



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: arreglos

Versión: 1.0

Descripción:

Función números

numeros

Tipo de variable: Alfanumérico

Código:

```
/**
 * primer ejercicio con arreglo
 * autor: ingrid
 * fecha: 08 de mayo del 2024
 */
let arreglo = [];
let numeroDatos = [];
let iteracion;
let datosArreglos = "";

arreglo = [2,3,4,5,6,7,8,9];

numeroDatos = arreglo.length;
console.log("valores iniciales: "+ arreglo)
console.log("numero de datos: "+ numeroDatos)

arreglo.push(10);
arreglo.push(11);
arreglo.push(12);

numeroDatos = arreglo.length;

console.log("valores iniciales "+ arreglo)
console.log("numero de datos "+ numeroDatos)

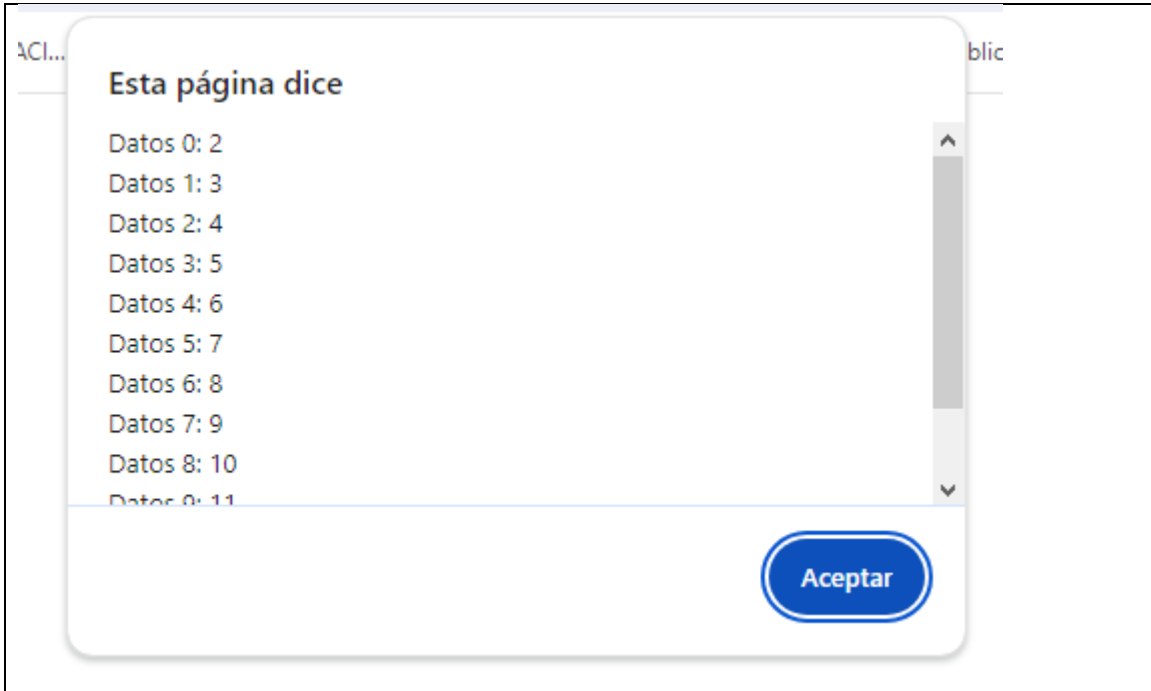
for(iteracion=0; iteracion<numeroDatos; iteracion++){
    console.log("Datos "+ iteracion +": "+ arreglo[iteracion]);
    datosArreglos+="Datos "+ iteracion +": "+ arreglo[iteracion]+"\n";
}
alert(datosArreglos);
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS



Nombre de la función: factura		Versión: 2.0
Descripción: Función factura		
factura	Tipo de variable: numerico	
Código:		
<pre>* ejercicios de arreglos * fecha: 15 de mayo * autor: ingrid medina */ let factura=[]; let valorALProducto; let iteracion; let TotalPagoProducto; let totalPago =[]; factura=[{codigo:1, nombreProducto: 'malteada', cantidad: 3, valorUnidad: 12000}, {codigo:2, nombreProducto: 'picada', cantidad: 3, valorUnidad: 25000}, {codigo:3, nombreProducto: 'hamburguesa mixta', cantidad: 4, valorUnidad: 14000}, {codigo:4, nombreProducto: 'churrasco', cantidad: 1, valorUnidad: 25000}, {codigo:5, nombreProducto: 'gaseosa', cantidad: 5, valorUnidad: 5000}, {codigo:6, nombreProducto: 'limonada', cantidad: 5, valorUnidad: 6000},]; valorALProducto = factura[3].cantidad *factura[3].valorUnidad; console.log("Total a pagar: " +valorALProducto); console.log(factura[3].nombreProducto); for(iteracion=0; iteracion < factura.length; iteracion++){ TotalPagoProducto = factura[iteracion].cantidad *factura[iteracion].valorUnidad; totalPago.push((nombre: factura[iteracion].nombreProducto, cantidad: factura[iteracion], precio: factura[iteracion].valorUnidad, total: TotalPagoProducto) } console.log(totalPago)</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

```
Total a pagar: 25000 factura.js:22
churrasco factura.js:23
▼ Array(6) factura.js:29
  ▼ 0:
    ▶ cantidad: {codigo: 1, nombreProducto: 'malteada', cantidad: 3, valorUnidad: 12000}
      nombre: "malteada"
      precio: 12000
      total: 36000
    ▶ [[Prototype]]: Object
    ▶ 1: {nombre: 'picada', cantidad: {...}, precio: 25000, total: 75000}
    ▶ 2: {nombre: 'hamburguesa mixta', cantidad: {...}, precio: 14000, total: 56000}
    ▶ 3: {nombre: 'churrasco', cantidad: {...}, precio: 25000, total: 25000}
    ▶ 4: {nombre: 'gaseosa', cantidad: {...}, precio: 5000, total: 25000}
    ▶ 5: {nombre: 'limonada', cantidad: {...}, precio: 6000, total: 30000}
    length: 6
    ▶ [[Prototype]]: Array(0)
```

Nombre de la función: nomina		Versión: 3.0
Descripción: Función nomina de varias personas		
nomina	Tipo de variable: String	
Código:		
<pre>/** * ejercicios realizar una nomina * autor: ingrid medina * fecha: 15 de mayo */ let nomina= []; let pagoNomina =[]; let iteracion; let pago; let salarioM = 1300000; let subTransporte; let bonoN; let retencion; let reten; let pension; let arl; let salud; let salario; nomina=[{codigo:1, cedula: 1080291867, nombres: 'ingrid yulissa', apellidos: 'medina esquivel', edad: 17, estrato: 3, valorDia: 120000, diasTrabajados: 80}, {codigo:2, cedula: 1077723426, nombres: 'brayan santiago', apellidos: 'guerrero mendez', edad: 17, estrato: 2, valorDia: 350000, diasTrabajados: 60}, {codigo:3, cedula: 1077724121, nombres: 'camilo andres', apellidos: 'losada ramirez', edad: 17, estrato: 1, valorDia: 710000, diasTrabajados: 20}, {codigo:4, cedula: 1075225114, nombres: 'yerson stiven', apellidos: 'rubiano', edad: 18, estrato: 1, valorDia: 43000, diasTrabajados: 30}, {codigo:5, cedula: 1075793094, nombres: 'karol natalia', apellidos: 'osorio poveda', edad: 17, estrato: 1, valorDia: 410000, diasTrabajados: 70}, {codigo:6, cedula: 1077029186, nombres: 'marcos', apellidos: 'rojas alvarez', edad: 17, estrato: 5, valorDia: 320000, diasTrabajados: 50}, {codigo:7, cedula: 1080967325, nombres: 'juan felipe', apellidos: 'gomez ruiz', edad: 19, estrato: 2, valorDia: 1200000, diasTrabajados: 20}, {codigo:8, cedula: 1012798043, nombres: 'jesus david', apellidos: 'fierro rivera', edad: 19, estrato: 4, valorDia: 1240000, diasTrabajados: 70}, {codigo:9, cedula: 1003810172, nombres: 'jhoan camilo', apellidos: 'charry perez', edad: 21, estrato: 2, valorDia: 120000, diasTrabajados: 40}, {codigo:10, cedula: 1647892847, nombres: 'juan manuel ', apellidos: 'gutierrez fierro', edad: 17, estrato: 3, valorDia: 120000, diasTrabajados: 10},]</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

```
function subTra(pago) {
  if (pago < 2 * salarioM) {
    subTransporte = 114000;
  } else {
    subTransporte = 0;
  }
  return subTransporte;
}

function bono(pago, estrato) {
  if ((pago < salarioM && estrato == 1) || pago < salarioM && estrato == 2) {
    bonoN = 100000;
  } else {
    bonoN = 0;
  }
  return bonoN;
}

function saludN(pago) {
  salud = pago * 0.12;
  return salud;
}

function pensionN(pago) {
  pension = pago * 0.16;
  return pension;
}

function arlN(pago) {
  arl = pago * 0.052;
  return arl;
}

function retencionN(pago, estrato) {
  if (pago > 8 * salarioM && nomina[iteracion].estrato == 6 ) {
    retencion = 0.05;
  } else if (pago > salarioM * 6) {
    retencion = 0.04;
  } else if (pago > salarioM * 4) {
    retencion = 0.03;
  } else {
    retencion = 0;
  }
  return retencion;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

```
for (iteracion = 0; iteracion < nomina.length; iteracion++) {  
    pago = nomina[iteracion].diasTrabajados * nomina[iteracion].valorDia;  
  
    subTra(pago);  
    bono(pago, nomina[iteracion].estrato);  
    saludN(pago);  
    pensionN(pago);  
    arlN(pago);  
    retencionN(pago, nomina[iteracion].estrato);  
  
    reten = pago * retencion  
    salario = ( pago + subTransporte + bonoN) - (salud + pension + arl + reten)  
    pagoNomina.push({  
        cedula: nomina[iteracion].cedula,  
        nombres: nomina[iteracion].nombres,  
        apellidos: nomina[iteracion].apellidos,  
        edad: nomina[iteracion].edad,  
        estrato: nomina[iteracion].estrato,  
        valorDia: nomina[iteracion].valorDia,  
        diasTrabajados: nomina[iteracion].diasTrabajados,  
        SalarioBruto: pago,  
        subTransporte: subTransporte,  
        bonoN: bonoN,  
        retencion: retencion,  
        salud: salud,  
        pension: pension,  
        arl: arl,  
    });  
}  
console.table(pagoNomina)
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

```
▼ Array(10) 1
  ▼ 0:
    SalarioBruto: 9600000
    apellidos: "medina esquivel"
    arl: 499200
    bonoN: 0
    cedula: 1080291867
    diasTrabajados: 80
    edad: 17
    estrato: 3
    nombres: "ingrid yulissa"
    pension: 1536000
    retencion: 0.04
    salud: 1152000
    subTransporte: 0
    valorDia: 120000
    ► [[Prototype]]: Object
  ► 1: {cedula: 1077723426, nombres: 'brayan santiago', apellidos: 'guerrero mendez', edad: 17, estrato: 2, ...}
  ► 2: {cedula: 1077724121, nombres: 'camilo andres', apellidos: 'losada ramirez', edad: 17, estrato: 1, ...}
  ► 3: {cedula: 1075225114, nombres: 'yerson stiven', apellidos: 'rubiano', edad: 18, estrato: 1, ...}
  ► 4: {cedula: 1075793094, nombres: 'karol natalia', apellidos: 'osorio poveda', edad: 17, estrato: 1, ...}
  ► 5: {cedula: 1077029186, nombres: 'marcos', apellidos: 'rojas alvarez', edad: 17, estrato: 5, ...}
  ► 6: {cedula: 1080967325, nombres: 'juan felipe', apellidos: 'gomez ruiz', edad: 19, estrato: 2, ...}
  ► 7: {cedula: 1012798043, nombres: 'jesus david', apellidos: 'fierro rivera', edad: 19, estrato: 4, ...}
  ► 8: {cedula: 1003810172, nombres: 'jhoan camilo', apellidos: 'charry perez', edad: 21, estrato: 2, ...}
  ► 9: {cedula: 1647892847, nombres: 'juan manuel ', apellidos: 'gutierrez fierro', edad: 17, estrato: 3, ...}
    length: 10
  ► [[Prototype]]: Array(0)
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: bingo

Versión: 4.0

Descripción:

Función realizar un bingo con 3 x

bingo

Tipo de variable: numérico

Código:

```
/**
 * ejercicio del bingo
 * autor: ingrid
 * fecha: 08 de mayo del 2024
 */
let bingo = []
let iteracion1;
let iteracion2;
let contador = 0;
let tabla;
let letraB = []
let letraI = []
let letraN = []
let letraG = []
let letraO = []
let cantidadImpares = 0;
let cantidadPares = 0;

for(iteracion1 = 0; iteracion1<5; iteracion1++){
  let interno = []
  for(iteracion2 = 0; iteracion2<5; iteracion2++){
    contador = contador + 1;
    tabla = contador *3;
    interno.push(tabla);
  }
  bingo.push(interno);
}

// letra B
for(iteracion1= 0; iteracion1<5; iteracion1++){
  letraB.push(bingo[iteracion1][0])
}
console.log("letra B: "+letraB);
// letra i
for(iteracion1= 0; iteracion1<5; iteracion1++){
  letraI.push(bingo[iteracion1][1])
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

```
console.log("letra I: "+letraI);
// letra N
for(iteracion1= 0; iteracion1<5; iteracion1++){
    letraN.push(bingo[iteracion1][2])
}
console.log("letra N: "+letraN);
// letra G
for(iteracion1= 0; iteracion1<5; iteracion1++){
    letraG.push(bingo[iteracion1][3])
}
console.log("letra G: "+letraG);
// letra o
for(iteracion1= 0; iteracion1<5; iteracion1++){
    letraO.push(bingo[iteracion1][4])
}
console.log("letra O: "+letraO);

console.log(bingo);

console.log("X,Grande");
console.log(bingo[0][0],bingo[1][1],bingo[2][2],bingo[3][3],bingo[4][4],bingo[0][4],bingo[1][3],bingo[2][2],bingo[3][1],bingo[4][0])

console.log("X1");
console.log(bingo[0][1],bingo[1][2],bingo[2][3],bingo[0][3],bingo[1][2],bingo[2][1]);

console.log("X2");
console.log(bingo[2][0],bingo[3][1],bingo[4][2],bingo[2][2],bingo[3][1],bingo[4][0]);

console.log("X3");
console.log(bingo[2][2],bingo[3][3],bingo[4][4],bingo[2][4],bingo[3][3],bingo[4][2]);

for(iteracion1 = 0; iteracion1<5; iteracion1++){
    for(iteracion2 = 0; iteracion2<5; iteracion2++){
        if(bingo[iteracion1][iteracion2]%2==0){
            cantidadPares = cantidadPares+1
        }
        else {
            cantidadImpares = cantidadImpares+1
        }
    }
}

console.log("cantidad de numeros Impares: "+ cantidadImpares);
console.log("cantidad de numeros Pares: "+ cantidadPares);
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

letra B: 3,18,33,48,63	bingo.js:32
letra I: 6,21,36,51,66	bingo.js:37
letra N: 9,24,39,54,69	bingo.js:42
letra G: 12,27,42,57,72	bingo.js:47
letra O: 15,30,45,60,75	bingo.js:52
▼ Array(5) i ▶ 0: (5) [3, 6, 9, 12, 15] ▶ 1: (5) [18, 21, 24, 27, 30] ▶ 2: (5) [33, 36, 39, 42, 45] ▶ 3: (5) [48, 51, 54, 57, 60] ▶ 4: (5) [63, 66, 69, 72, 75] length: 5 ▶ [[Prototype]]: Array(0)	bingo.js:54
X, Grande	bingo.js:56
3 21 39 57 75 15 27 39 51 63	bingo.js:57
X1	bingo.js:60
6 24 42 12 24 36	bingo.js:61
X2	bingo.js:63
33 51 69 39 51 63	bingo.js:64
X3	bingo.js:66
39 57 75 45 57 69	bingo.js:67
cantidad de numeros Impares: 13	bingo.js:80
cantidad de numeros Pares: 12	bingo.js:81