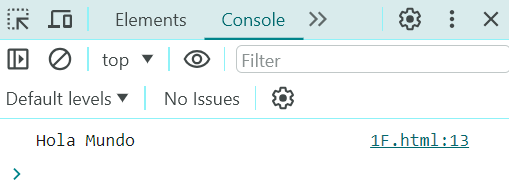
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: saludo(psaludo)** | | **Versión: 1.0** |
| **Descripción:** Función que saluda | | |
| saludo | Tipo de variable: Alfanumérico | |
| Código:  function saludo(psaludo){      let saludar = psaludo;      return saludar;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnumUno, pnumDos)** | | **Versión: 1.0** |
| **Descripción:** Función que suma dos números | | |
| sumar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  function suma(pnumUno,pnumDos){      let sumar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      sumar = numUno + numDos;      return sumar;  } | | |

Consola:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: rest(pnumUno, pnumDos)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| restar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  function rest(pnumUno,pnumDos){      let restar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      restar = numUno - numDos;      return restar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnumUno, pnumDos)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| multiplicar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  function mult(pnumUno,pnumDos){      let multiplicar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      multiplicar = numUno \* numDos;      return multiplicar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: div(pnumUno, pnumDos)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| dividir | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  function div(pnumUno,pnumDos){      let dividir;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      dividir = numUno / numDos;      return dividir;  } | | |

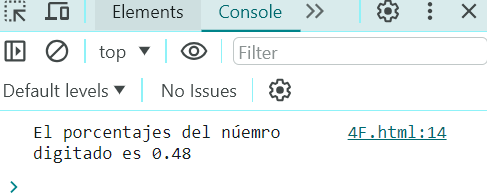
Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

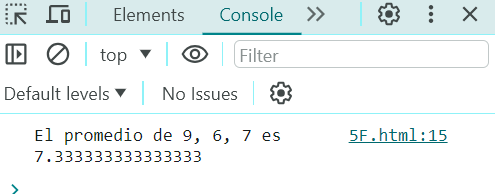
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: por(ppor)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el porcentaje de un número | | |
| porc | Tipo de variable: Int | |
| rpor | Tipo de variable: Float | |
| Código:  function por(ppor) {      let porc = ppor;      let rpor;      rpor = porc / 100;      return rpor;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: div(pnumUno, pnumDos, pnumTre)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el promedio de 3 números | | |
| prom | Tipo de variable: Float | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| numTres | Tipo de variable: Int | |
| Código:  function prom(pnumUno,pnumDos,pnumTre) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let prom;      prom = (numUno + numDos + numTre) / 3;      return prom;  } | | |

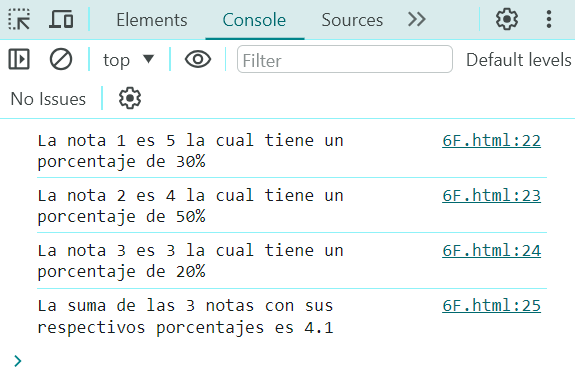
Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: porNot(pnot, ppor)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el promedio de 3 notas | | |
| not | Tipo de variable: Float | |
| porce | Tipo de variable: Float | |
| rnot | Tipo de variable: Float | |
| Código:  function porNot(pnot, pporc) {      let not = pnot;      let porce = pporc;      let rnot;      rnot = not \* porce;      return rnot;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnum1, pnum2, pnum3)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el promedio de 3 notas | | |
| num1 | Tipo de variable: Float | |
| num2 | Tipo de variable: Float | |
| num3 | Tipo de variable: Float | |
| sumar | Tipo de variable: Float | |
| Código:  function suma(pnum1, pnum2, pnum3) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let sumar;      sumar = numUno + numDos + numTre;      return sumar;  } | | |

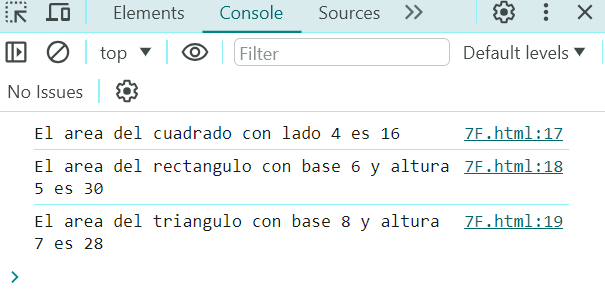
Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: arCuAndRec(paltRec, pbasRec)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el área de un cuadrado, rectángulo y triangulo | | |
| alt | Tipo de variable: Int | |
| bas | Tipo de variable: Int | |
| arCuRec | Tipo de variable: Int | |
| Código:  function arCuAndRec(paltRec, pbasRec) {      let alt = paltRec;      let bas = pbasRec;      let arCuRec;      arCuRec = alt \* bas;      return arCuRec;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: arTri(paltTri, pbasTri)** | | **Versión: 1.0** |
| **Descripción:** Función para sacar el área de un cuadrado, rectángulo y triangulo | | |
| alt | Tipo de variable: Int | |
| bas | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  function arTri(paltTri, pbasTri) {      let alt = paltTri;      let bas = pbasTri;      let ar;      ar = (alt \* bas)/2;      return ar;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suel(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| su | Tipo de variable: Int | |
| Código:  function suel(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let su = diaTra \* valDia;      return   su;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salul(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sa | Tipo de variable: Float | |
| Código:  function salu(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let sa = suel(valDia, diaTra) \* 0.12;      return sa;  } | | |

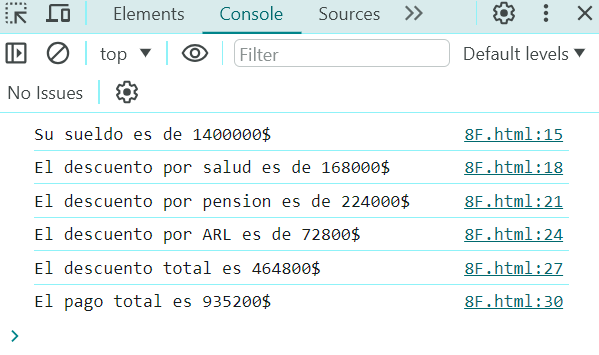
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pens(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| pe | Tipo de variable: Float | |
| Código:  function pens(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let pe = suel(valDia, diaTra) \* 0.16;      return pe;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: Arl(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  function Arl(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let ar = suel(valDia, diaTra) \* 0.052;      return ar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| salud | Tipo de variable: Float | |
| pension | Tipo de variable: Float | |
| arl | Tipo de variable: Float | |
| de | Tipo de variable: Float | |
| Código:  function desc(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let salud = salu(valDia, diaTra);      let pension = pens(valDia, diaTra);      let arl = Arl(valDia, diaTra);      let de;      de = salud + pension + arl;      return de;  } | | |

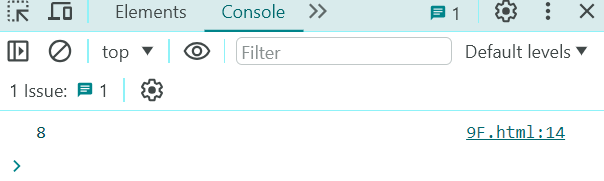
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sueldo | Tipo de variable: Int | |
| descuento | Tipo de variable: Float | |
| total | Tipo de variable: Float | |
| Código:  function to(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let sueldo = suel(valDia, diaTra);      let descuento = desc(valDia, diaTra);      let total;      total = sueldo - descuento;      return total;  } | | |

Consola:



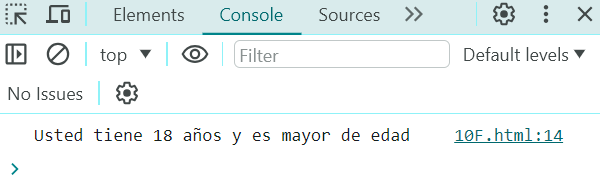
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pnum1, pnum2)** | | **Versión: 1.0** |
| **Descripción:** Función para imprimir que número es mayor | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| Código:  function may(pnum1,pnum2){      let num1 = pnum1;      let num2 = pnum2;      if(num1 == num2){          return "los valores ingresados son iguales";      }      else{          if(num1 > num2){              return num1;          }          else{              return num2;          }      }    } | | |

Consola:



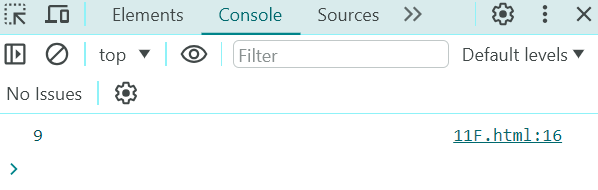
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: ed(pfecNa)** | | **Versión: 1.0** |
| **Descripción:** Función para saber si la persona es mayor de edad o no | | |
| fecNa | Tipo de variable: Date | |
| fecAc | Tipo de variable: Date | |
| eda | Tipo de variable: IntegerToDate | |
| Código:  function ed(pfecNa) {      let fecNa = pfecNa;      let fecAc = 2024;      let eda;      eda = parseInt(fecAc) - parseInt(fecNa);      return eda;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pnum1, pnum2, pnum3)** | | **Versión: 1.0** |
| **Descripción:** Función para imprimir que número es el mayor de 3 números | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| num3 | Tipo de variable: Int | |
| Código:  function may(pnum1, pnum2, pnum3) {      let num1 = pnum1;      let num2 = pnum2;      let num3 = pnum3;      if (parseFloat(num1) == parseFloat(num2) && parseFloat(num1) == parseFloat(num3) && parseFloat(num2) == parseFloat(num3)) {          return "los 3 números digitados son iguales";      }      else{          if (parseFloat(num1)>parseFloat(num2) && parseFloat(num1)>parseFloat(num3)) {              return num1;          }          else{              if (parseFloat(num2)>parseFloat(num1) && parseFloat(num2)>parseFloat(num3)) {                  return num2;              }              else{                  return num3;              }          }      }  } | | |

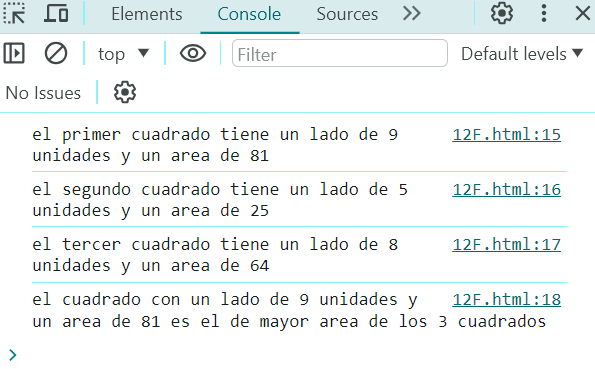
Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: are(plad)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el área de 3 cuadrados e imprimir cuál tiene la mayor área | | |
| lad | Tipo de variable: Int | |
| ar | Tipo de variable: Int | |
| Código:  function are(plad){      let lad = plad;      let ar;      ar = lad \* lad;      return ar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pcu1, pcu2, pcu3)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el área de 3 cuadrados e imprimir cuál tiene la mayor área | | |
| cu1 | Tipo de variable: Int | |
| cu2 | Tipo de variable: Int | |
| cu3 | Tipo de variable: Int | |
| Código:  function may(pcu1, pcu2, pcu3){      let cu1 = pcu1;      let cu2 = pcu2;      let cu3 = pcu3;      if (cu1==cu2 && cu1==cu3 && cu2==cu3) {          return "ERROR SON IGUALES";      }      else{          if (cu1>cu2 && cu1>cu3) {              return cu1;          }          else{              if (cu2>cu1 && cu2>cu3) {                  return cu2;              }              else{                  return cu3;              }          }      }  } | | |

Consola:

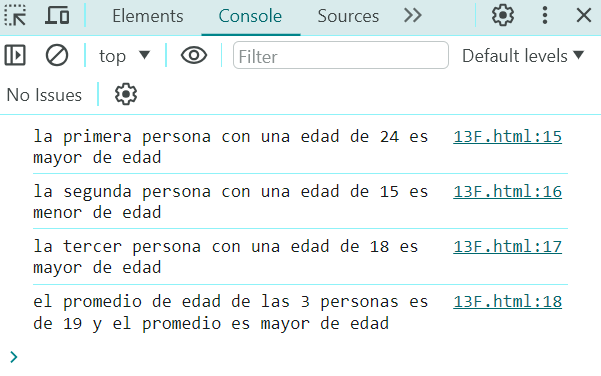


|  |  |  |
| --- | --- | --- |
| **Nombre de la función: eda(pfecNa)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| fecNa | Tipo de variable: Date | |
| fecAc | Tipo de variable: Date | |
| edad | Tipo de variable: IntegerToDate | |
| Código:  function eda(pfecNa){      let fecNa = pfecNa;      let fecAc = 2024;      let edad;      edad = fecAc - fecNa;      return edad;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pro(peda1, peda2, peda3)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| eda1 | Tipo de variable: IntegerToDate | |
| eda2 | Tipo de variable: IntegerToDate | |
| eda3 | Tipo de variable: IntegerToDate | |
| prom | Tipo de variable: Float | |
| Código:  function pro(peda1, peda2, peda3){      let eda1 = peda1;      let eda2 = peda2;      let eda3 = peda3;      let prom;      prom = (eda1+eda2+eda3)/3;      return prom;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(ped)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| ed | Tipo de variable: Float | |
| Código:  function may(ped){      let ed = ped;      if (parseInt(ed)>17) {          return "es mayor de edad";      }      else{          return "es menor de edad";      }  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suel(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| su | Tipo de variable: Int | |
| Código:  function suel(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let su = diaTra \* valDia;      return   su;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salu(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sa | Tipo de variable: Float | |
| Código:  function salu(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let sa = suel(valDia, diaTra) \* 0.12;      return sa;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pens(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| pe | Tipo de variable: Float | |
| Código:  function pens(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let pe = suel(valDia, diaTra) \* 0.16;      return pe;  } | | |
| **Nombre de la función: Arl(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  function Arl(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let ar = suel(valDia, diaTra) \* 0.052;      return ar;  } | | |

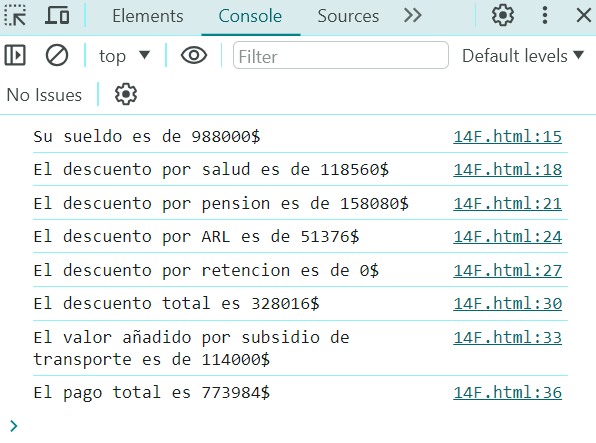
|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| salud | Tipo de variable: Float | |
| pension | Tipo de variable: Float | |
| arl | Tipo de variable: Float | |
| de | Tipo de variable: Float | |
| Código:  function desc(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let salud = salu(valDia, diaTra);      let pension = pens(valDia, diaTra);      let arl = Arl(valDia, diaTra);      let de;      de = salud + pension + arl;      return de;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: subTrans(pvalDia, pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| subTra | Tipo de variable: Int | |
| Código:  function subTrans(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let subTra = 0;      if(suel(valDia, diaTra) < (2\*1300000)){          subTra = subTra + 114000;      }else{          subTra = subTra + 0;      }      return subTra;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: rete(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| reten | Tipo de variable: Float | |
| Código:  function rete(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let reten = 0;      if(suel(valDia, diaTra)>(4\*1300000)){          reten = suel(valDia, diaTra) \* 0.04;      }else{          reten = reten + 0;      }      return reten;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: to(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sueldo | Tipo de variable: Int | |
| subTransporte | Tipo de variable: Int | |
| descuento | Tipo de variable: Float | |
| retencion | Tipo de variable: Float | |
| total | Tipo de variable: Float | |
| Código:  function to(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let sueldo = suel(valDia, diaTra);      let subTransporte = subTrans(valDia, diaTra);      let descuento = desc(valDia, diaTra);      let retencion = rete(valDia, diaTra);      let total;      total = (sueldo + subTransporte) - (descuento + retencion);      return total;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pornot(pnot, pporc)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| not | Tipo de variable: Float | |
| porce | Tipo de variable: Float | |
| rnot | Tipo de variable: Float | |
| Código:  function pornot(pnot, pporc) {      let not = pnot;      let nporce = pporc;      let rnot;      rnot = not \* porce;      return rnot;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnumUno, pnumDos, pnumTre)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| numUno | Tipo de variable: Float | |
| numDos | Tipo de variable: Float | |
| numTre | Tipo de variable: Float | |
| sumar | Tipo de variable: Float | |
| Código:  function suma(pnumUno, pnumDos, pnumTre) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let sumar;      sumar = numUno + numDos + numTre;      return sumar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(psuma)** | | **Versión: 1.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| suma | Tipo de variable: Float | |
| Código:  function may(psuma){      let suma = psuma;      if (suma>4.5) {          return "Su nota final es superior";      }      else{          if (suma<=4.5 && suma>3.5) {              return "Su nota final es buena";          }          else{              if (suma<=3.5 && suma>3) {                  return "Su nota final es media";              }              else{                  return "Su nota final es mala";              }          }      }  } | | |

Consola:

Texto

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salu(pvalDia,pdiaTra)** | | **Versión: 1.0** |
| **Descripción:** Función para contar el número que ingrese el usuario | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: String | |
|  |  | |
| Código:  function cont(pnum){      let num = pnum;      let con = 0;      let res = "";      while(con<num){          con = con+1;          res += con+"\n";      }      return res;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: fact(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer el factorial del número que sea ingresado | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| fac | Tipo de variable: Int | |
| Código:  function fact(pnum){      let num = pnum;      let con = 0;      let fac = 1;      while (con<num) {          con = parseInt(con) + 1;          fac = parseFloat(fac)\*parseFloat(con);      }      return fac;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer la tabla de multiplicar hasta 10 del número ingresado | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| Código:  function mult(pnum){      let num = pnum;      let con = 0;      let res = 0;      let multi = "";      while (con < 10) {          con = con + 1;          res = num \* con;          multi += num+" x "+con+" = "+res+"\n";      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 y decir si el resultado es par p impar | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| Código:  function mult(pnum){      let num = pnum;      let con = 0;      let res = 0;      let multi = "";      while (con < 5) {          con = con + 1;          multi = num \* con;          if(multi % 2 == 0){              multi += res+" es par"+"\n";          }else{              multi += res+" es impar"+"\n";          }      }      return multi;  } | | |

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteConsola:

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: cont(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para contar hasta el número ingresado | | |
| num | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| Código:  function cont(pnum){      let num = pnum;      let res = "";      for(let con = 1; con <= num; con++){          res += con+"\n";      }      return res;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: fact(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer el factorial del número ingresado | | |
| num | Tipo de variable: Int | |
| fac | Tipo de variable: Int | |
| Código:  function fact(pnum){      let num = pnum;      let fac = 1;      for (let con = 1; con <= num; con++) {          fac = parseFloat(fac)\*parseFloat(con);      }      return fac;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado | | |
| num | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| con | Tipo de variable: Int | |
| Código:  function mult(pnum){      let num = pnum;      let multi  = 0;      let res = "";      for(let con = 1; con <= 10; con++){          res = num \* con;          multi += num+" x "+con+" = "+res+"\n";      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| con | Tipo de variable: Int | |
| Código:  function mult(pnum){      let num = pnum;      let res = 1;      let multi = "";      for(let con = 1; con <= 5; con++){          multi = num \* con;          if (multi %2 == 0){              multi += res+" es par\n";          }else{              multi += res+" es impar\n";          }      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| con1 | Tipo de variable: Int | |
| con2 | Tipo de variable: Int | |
| par | Tipo de variable: Int | |
| imp | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| Código:  function mult(pnum1, pnum2){      let num1 = pnum1;      let num2 = pnum2;      let con1 = 0;      let par = 0;      let imp = 0;      let multi;      let res="";      while(con1 < num1){          let con2 = 0          con1 = con1+1;          res += "\n  TABLA DEL "+con1+"\n";          while(con2 < num2){              con2 = con2+1;              multi = con1\*con2;              res += con1+" x "+con2+" = "+multi+"\n";              if(multi %2 == 0){                  par = par+1;                  res = res+"Buzz\n";              }else{                  imp = imp+1                  res = res+"Bass\n";              }          }      }      return res+"\nhay "+par+" pares"+"\n"+"hay "+imp+" impares";  } | | |

Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto, Carta

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| con1 | Tipo de variable: Int | |
| con2 | Tipo de variable: Int | |
| par | Tipo de variable: Int | |
| imp | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| Código:  function mult(pnum1, pnum2){      let num1 = pnum1;      let num2 = pnum2;      let par = 0;      let imp = 0;      let multi;      let res="";      for (let con1 = 1; con1 <= num1; con1++){          res += "\n  TABLA DEL "+con1+"\n";          for (let con2 = 1; con2 <= num2; con2++){              multi = con1\*con2;              res += con1+" x "+con2+" = "+multi+"\n";              if(multi %2 == 0){                  par = par+1;                  res = res+"Buzz\n";              }else{                  imp = imp+1                  res = res+"Bass\n";              }          }      }      return res+"\nhay "+par+" pares"+"\n"+"hay "+imp+" impares";  } | | |

Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto, Carta

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: saludo(psaludo)** | | **Versión: 2.0** |
| **Descripción:** Función que saluda | | |
| saludo | Tipo de variable: Alfanumérico | |
| Código:  const saludo = function(psaludo){      let saludar = psaludo;      return saludar;  } | | |

Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnumUno, pnumDos)** | | **Versión: 2.0** |
| **Descripción:** Función que suma dos números | | |
| sumar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  const suma = function(pnumUno,pnumDos){      let sumar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      sumar = numUno + numDos;      return sumar;  } | | |

Consola:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: rest(pnumUno, pnumDos)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| restar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  const rest = function(pnumUno,pnumDos){      let restar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      restar = numUno - numDos;      return restar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnumUno, pnumDos)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| multiplicar | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  const mult = function(pnumUno,pnumDos){      let multiplicar;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      multiplicar = numUno \* numDos;      return multiplicar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: div(pnumUno, pnumDos)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer la resta, multiplicación y división de dos números | | |
| dividir | Tipo de variable: Int | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| Código:  const div = function(pnumUno,pnumDos){      let dividir;  let numUno;  let numDos;      numUno = pnumUno;      numDos = pnumDos;      dividir = numUno / numDos;      return dividir;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: por(ppor)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el porcentaje de un número | | |
| porc | Tipo de variable: Int | |
| rpor | Tipo de variable: Float | |
| Código:  const por = function(ppor) {      let porc = ppor;      let rpor;      rpor = porc / 100;      return rpor;  } | | |

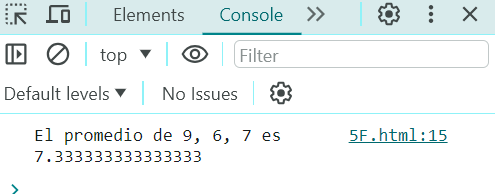
Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: div(pnumUno, pnumDos, pnumTre)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el promedio de 3 números | | |
| prom | Tipo de variable: Float | |
| numUno | Tipo de variable: Int | |
| numDos | Tipo de variable: Int | |
| numTres | Tipo de variable: Int | |
| Código:  const prom = function(pnumUno,pnumDos,pnumTre) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let prom;      prom = (numUno + numDos + numTre) / 3;      return prom;  } | | |

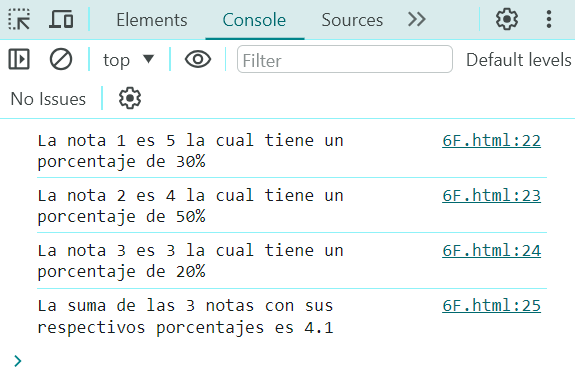
Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: porNot(pnot, ppor)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el promedio de 3 notas | | |
| not | Tipo de variable: Float | |
| porce | Tipo de variable: Float | |
| rnot | Tipo de variable: Float | |
| Código:  const porNot = function(pnot, pporc) {      let not = pnot;      let porce = pporc;      let rnot;      rnot = not \* porce;      return rnot;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnum1, pnum2, pnum3)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el promedio de 3 notas | | |
| num1 | Tipo de variable: Float | |
| num2 | Tipo de variable: Float | |
| num3 | Tipo de variable: Float | |
| sumar | Tipo de variable: Float | |
| Código:  const suma = function(pnum1, pnum2, pnum3) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let sumar;      sumar = numUno + numDos + numTre;      return sumar;  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: arCuAndRec(paltRec, pbasRec)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el área de un cuadrado, rectángulo y triangulo | | |
| alt | Tipo de variable: Int | |
| bas | Tipo de variable: Int | |
| arCuRec | Tipo de variable: Int | |
| Código:  const arCuAndRec = function arCuAndRec(paltRec, pbasRec) {      let alt = paltRec;      let bas = pbasRec;      let arCuRec;      arCuRec = alt \* bas;      return arCuRec;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: arTri(paltTri, pbasTri)** | | **Versión: 2.0** |
| **Descripción:** Función para sacar el área de un cuadrado, rectángulo y triangulo | | |
| alt | Tipo de variable: Int | |
| bas | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  const arTri = function(paltTri, pbasTri) {      let alt = paltTri;      let bas = pbasTri;      let ar;      ar = (alt \* bas)/2;      return ar;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suel(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| su | Tipo de variable: Int | |
| Código:  const suel = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let su = diaTra \* valDia;      return   su;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salul(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sa | Tipo de variable: Float | |
| Código:  const salu = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let sa = suel(valDia, diaTra) \* 0.12;      return sa;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pens(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| pe | Tipo de variable: Float | |
| Código:  const pens = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let pe = suel(valDia, diaTra) \* 0.16;      return pe;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: Arl(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  const Arl = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let ar = suel(valDia, diaTra) \* 0.052;      return ar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| salud | Tipo de variable: Float | |
| pension | Tipo de variable: Float | |
| arl | Tipo de variable: Float | |
| de | Tipo de variable: Float | |
| Código:  const desc = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let salud = salu(valDia, diaTra);      let pension = pens(valDia, diaTra);      let arl = Arl(valDia, diaTra);      let de;      de = salud + pension + arl;      return de;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para obtener el sueldo ganado por un trabajador | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sueldo | Tipo de variable: Int | |
| descuento | Tipo de variable: Float | |
| total | Tipo de variable: Float | |
| Código:  Const to = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let sueldo = suel(valDia, diaTra);      let descuento = desc(valDia, diaTra);      let total;      total = sueldo - descuento;      return total;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pnum1, pnum2)** | | **Versión: 2.0** |
| **Descripción:** Función para imprimir que número es mayor | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| Código:  const may = function(pnum1,pnum2){      let num1 = pnum1;      let num2 = pnum2;      if(num1 == num2){          return "los valores ingresados son iguales";      }      else{          if(num1 > num2){              return num1;          }          else{              return num2;          }      }    } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: ed(pfecNa)** | | **Versión: 2.0** |
| **Descripción:** Función para saber si la persona es mayor de edad o no | | |
| fecNa | Tipo de variable: Date | |
| fecAc | Tipo de variable: Date | |
| eda | Tipo de variable: IntegerToDate | |
| Código:  const ed = function(pfecNa) {      let fecNa = pfecNa;      let fecAc = 2024;      let eda;      eda = parseInt(fecAc) - parseInt(fecNa);      return eda;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pnum1, pnum2, pnum3)** | | **Versión: 2.0** |
| **Descripción:** Función para imprimir que número es el mayor de 3 números | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| num3 | Tipo de variable: Int | |
| Código:  const may = function(pnum1, pnum2, pnum3) {      let num1 = pnum1;      let num2 = pnum2;      let num3 = pnum3;      if (parseFloat(num1) == parseFloat(num2) && parseFloat(num1) == parseFloat(num3) && parseFloat(num2) == parseFloat(num3)) {          return "los 3 números digitados son iguales";      }      else{          if (parseFloat(num1)>parseFloat(num2) && parseFloat(num1)>parseFloat(num3)) {              return num1;          }          else{              if (parseFloat(num2)>parseFloat(num1) && parseFloat(num2)>parseFloat(num3)) {                  return num2;              }              else{                  return num3;              }          }      }  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: are(plad)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el área de 3 cuadrados e imprimir cuál tiene la mayor área | | |
| lad | Tipo de variable: Int | |
| ar | Tipo de variable: Int | |
| Código:  const are = function(plad){      let lad = plad;      let ar;      ar = lad \* lad;      return ar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(pcu1, pcu2, pcu3)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el área de 3 cuadrados e imprimir cuál tiene la mayor área | | |
| cu1 | Tipo de variable: Int | |
| cu2 | Tipo de variable: Int | |
| cu3 | Tipo de variable: Int | |
| Código:  const may = function(pcu1, pcu2, pcu3){      let cu1 = pcu1;      let cu2 = pcu2;      let cu3 = pcu3;      if (cu1==cu2 && cu1==cu3 && cu2==cu3) {          return "ERROR SON IGUALES";      }      else{          if (cu1>cu2 && cu1>cu3) {              return cu1;          }          else{              if (cu2>cu1 && cu2>cu3) {                  return cu2;              }              else{                  return cu3;              }          }      }  } | | |

Consola:

Texto

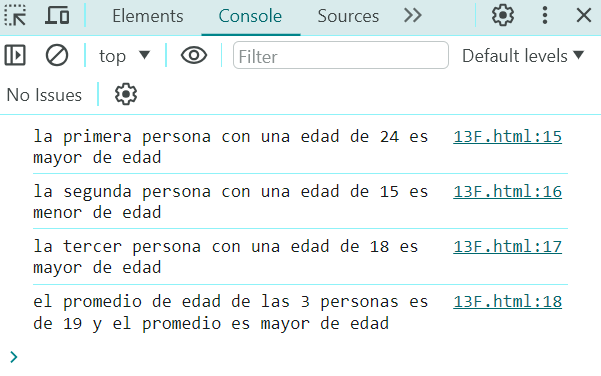
Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: eda(pfecNa)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| fecNa | Tipo de variable: Date | |
| fecAc | Tipo de variable: Date | |
| edad | Tipo de variable: IntegerToDate | |
| Código:  const eda = function(pfecNa){      let fecNa = pfecNa;      let fecAc = 2024;      let edad;      edad = fecAc - fecNa;      return edad;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pro(peda1, peda2, peda3)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| eda1 | Tipo de variable: IntegerToDate | |
| eda2 | Tipo de variable: IntegerToDate | |
| eda3 | Tipo de variable: IntegerToDate | |
| prom | Tipo de variable: Float | |
| Código:  const pro = function(peda1, peda2, peda3){      let eda1 = peda1;      let eda2 = peda2;      let eda3 = peda3;      let prom;      prom = (eda1+eda2+eda3)/3;      return prom;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(ped)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular 3 edades, su promedio y si es mayor de edad o no | | |
| ed | Tipo de variable: Float | |
| Código:  const may = function(ped){      let ed = ped;      if (parseInt(ed)>17) {          return "es mayor de edad";      }      else{          return "es menor de edad";      }  } | | |

Consola:



|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suel(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| su | Tipo de variable: Int | |
| Código:  const suel = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let su = diaTra \* valDia;      return   su;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salu(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sa | Tipo de variable: Float | |
| Código:  const salu = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let sa = suel(valDia, diaTra) \* 0.12;      return sa;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pens(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| pe | Tipo de variable: Float | |
| Código:  const pens = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let pe = suel(valDia, diaTra) \* 0.16;      return pe;  } | | |
| **Nombre de la función: Arl(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| ar | Tipo de variable: Float | |
| Código:  const Arl = function(pvalDia, pdiaTra) {      let valDia = pvalDia;      let diaTra = pdiaTra;      let ar = suel(valDia, diaTra) \* 0.052;      return ar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: desc(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| salud | Tipo de variable: Float | |
| pension | Tipo de variable: Float | |
| arl | Tipo de variable: Float | |
| de | Tipo de variable: Float | |
| Código:  const desc = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let salud = salu(valDia, diaTra);      let pension = pens(valDia, diaTra);      let arl = Arl(valDia, diaTra);      let de;      de = salud + pension + arl;      return de;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: subTrans(pvalDia, pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| subTra | Tipo de variable: Int | |
| Código:  const subTrans = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let subTra = 0;      if(suel(valDia, diaTra) < (2\*1300000)){          subTra = subTra + 114000;      }else{          subTra = subTra + 0;      }      return subTra;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: rete(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| reten | Tipo de variable: Float | |
| Código:  const rete = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let reten = 0;      if(suel(valDia, diaTra)>(4\*1300000)){          reten = suel(valDia, diaTra) \* 0.04;      }else{          reten = reten + 0;      }      return reten;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: to(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular el sueldo de una persona | | |
| valDia | Tipo de variable: Int | |
| diaTra | Tipo de variable: Int | |
| sueldo | Tipo de variable: Int | |
| subTransporte | Tipo de variable: Int | |
| descuento | Tipo de variable: Float | |
| retencion | Tipo de variable: Float | |
| total | Tipo de variable: Float | |
| Código:  const to = function(pvalDia, pdiaTra){      let valDia = pvalDia;      let diaTra = pdiaTra;      let sueldo = suel(valDia, diaTra);      let subTransporte = subTrans(valDia, diaTra);      let descuento = desc(valDia, diaTra);      let retencion = rete(valDia, diaTra);      let total;      total = (sueldo + subTransporte) - (descuento + retencion);      return total;  } | | |

Consola:

Texto

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pornot(pnot, pporc)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| not | Tipo de variable: Float | |
| porce | Tipo de variable: Float | |
| rnot | Tipo de variable: Float | |
| Código:  const pornot = function(pnot, pporc) {      let not = pnot;      let nporce = pporc;      let rnot;      rnot = not \* porce;      return rnot;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: suma(pnumUno, pnumDos, pnumTre)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| numUno | Tipo de variable: Float | |
| numDos | Tipo de variable: Float | |
| numTre | Tipo de variable: Float | |
| sumar | Tipo de variable: Float | |
| Código:  const suma = function(pnumUno, pnumDos, pnumTre) {      let numUno = pnumUno;      let numDos = pnumDos;      let numTre = pnumTre;      let sumar;      sumar = numUno + numDos + numTre;      return sumar;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: may(psuma)** | | **Versión: 2.0** |
| **Descripción:** Función para calcular la nota de una persona | | |
| suma | Tipo de variable: Float | |
| Código:  const may = function(psuma){      let suma = psuma;      if (suma>4.5) {          return "Su nota final es superior";      }      else{          if (suma<=4.5 && suma>3.5) {              return "Su nota final es buena";          }          else{              if (suma<=3.5 && suma>3) {                  return "Su nota final es media";              }              else{                  return "Su nota final es mala";              }          }      }  } | | |

Consola:

Texto

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salu(pvalDia,pdiaTra)** | | **Versión: 2.0** |
| **Descripción:** Función para contar el número que ingrese el usuario | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: String | |
|  |  | |
| Código:  const cont = function(pnum){      let num = pnum;      let con = 0;      let res = "";      while(con<num){          con = con+1;          res += con+"\n";      }      return res;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: fact(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer el factorial del número que sea ingresado | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| fac | Tipo de variable: Int | |
| Código:  const fact = function(pnum){      let num = pnum;      let con = 0;      let fac = 1;      while (con<num) {          con = parseInt(con) + 1;          fac = parseFloat(fac)\*parseFloat(con);      }      return fac;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer la tabla de multiplicar hasta 10 del número ingresado | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| Código:  const mult = function mult(pnum){      let num = pnum;      let con = 0;      let res = 0;      let multi = "";      while (con < 10) {          con = con + 1;          res = num \* con;          multi += num+" x "+con+" = "+res+"\n";      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 y decir si el resultado es par p impar | | |
| num | Tipo de variable: Int | |
| con | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| Código:  const mult = function(pnum){      let num = pnum;      let con = 0;      let res = 0;      let multi = "";      while (con < 5) {          con = con + 1;          multi = num \* con;          if(multi % 2 == 0){              multi += res+" es par"+"\n";          }else{              multi += res+" es impar"+"\n";          }      }      return multi;  } | | |

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteConsola:

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: cont(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para contar hasta el número ingresado | | |
| num | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| con | Tipo de variable: Int | |
| Código:  const cont = function(pnum){      let num = pnum;      let res = "";      for(let con = 1; con <= num; con++){          res += con+"\n";      }      return res;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: fact(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer el factorial del número ingresado | | |
| num | Tipo de variable: Int | |
| fac | Tipo de variable: Int | |
| Código:  const fact = function(pnum){      let num = pnum;      let fac = 1;      for (let con = 1; con <= num; con++) {          fac = parseFloat(fac)\*parseFloat(con);      }      return fac;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 2.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado | | |
| num | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| con | Tipo de variable: Int | |
| Código:  const mult = function(pnum){      let num = pnum;      let multi  = 0;      let res = "";      for(let con = 1; con <= 10; con++){          res = num \* con;          multi += num+" x "+con+" = "+res+"\n";      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num | Tipo de variable: Int | |
| res | Tipo de variable: Int | |
| multi | Tipo de variable: String | |
| con | Tipo de variable: Int | |
| Código:  const mult = function(pnum){      let num = pnum;      let res = 1;      let multi = "";      for(let con = 1; con <= 5; con++){          multi = num \* con;          if (multi %2 == 0){              multi += res+" es par\n";          }else{              multi += res+" es impar\n";          }      }      return multi;  } | | |

Consola:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| con1 | Tipo de variable: Int | |
| con2 | Tipo de variable: Int | |
| par | Tipo de variable: Int | |
| imp | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| Código:  const mult = function(pnum1, pnum2){      let num1 = pnum1;      let num2 = pnum2;      let con1 = 0;      let par = 0;      let imp = 0;      let multi;      let res="";      while(con1 < num1){          let con2 = 0          con1 = con1+1;          res += "\n  TABLA DEL "+con1+"\n";          while(con2 < num2){              con2 = con2+1;              multi = con1\*con2;              res += con1+" x "+con2+" = "+multi+"\n";              if(multi %2 == 0){                  par = par+1;                  res = res+"Buzz\n";              }else{                  imp = imp+1                  res = res+"Bass\n";              }          }      }      return res+"\nhay "+par+" pares"+"\n"+"hay "+imp+" impares";  } | | |

Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto, Carta

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: mult(pnum)** | | **Versión: 1.0** |
| **Descripción:** Función para hacer una tabla de multiplicar hasta 10 del número ingresado e imprimir los pares e impares | | |
| num1 | Tipo de variable: Int | |
| num2 | Tipo de variable: Int | |
| con1 | Tipo de variable: Int | |
| con2 | Tipo de variable: Int | |
| par | Tipo de variable: Int | |
| imp | Tipo de variable: Int | |
| multi | Tipo de variable: Int | |
| res | Tipo de variable: String | |
| Código:  const mult = function(pnum1, pnum2){      let num1 = pnum1;      let num2 = pnum2;      let par = 0;      let imp = 0;      let multi;      let res="";      for (let con1 = 1; con1 <= num1; con1++){          res += "\n  TABLA DEL "+con1+"\n";          for (let con2 = 1; con2 <= num2; con2++){              multi = con1\*con2;              res += con1+" x "+con2+" = "+multi+"\n";              if(multi %2 == 0){                  par = par+1;                  res = res+"Buzz\n";              }else{                  imp = imp+1                  res = res+"Bass\n";              }          }      }      return res+"\nhay "+par+" pares"+"\n"+"hay "+imp+" impares";  } | | |

Consola:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto, Carta

Descripción generada automáticamente