | } |
|  | z = a(); |
|  | console.log(z); |
|  | </script> |
|  | 15 |
|  | 15 |
|  | </body> |
|  | </html> |