MANUAL DE ARREGLOS

INSTRUCTOR:

ANDRES MORENO COLLAZOS

APRENDIZ:

DARWIN CUELLAR MORALES

**Primer ejercicio**

|  |  |  |
| --- | --- | --- |
| **Nombre del arreglos: arreglo[]** | | **Versión: 1.0** |
| **Descripción:**  Arreglo que almacena 5 números y ciclo para imprimir cada uno | | |
| arreglo  numero1  numero2  numero3  cantidadDatos  iterar | Tipo de variable: arreglo(array) int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:          let arreglo = [];          arreglo = [1,2,3,4,5];          let numero1 = arreglo[0]//1          let numero2 = arreglo[2]//3          let numero3 = arreglo[4]//5          let cantidadDatos = arreglo.length;          let iterar;          for(iterar = 0; iterar < cantidadDatos; iterar++){              console.log(arreglo[iterar])          }  Pantalla | | |

**Segundo ejercicio**

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: arreglos, pares, impares** | | **Versión: 1.0** |
| **Descripción:**  Ciclo que genere numeros del 1 al 10 y almacenarlos en un arreglo, luego mostrar numero pares e impares almacenados en otros arreglos | | |
| arreglo  pares  impares  iterar | Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: int | |
| Código:          let arreglo = [];          let pares = [];          let impares = [];          let iterar;          for(iterar = 1; iterar <= 10; iterar++){              arreglo.push(iterar)              if(iterar %2 == 0){                  pares.push(iterar)              }              else{                  impares.push(iterar)              }          }          console.log("Numeros: " + arreglo)          console.log("Numeros pares: " + pares)          console.log("Numeros impares: " + impares)  Pantalla | | |

**Tercer ejercicio tabla 5**

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: tabla5, pares, impares** | | **Versión: 1.0** |
| **Descripción:**  Ciclo para la tabla del 5 y que muestre que numeros son pares e impares, luego que muestre los numeros que están en x en la matriz | | |
| tabla5  pares  impares  equis  mult  fila  colum  paresSuma  imparesSuma  iterar  iterar2 | Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:          let tabla5 = []          let pares = []          let impares = []          let equis = []          let mult = 1          let paresSuma          let imparesSuma          for(let fila = 0; fila < 5; fila++){              tabla5[fila] = []              for(let colum = 0; colum < 5; colum++){                  tabla5[fila][colum] = mult \* 5                  mult ++                    if(tabla5[fila][colum] %2 == 0){                      pares.push(tabla5[fila][colum])                  }                  else{                      impares.push(tabla5[fila][colum])                  }                  if(fila===colum || fila+colum===4){                      equis += tabla5[fila][colum] + "\t"                  }                  else{                      equis += "\t"                  }                  paresSuma = 0;                  for(let iterar = 0; iterar < pares.length; iterar++){                      paresSuma += pares[iterar]                    imparesSuma = 0                  for(let iterar2 = 0; iterar2 < impares.length; iterar2++){                      imparesSuma += impares[iterar2]                  }              }          }          equis += "\n"      }          console.log("Tabla del 5: ", tabla5)          console.log("Pares de la tabla del 5: "+ pares)          console.log("impares de la tabla del 5: "+ impares)          console.log("Suma de los pares : "+ paresSuma)          console.log("suma de los impares: "+ imparesSuma)          console.log("equis de la tabla : " +"\n"+ equis)  Pantalla | | |

**Tercer ejercicio tabla 9**

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: tabla9, pares, impares** | | **Versión: 1.0** |
| **Descripción:**  Ciclo para la tabla del 9 y que muestre que numeros son pares e impares, luego que muestre los numero que están en x en la matriz | | |
| Tabla9  pares  impares  mult  fila  colum  paresSuma  imparesSuma  iterar  iterar2 | Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:          let tabla9 = []          let pares = []          let impares = []          let equis = []          let mult = 1          let paresSuma          let imparesSuma          for(let fila = 0; fila < 5; fila++){              tabla9[fila] = []              for(let colum = 0; colum < 5; colum++){                  tabla9[fila][colum] = mult \* 9                  mult ++                    if(tabla9[fila][colum] %2 == 0){                      pares.push(tabla9[fila][colum])                  }                  else{                      impares.push(tabla9[fila][colum])                  }                  if(fila===colum || fila+colum===4){                      equis += tabla9[fila][colum] + "\t"                  }                  else{                      equis += "\t"                  }                  paresSuma = 0;                  for(let iterar = 0; iterar < pares.length; iterar++){                      paresSuma += pares[iterar]                    imparesSuma = 0                  for(let iterar2 = 0; iterar2 < impares.length; iterar2++){                      imparesSuma += impares[iterar2]                  }              }          }          equis += "\n"      }          console.log("Tabla del 9: ", tabla9)          console.log("Pares de la tabla del 9: "+ pares)          console.log("impares de la tabla del 9: "+ impares)          console.log("Suma de los pares : "+ paresSuma)          console.log("suma de los impares: "+ imparesSuma)          console.log("Equis de la tabla " +"\n"+ equis)  Pantalla | | |

**Cuarto ejercicio**

|  |  |  |
| --- | --- | --- |
| **Nombre del arreglos: tabla2, letraB, letraI, letraN, letraG, letraO,** | | **Versión: 1.0** |
| **Descripción:**  Bingo | | |
| tabla2  letraB  letraI  letraN  letraG  letraO  num  fil  colum | Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: arreglo(array) int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:          let tabla2 = []          let letraB = []          let letraI = []          let letraN = []          let letraG = []          let letraO = []          let num = 1          for(let fil = 0; fil < 5; fil++){              tabla2[fil] = []              for(let colum = 0; colum < 5; colum++){              tabla2[fil][colum] = num \* 2              if(colum === 0) letraB.push(tabla2[fil][colum])                  else if (colum === 1) letraI.push(tabla2[fil][colum])                  else if (colum === 2) letraN.push(tabla2[fil][colum])                  else if (colum === 3) letraG.push(tabla2[fil][colum])                  else if (colum === 4) letraO.push(tabla2[fil][colum])                  num ++              }          }          console.log("Tabla del 2 ", tabla2)          console.log("B= " + letraB)          console.log("I= " + letraI)          console.log("O= " + letraN)          console.log("N= " + letraG)          console.log("O= " + letraO)  Pantalla | | |

**Quinto ejercicio**

|  |  |  |
| --- | --- | --- |
| **Nombre del arreglos: tienda y listaBusqueda** | | **Versión: 1.0** |
| **Descripción:**  Registro de una tienda | | |
| tienda  nuevoProducto  listaBusqueda | Tipo de variable: arreglo(array) int  Tipo de variable: int  Tipo de variable: arreglo(array) int | |
| Código:      Pantalla | | |

**Sexto ejercicio**

|  |  |  |
| --- | --- | --- |
| **Nombre de las funciones: salarioPer, saludPer, pensionPer, arlPer, retencionPer, subTransportePer totalPagarPer** | | **Versión: 1.0** |
| **Descripción:**  Fnciones para calcular el salario , salud, pensión, etc. de una persona en un registro | | |
| valorDia  diasTrabajados  sueldoPe  salario  pensionPe  arlPe  retencionPe  subTransporte  totalPagarPe | Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:  function salarioPer(pvalorDia,pdiasTrabajados){      let valorDia = pvalorDia      let diasTrabajados = pdiasTrabajados      let sueldoPe      sueldoPe = valorDia \* diasTrabajados      return sueldoPe  }  function saludPer(sueldoPe){      let salario = sueldoPe      let saludPe      saludPe = salario \* 0.12      return saludPe  }  function pensionPer(sueldoPe){      let salario = sueldoPe      let pensionPe      pensionPe = salario \* 0.16;      return pensionPe  }  function arlPer(sueldoPe){      let salario = sueldoPe      let arlPe      arlPe = salario \* 0.052      return arlPe  }  function retencionPer(sueldoPe){      let salario = sueldoPe      let retencionPe      if(salario > 13200000){          retencionPe = salario \* 0.08      }      else if(salario > 9200000 && sueldoPe <= 13200000){          retencionPe = salario \* 0.04      }      else if(salario > 6900000 && sueldoPe <= 9200000){          retencionPe = salario \* 0.02      }      else{          retencionPe = 0      }      return retencionPe  }  function subTransportePer(sueldoPe){      let salario = sueldoPe      let subTransporte      if(salario < 2300000){          subTransporte = 120000      }      else{          subTransporte = 0      }      return subTransporte  }  function totalPagarPer(sueldoPe, psalud, ppension, parl, psubTransporte, pretencion){      let salario = sueldoPe      let saludPe  = psalud      let pensionPe = ppension      let arlPe = parl      let subTransporte = psubTransporte      let retencionPe = pretencion      let totalPagarPe      totalPagarPe = salario + subTransporte - (saludPe + pensionPe + arlPe + retencionPe)      return totalPagarPe  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: registro1, registro2** | | **Versión: 1.0** |
| **Descripción:**  2 arreglos en el que el primero se registra un numero de personas con su nombres, cargos, salarios, etc. Y en el otro en el que se reunirá y se calculara el salario, retención, arl, etc. De esa persona. Variables en el que se guardara el resultado de las funciones ya mencionadas dándole valores del registro | | |
| registro1  registro2  nuevoRegistro  numeroRegistros  iterar  sueldoPe  saludPe  pensionPe  arlPe  subTransPe  retencionPe  totalPagarPe | Tipo de variable: arreglo(array)  Tipo de variable: arreglo(array)  Tipo de variable: arreglo(array)  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int  Tipo de variable: int | |
| Código:      let registro1= []      let registro2 = []      let nuevoRegistro      let numeroRegistros      let iterar      let sueldoPe      let saludPe      let pensionPe      let arlPe      let subTransPe      let retencionPe      let totalPagarPe | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: registro1, nuevoRegistro** | | **Versión: 1.0** |
| **Descripción:**  Arreglo en el que se almacenan un registro de 10 personas y otro arreglo para agregar mas registros | | |
| registro1  nuevoRegistro  id  nombres  apellidos  cargo  valorDia  disaTrabajados | Tipo de variable: arreglo(array)  Tipo de variable: arreglo(array)  Tipo de propiedad: int  Tipo de propiedad: string  Tipo de propiedad: string  Tipo de propiedad: string  Tipo de propiedad: int  Tipo de propiedad: int | |
| Código: | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de los arreglos: registro2, nuevoRegistro** | | **Versión: 1.0** |
| **Descripción:**  Ciclo para almacenar el resultado de el salario, pensión,arl,etc. De una persona y registrarlo en un arreglo | | |
| Registro2  numeroRegistros  iterar  salario  subTransporte  salud  pension  arl  retención  totalPagar | Tipo de variable: arreglo(array)  Tipo de variable: int  Tipo de variable: int  Tipo de propiedad: int  Tipo de propiedad: int  Tipo de propiedad: int  Tipo de propiedad: int  Tipo de propiedad: int  Tipo de propiedad: int  Tipo de propiedad: int | |
| Código:    Pantalla: | | |