



MANUAL: **FUNCIONES**

JUAN PABLO SAAVEDRA CHAMBO

FUNCION: HOLA MUNDO

Función: saludar

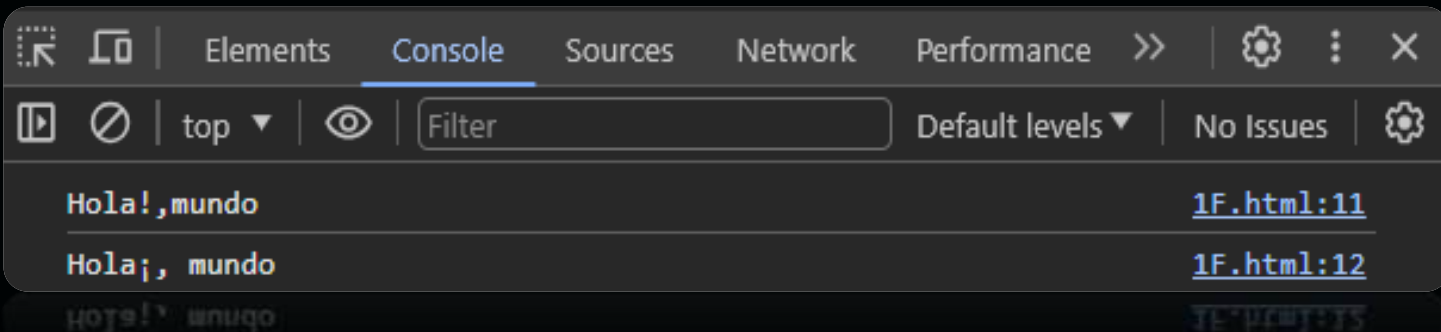
Esta función toma un mensaje como entrada y lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
mensaje	STRING

Función: saludarExp

Al igual que la función saludar, toma un mensaje como entrada y lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
mensaje	STRING



FUNCION: HOLA MUNDO

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <title>Saludo</title>
  <script src="./1F.js"></script>
</head>
<body>
  <script>
    console.log(saludar("¡Hola!
    mundo"));
    console.log(saludarExp("¡Hol
    a!"));
  </script>
</body>
</html>
```

JS

```
function saludar(psaludar) {
  let mensaje = psaludar;
  return mensaje;
}

const saludarExp = function(psaludar) {
  let mensaje = psaludar;
  return mensaje;
};
```

FUNCION: SUMA

Función: resul

Esta función toma dos números como entrada (pnumero1 y pnumero2), los suma y devuelve el resultado.

VARIABLE	TIPO
numero1	INT
numero2	INT
resul	INT

Función: resulExp

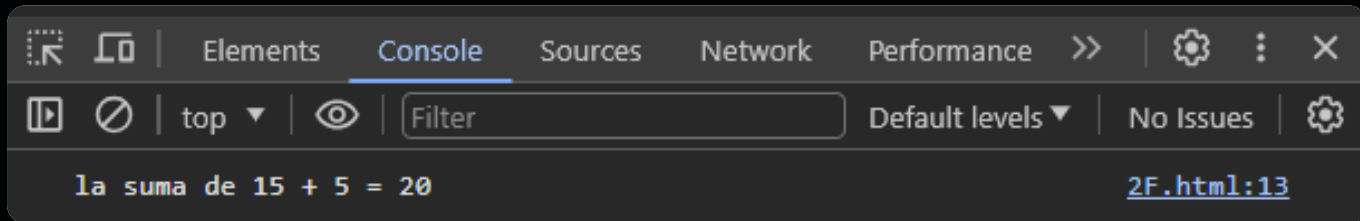
Al igual que la función resul, esta función toma dos números como entrada y los suma y devuelve el resultado. La diferencia principal radica en que esta función está definida utilizando la sintaxis de expresión de función en lugar de la sintaxis de declaración de función.

VARIABLE	TIPO
numero1	INT
numero2	INT
resul	INT

FUNCION: SUMA

DESCRIPCION: El código realiza la suma de dos números predefinidos y muestra el resultado en la consola del navegador utilizando funciones para sumar los números proporcionados como argumentos.

VARIABLE	TIPO
numero1	INT
numero2	INT
resul	INT



FUNCION: SUMA

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>operadores</title>
  <script src="./2F.js"></script>
</head>
<body>
  <script>
    let numero1 = 15
    let numero2 = 5
    console.log("la suma de "+numero1+"
+ "+numero2+" = "+resul(15,5));
  </script>
</body>
</html>
```

JS

```
function resul(pnumero1,pnumero2){
  let numero1=pnumero1;
  let numero2=pnumero2;
  let resul;
  resul=numero1+numero2;
  return resul;
}

const resulExp=function(pnumero1,pnumero2){
  let numero1=pnumero1;
  let numero2=pnumero2;
  let resul;
  resul=numero1+numero2;
  return resul;
}
```

FUNCION: OPERACIONES

Funcion: restaExp

Esta función toma dos números como entrada (num1 y num2), realiza una resta entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
restar	INT

Funcion: opeExp

Esta función toma un operador (poperador) y dos números (num1 y num2) como entrada, realiza la operación especificada por el operador y devuelve el resultado. Admite operaciones de resta, multiplicación y división.

VARIABLE	TIPO
num1	INT
num2	INT
dividir	FLOAT
operador	STRING
opeResultado	FLOAT

Funcion: multiplicacionExp

Esta función toma dos números como entrada (num1 y num2), realiza una multiplicación entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
mutiplicar	INT

Funcion: divisionExp

Esta función toma dos números como entrada (num1 y num2), realiza una división entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
dividir	FLOAT

FUNCION: OPERACIONES

Funcion: resta

Esta función toma dos números como entrada (num1 y num2), realiza una resta entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
restar	INT

Funcion:ope

Esta función toma un operador (poprador) y dos números (num1 y num2) como entrada, realiza la operación especificada por el operador y devuelve el resultado. Admite operaciones de resta, multiplicación y división.

VARIABLE	TIPO
num1	INT
num2	INT
dividir	FLOAT
operador	STRING
opeResultado	FLOAT

Funcion: multiplicacion

Esta función toma dos números como entrada (num1 y num2), realiza una multiplicación entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
mutiplicar	INT

Funcion: division

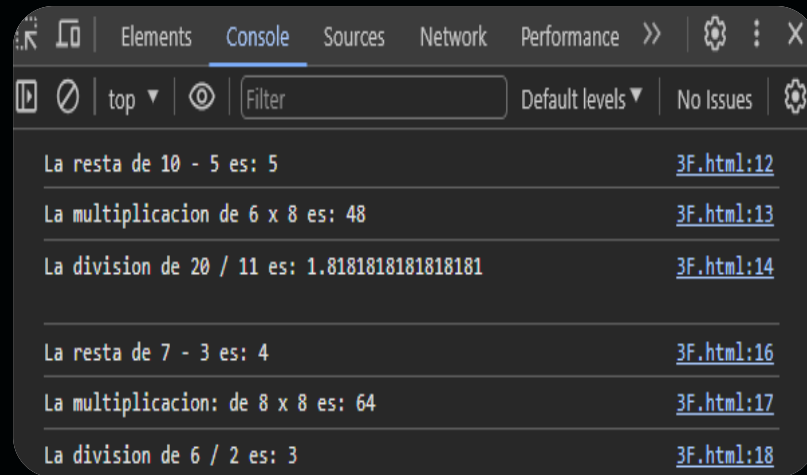
Esta función toma dos números como entrada (num1 y num2), realiza una división entre ellos y devuelve el resultado.

VARIABLE	TIPO
num1	INT
num2	INT
dividir	FLOAT

FUNCION: OPERACIONES

DESCRIPCION: El código realiza operaciones aritméticas (resta, multiplicación y división) entre números predefinidos y muestra los resultados en la consola del navegador utilizando funciones para realizar las operaciones tanto de manera directa como mediante funciones auxiliares.

VARIABLE	TIPO
num1	INT
num2	INT
restar	INT
mutiplicar	INT
dividir	FLOAT
operador	STRING
opeResultado	FLOAT



FUNCION: OPERACIONES

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Operaciones</title>
  <script src="./3F.js"></script>
</head>
<body>
  <script>

    console.log("La resta de 10 - 5 es: "+ope("resta", 10,5));
    console.log("La multiplicacion de 6 x 8 es: "+ope("multiplicacion", 6,8));
    console.log("La division de 20 / 11 es: "+ope("division", 20,11)+"\n"+"");

    console.log("La resta de 7 - 3 es: "+opeExp("resta", 7,3));
    console.log("La multiplicacion: de 8 x 8 es: "+opeExp("multiplicacion", 8,8));
    console.log("La division de 6 / 2 es: "+opeExp("division", 6,2));

  </script>
</body>
</html>
```

FUNCION: OPERACIONES

JS

```
let num1
let num2
```

```
const restaExp=function(num1,num2){
  let restar;
  num1 = num1;
  num2 = num2;
  restar = num1 - num2;
  return restar;
}
```

```
const multiplicacionExp=function(num1,num2){
  let multiplicar;
  num1 = num1;
  num2 = num2;
  multiplicar = num1 * num2;
  return multiplicar;
}
```

```
const divisionExp=function(num1,num2){
  let dividir;
  num1 = num1;
  num2 = num2;
  dividir = num1 / num2;
  return dividir;
}
```

```
const opeExp = function(poperador,num1,num2){
  let operador = poperador;
  num1 = num1;
  num2 = num2;
  if (operador == "resta") {
    opeResultado = restaExp(num1,num2);
  }else if (operador == "multiplicacion") {
    opeResultado = multiplicacionExp(num1,num2);
  }else if (operador == "division") {
    opeResultado = divisionExp(num1,num2);
  }else{
    opeResultado = "Error!!! no se puede operar";
  }
  return opeResultado;
}
```

```
function resta(num1,num2){
  let restar;
  num1 = num1;
  num2 = num2;
  restar=num1-num2;
  return restar;
}
```

JS

FUNCION: OPERACIONES

```
function multiplicacion(num1,num2){
    let multiplicar;
    num1 = num1;
    num2 = num2;
    multiplicar=num1*num2;
    return multiplicar;
}

function division(num1,num2){
    let dividir;
    num1 = num1;
    num2 = num2;
    dividir=num1/num2;
    return dividir;
}

function ope(poperador,num1,num2){
    let operador=poperador;
    num1 = num1;
    num2 = num2;
    if(operador=="resta"){
        resultado=resta(num1,num2);
    }else if(operador=="multiplicacion"){
        resultado=multiplicacion(num1,num2);
    }else if(operador=="division"){
        resultado=division(num1,num2);
    }else{
        resultado="Error";
    }
    return resultado;
}
```

FUNCION:PORCENTAJE

Funcion: por

Esta función toma un número como entrada (pnum), calcula el porcentaje de ese número (dividiéndolo por 100) y devuelve el resultado.

VARIABLE	TIPO
num	INT
porciento	FLOAT

Funcion: porExp

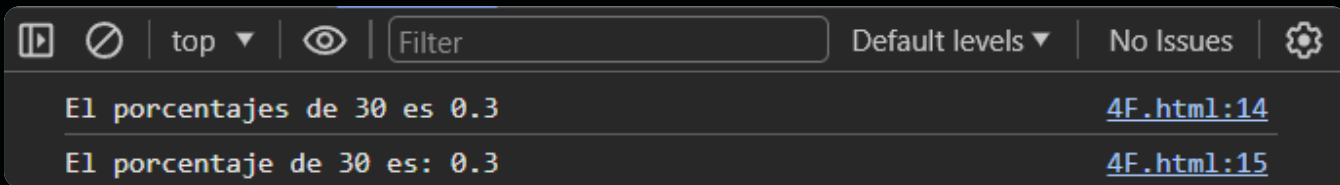
Al igual que la función por, esta función toma un número como entrada (pnum), calcula el porcentaje de ese número (dividiéndolo por 100) y devuelve el resultado.

VARIABLE	TIPO
num	INT
porcentaje	FLOAT

FUNCION:PORCENTAJE

DESCRIPCION: El código sirve para calcular y mostrar porcentajes de un número específico utilizando JavaScript en una página web HTML. Utiliza dos funciones, `por` y `porExp`, para realizar estos cálculos y mostrar los resultados en la consola del navegador. La página web proporciona una estructura básica para mostrar estos resultados de manera visual.

VARIABLE	TIPO
num	INT
porciento	FLOAT
porcentaje	FLOAT



FUNCION:PORCENTAJE

HTML

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
  <script src="./4F.js"></script>
</head>
<body>
  <script>
    let num = 30

    console.log("El porcentajes de 30 es
"+por(30));
    console.log("El porcentaje de 30 es:
"+porExp(30));
  </script>
</body>
</html>
```

JS

```
function por(pnum){
  let num=pnum;
  let porciento;
  porciento=num/100;
  return porciento;
}

const porExp=function(pnum){
  let num=pnum;
  let porcentaje;
  porcentaje=num/100;
  return porcentaje;
}
```

FUNCION:PROMEDIO

Funcion: prom

Esta función toma tres notas como entrada (pnota1, pnota2 y pnota3), calcula la suma de estas notas, calcula el promedio dividiendo la suma entre 3, y devuelve el resultado.

VARIABLE	TIPO
nota1	FLOAT
nota2	FLOAT
nota3	FLOAT
suma	FLOAT
prom	FLOAT

Funcion: promExp





Al igual que la función prom, esta función toma tres notas como entrada (pnota1, pnota2 y pnota3), calcula la suma de estas notas, calcula el promedio dividiendo la suma entre 3, y devuelve el resultado.

VARIABLE	TIPO
nota1	FLOAT
nota2	FLOAT
nota3	FLOAT
suma	FLOAT
prom	FLOAT

FUNCION:PROMEDIO

DESCRIPCION: El código calcula el promedio de tres notas utilizando dos funciones definidas en un archivo JavaScript externo. Las funciones toman tres parámetros (las notas) y devuelven el promedio. El resultado se muestra en la consola del navegador.

VARIABLE	TIPO
nota1	FLOAT
nota2	FLOAT
nota3	FLOAT
suma	FLOAT
prom	FLOAT

  top ▼  Filter Default levels ▼ No Issues 

El promedio de las tres notas es: 3.83333333333333355F.html:12

El promedio de las tres notas es: 3.83333333333333355F.html:14

FUNCION:PROMEDIO

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
  <script src="./5F.js"></script>
</head>
<body>
  <script>

    console.log("El promedio de las tres notas
es: "+prom(2,5,4.5)+"\n"+"");

    console.log("El promedio de las tres notas
es: "+promExp(2,5,4.5));

  </script>
</body>
</html>
```

JS

```
function prom(pnota1,pnota2,pnota3){
  let nota1=pnota1;
  let nota2=pnota2;
  let nota3=pnota3;
  let suma;
  let prom;
  suma=nota1+nota2+nota3;
  prom=suma/3;
  return prom;
}

const promExp=function(pnota1,pnota2,pnota3){
  let nota1=pnota1;
  let nota2=pnota2;
  let nota3=pnota3;
  let suma;
  let prom;
  suma=nota1+nota2+nota3;
  prom=suma/3;
  return prom;
}
```

FUNCION:PORCENTAJES DE TRES NOTAS Y LA SUMA DE LOS PORCENTAJES

Funcion: por

Esta función toma dos parámetros, pnota que representa una nota y ppor que representa un porcentaje. Luego calcula el producto de nota y porce y devuelve el resultado.

VARIABLE	TIPO
nota	FLOAT
porce	FLOAT
respuesta	FLOAT

Funcion: porExp

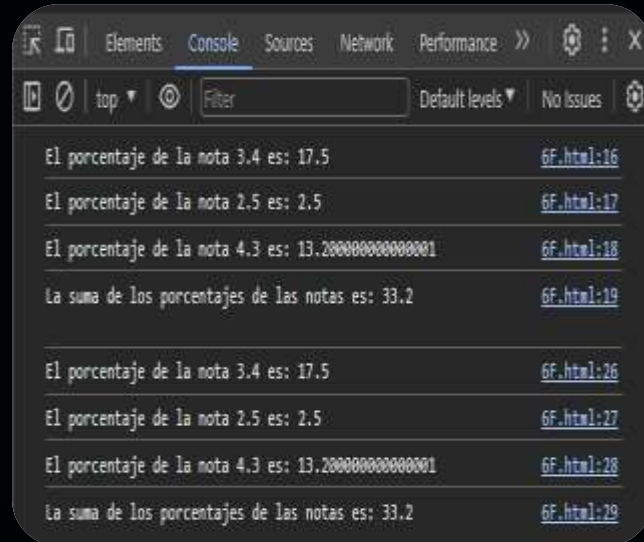
Al igual que la función por, esta función toma dos parámetros, pnota que representa una nota y ppor que representa un porcentaje. Luego calcula el producto de nota y porce y devuelve el resultado.

VARIABLE	TIPO
nota	FLOAT
porce	FLOAT
respuesta	FLOAT

DESCRIPCION: El código calcula el porcentaje de una nota en base a un valor numérico de la nota y un porcentaje dado. Luego, muestra estos cálculos en la consola del navegador web.

VARIABLE	TIPO
por1	FLOAT
por2	FLOAT
por3	FLOAT
porExp1	FLOAT
porExp2	FLOAT
porExp3	FLOAT
nota	FLOAT
porce	FLOAT
respuesta	FLOAT
suma	FLOAT
porFinal	FLOAT
sumaExp	FLOAT
porFinalExp	FLOAT

FUNCION:PORCENTAJES DE TRES NOTAS Y LA SUMA DE LOS PORCENTAJES



The screenshot shows a web browser's developer console with the 'Console' tab selected. It displays five lines of JavaScript output, each with a corresponding file path and line number on the right. The output text is as follows:

```
El porcentaje de la nota 3.4 es: 17.5 6F.html:16
El porcentaje de la nota 2.5 es: 2.5 6F.html:17
El porcentaje de la nota 4.3 es: 13.200000000000001 6F.html:18
La suma de los porcentajes de las notas es: 33.2 6F.html:19
El porcentaje de la nota 3.4 es: 17.5 6F.html:26
El porcentaje de la nota 2.5 es: 2.5 6F.html:27
El porcentaje de la nota 4.3 es: 13.200000000000001 6F.html:28
La suma de los porcentajes de las notas es: 33.2 6F.html:29
```

FUNCION:PORCENTAJES DE TRES NOTAS Y LA SUMA DE LOS PORCENTAJES

HTML

```
</head>
<body>
  <script>
    let por1= por(5,3.5);
    let por2= por(2.5,1);
    let por3= por(4.4,3);
    let suma,porFinal;
    suma=por1+por2+por3;
    console.log("El porcentaje de la nota 3.4 es: "+por1);
    console.log("El porcentaje de la nota 2.5 es: "+por2);
    console.log("El porcentaje de la nota 4.3 es: "+por3);
    console.log("La suma de los porcentajes de las notas es: "+suma+"\n"+"");

    let porExp1= porExp(5,3.5);
    let porExp2= porExp(2.5,1);
    let porExp3= porExp(4.4,3);
    let sumaExp,porFinalExp;
    sumaExp=porExp1+porExp2+porExp3;
    console.log("El porcentaje de la nota 3.4 es: "+porExp1);
    console.log("El porcentaje de la nota 2.5 es: "+porExp2);
    console.log("El porcentaje de la nota 4.3 es: "+porExp3);
    console.log("La suma de los porcentajes de las notas es: "+sumaExp);
  </script>
</body>
</html>
```

FUNCION:PORCENTAJES DE TRES NOTAS Y LA SUMA DE LOS PORCENTAJES

JS

```
function por(pnota,ppor){  
    let nota=pnota;  
    let porce=ppor;  
    let respuesta;  
    respuesta=nota*porce;  
    return respuesta;  
}
```

```
const porExp=function(pnota,ppor){  
    let nota=pnota;  
    let porce=ppor;  
    let respuesta;  
    respuesta=nota*porce;  
    return respuesta;  
}
```

FUNCION:AREA

Funcion: fig

Esta función toma tres parámetros: pFiguras (tipo de figura), pBase (base de la figura) y pAltura (altura de la figura). Luego, según el tipo de figura especificada, calcula el área correspondiente (área de un cuadrado, área de un rectángulo o área de un triángulo) y devuelve el resultado. Si el tipo de figura no es reconocido, devuelve un mensaje indicando que la figura no fue encontrada.

VARIABLE	TIPO
figuras	STRING
base	INT
altura	INT
operacion	INT

Funcion: figExp

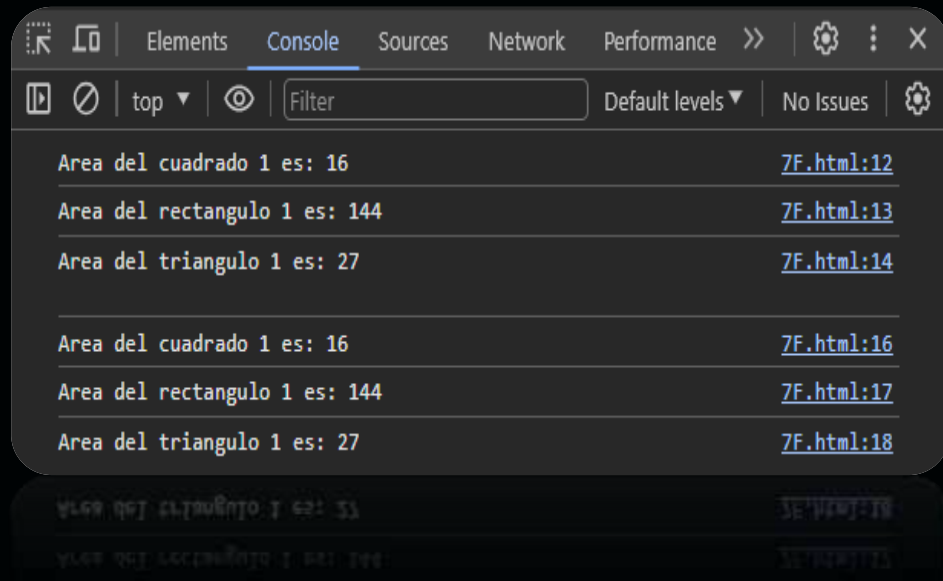
Al igual que la función fig, esta función toma tres parámetros (pFiguras, pBase y pAltura) y calcula el área correspondiente según el tipo de figura especificada (cuadrado, rectangulo o triangulo). La diferencia entre estas funciones radica en la forma en que están definidas: fig utiliza la sintaxis de declaración de función, mientras que figExp utiliza la sintaxis de expresión de función.

VARIABLE	TIPO
figuras	STRING
base	INT
altura	INT
operacion	INT

FUNCION:AREA

DESCRIPCION: El código calcula el promedio de tres notas utilizando dos funciones definidas en un archivo JavaScript externo. Las funciones toman tres parámetros (las notas) y devuelven el promedio. El resultado se muestra en la consola del navegador.

VARIABLE	TIPO
figuras	STRING
base	INT
altura	INT
operacion	INT



FUNCION: AREA

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="./7F.js"></script>
</head>
<body>
  <script>

    console.log("Area del cuadrado 1 es: "+fig("cuadrado",4));
    console.log("Area del rectangulo 1 es: "+fig("rectangulo",16,9));
    console.log("Area del triangulo 1 es: "+fig("triangulo",9,6)+"\n"+"");

    console.log("Area del cuadrado 1 es: "+figExp("cuadrado",4));
    console.log("Area del rectangulo 1 es: "+figExp("rectangulo",16,9));
    console.log("Area del triangulo 1 es: "+figExp("triangulo",9,6));

  </script>
</body>
</html>
```

FUNCION: AREA

JS

```
function fig(pFiguras,pBase,pAltura){
  let figuras=pFiguras;
  let base=pBase;
  let altura=pAltura;
  let operacion;
  if(figuras=="cuadrado"){
    operacion=base*base;
    return operacion;
  }else if(figuras=="rectangulo"){
    operacion=base*altura;
    return operacion;
  }else if(figuras=="triangulo"){
    operacion=(base*altura)/2;
    return operacion;
  }else{
    return "figura no encontrada";
  }
}
```

```
const figExp=function(pFiguras,pBase,pAltura){
  let figuras=pFiguras;
  let base=pBase;
  let altura=pAltura;
  let operacion;
  if(figuras=="cuadrado"){
    operacion=base*base;
    return operacion;
  }else if(figuras=="rectangulo"){
    operacion=base*altura;
    return operacion;
  }else if(figuras=="triangulo"){
    operacion=(base*altura)/2;
    return operacion;
  }else{
    return "figura no encontrada";
  }
}
```

FUNCION: SUELDO

Funcion: subsidios

Esta función calcula el monto del subsidio para un tipo específico de subsidio (salud, pensión, arl), multiplicando el salario diario (pdías * pvdia) por el porcentaje correspondiente. Devuelve el monto del subsidio.

VARIABLE	TIPO
subsidio	STRING
salario	INT
porcentaje	FLOAT

Funcion: pagoTotal

Esta función calcula el pago total descontando los subsidios de salud, pensión y arl del salario total. Utiliza la función subsidios para calcular los subsidios y devuelve el salario total después de los descuentos.

VARIABLE	TIPO
salario	INT
deducible	INT

Funcion: des

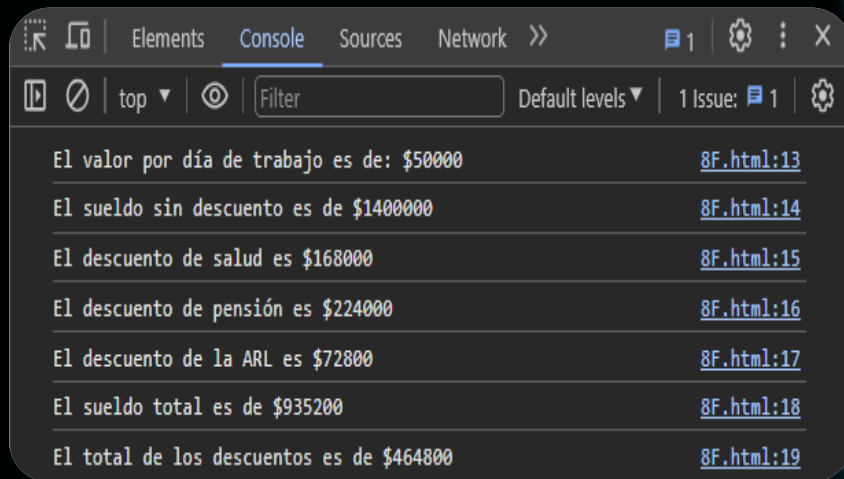
Esta función devuelve la suma de los valores de los subsidios de salud, pensión y arl.

VARIABLE	TIPO
Salud	INT
pension	INT
arl	INT

FUNCION: SUELDO

DESCRIPCION: El código calcula el sueldo total de un trabajador considerando los descuentos por salud, pensión y ARL (Administradora de Riesgos Laborales), basado en un valor diario de trabajo y el número de días trabajados. Además, proporciona el detalle de cada descuento por separado, así como el sueldo total después de aplicar los descuentos.

VARIABLE	TIPO
valDia	INT
diaTra	INT
subsidio	STRING
salario	INT
porcentaje	FLOAT
deducible	INT



FUNCION: OBTENER EL SUELDO GANADO POR UN TRABAJADOR

HTML

```
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="./8F.js"></script>
</head>
<body>
  <script>
    let valDia = 50000;
    let diaTra = 28;

    console.log("El valor por día de trabajo es de $" + valDia);
    console.log("El sueldo sin descuento es de $" + (diaTra * valDia));
    console.log("El descuento de salud es $" + subsidios("salud", diaTra, valDia));
    console.log("El descuento de pensión es $" + subsidios("pension", diaTra, valDia));
    console.log("El descuento de la ARL es $" + subsidios("arl", diaTra, valDia));
    console.log("El sueldo total es de $" + pagoTotal(diaTra, valDia));
    console.log("El total de los descuentos es de $" + des(subsidios("salud", diaTra, valDia),
subsidios("pension", diaTra, valDia), subsidios("arl", diaTra, valDia)));

  </script>
</body>
</html>
```

FUNCION: OBTENER EL SUELDO GANADO POR UN TRABAJADOR

JS

```
function subsidios(psubsidio, pdias, pvdia) {  
  let subsidio = psubsidio;  
  let salario = pdias * pvdia;  
  let porcentaje;  
  
  switch (subsidio) {  
    case "salud":  
      porcentaje = 0.12;  
      break;  
    case "pension":  
      porcentaje = 0.16;  
      break;  
    case "ar1":  
      porcentaje = 0.052;  
      break;  
    default:  
      return "No hay más subsidios";  
  }  
  
  return salario * porcentaje;  
}  
  
function pagoTotal(pdias, pvdia) {  
  let salario = pdias * pvdia;  
  let deducible = subsidios("salud", pdias, pvdia) + subsidios("pension", pdias, pvdia) + subsidios("ar1", pdias, pvdia);  
  return salario - deducible;  
}  
  
function des(salud, pension, ar1) {  
  return salud + pension + ar1;  
}
```

FUNCION: SI ES MAYOR DE EDAD

Funcion: mayEdad

Esta función toma un parámetro pedad que representa la edad de una persona. Luego, verifica si la edad es mayor que 17. Si es así, devuelve un mensaje indicando que la persona es mayor de edad. Si no, devuelve un mensaje indicando que la persona es menor de edad.

VARIABLE	TIPO
edad	INT

Funcion: mayEdadExp

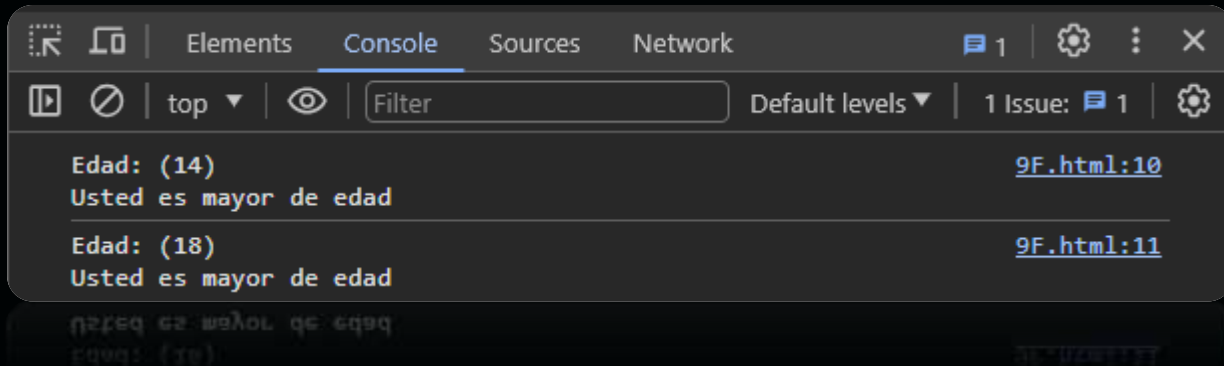
Al igual que la función mayEdad, esta función toma un parámetro pedad que representa la edad de una persona. Luego, verifica si la edad es mayor que 17 y devuelve un mensaje similar dependiendo del resultado.

VARIABLE	TIPO
edad	INT

FUNCION: SI ES MAYOR DE EDAD

DESCRIPCION: El código imprime en la consola del navegador si una persona es mayor de edad o no, según la edad proporcionada.

VARIABLE	TIPO
edad	INT



FUNCION: SI ES MAYOR DE EDAD

HTML

```
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="./9F.js"></script>
</head>
<body>
  <script>
    console.log(mayEdad(14));
    console.log(mayEdadExp(18));
  </script>
</body>
</html>
```

JS

```
function mayEdad(pedad){
  let edad=pedad;
  if(edad>17){
    return "Edad: "+"("+edad+") "+"\nUsted
es mayor de edad";
  }else{
    return "Edad: "+"("+edad+") "+"\nUsted
es mayor de edad";
  }
}

function mayEdadExp(pedad){
  let edad=pedad;
  if(edad>17){
    return "Edad: "+"("+edad+") "+"\nUsted
es mayor de edad";
  }else{
    return "Edad: "+"("+edad+") "+"\nUsted
es mayor de edad";
  }
}
```

FUNCION: EDAD CON EL AÑO DE NACIMIENTO

Funcion: edadF

Esta función toma un parámetro pNacimiento que representa el año de nacimiento de una persona. Luego, calcula la edad de la persona restando el año de nacimiento del año actual. Finalmente, devuelve un mensaje indicando si la persona es mayor o menor de edad, según la edad calculada.

VARIABLE	TIPO
nac	INT
añoact	INT
edad	INT

Funcion: edadFExp

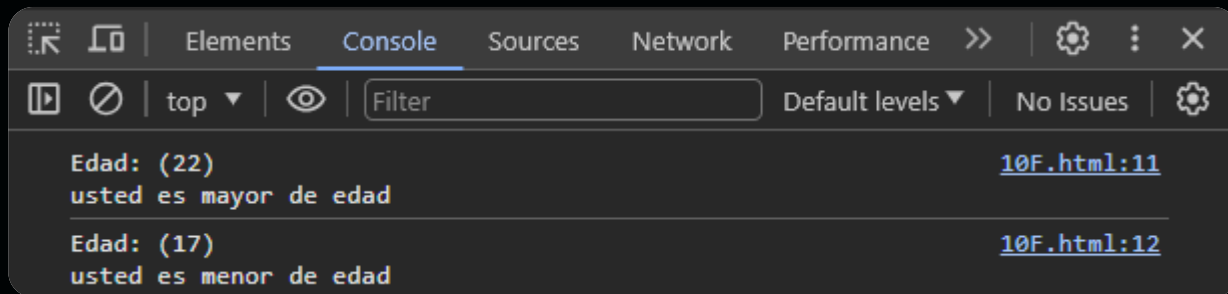
Al igual que la función edadF, esta función toma un parámetro pNacimiento que representa el año de nacimiento de una persona. Luego, calcula la edad de la persona restando el año de nacimiento del año actual. Finalmente, devuelve un mensaje similar dependiendo de si la persona es mayor o menor de edad.

VARIABLE	TIPO
nac	INT
añoact	INT
edad	INT

FUNCION: EDAD CON EL AÑO DE NACIMIENTO

DESCRIPCION: El código imprime en la consola del navegador si una persona es mayor o menor de edad, basándose en su año de nacimiento proporcionado.

VARIABLE	TIPO
nac	INT
añoact	INT
edad	INT



FUNCION: EDAD CON EL AÑO DE NACIMIENTO

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="./10F.js"></script>
</head>
<body>
  <script>
    console.log(edadF(2002));
    console.log(edadFExp(2007));
  </script>
</body>
</html>
```

JS

```
function edadF(pNacimiento){
  let nac=pNacimiento;
  let añoact=2024;
  let edad;
  edad=añoact-nac;
  if(edad>17){
    return "Edad: "+"(+edad+) "+"\nusted es mayor de edad";
  }else{
    return "Edad: "+"(+edad+) "+"\nusted es menor de edad";
  }
}

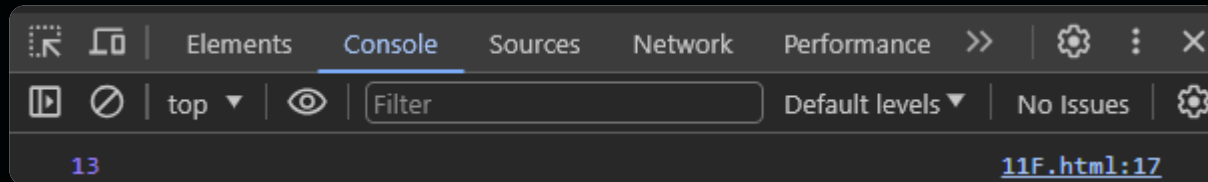
const edadFExp=function(pNacimiento){
  let nac=pNacimiento;
  let añoact=2024;
  let edad;
  edad =añoact-nac;
  if(edad>17){
    return "Edad: "+"(+edad+) "+"\nusted es mayor de edad";
  }else{
    return "Edad: "+"(+edad+) "+"\nusted es menor de edad";
  }
}
```

FUNCION: IMPRIMIR QUE NÚMERO ES EL MAYOR DE 3 NÚMEROS

Funcion: may

Esta función toma tres números como parámetros (pnum1, pnum2 y pnum3). Luego, compara estos números para determinar cuál es el mayor de los tres. Si los tres números son iguales, devuelve un mensaje indicando que los tres números son iguales. De lo contrario, compara los números entre sí para encontrar el mayor y lo devuelve.

VARIABLE	TIPO
num1	INT
num2	INT
num3	INT



FUNCION: EDAD CON EL AÑO DE NACIMIENTO

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
  <script src="./11F.js"></script>
</head>
<body>
  <script>
    let num1 = 9
    let num2 = 10
    let num3 = 13

    alert("A continuacion le aparecerá el
    número mayor de los tres números digitados o si
    estos son iguales");

    console.log(may(num1, num2, num3));
  </script>
</body>
</html>
```

JS

```
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;
      }
    }
  }
}
```

FUNCION: CALCULAR EL ÁREA DE 3 CUADRADOS E IMPRIMIR CUÁL TIENE EL MAYOR ÁREA.

Funcion: areacuad

Esta función toma un número plado como parámetro, que representa la longitud del lado de un cuadrado. Luego, calcula el área del cuadrado multiplicando la longitud del lado por sí mismo y devuelve el resultado.

VARIABLE	TIPO
lado	INT
cuadrado	INT

Funcion:may

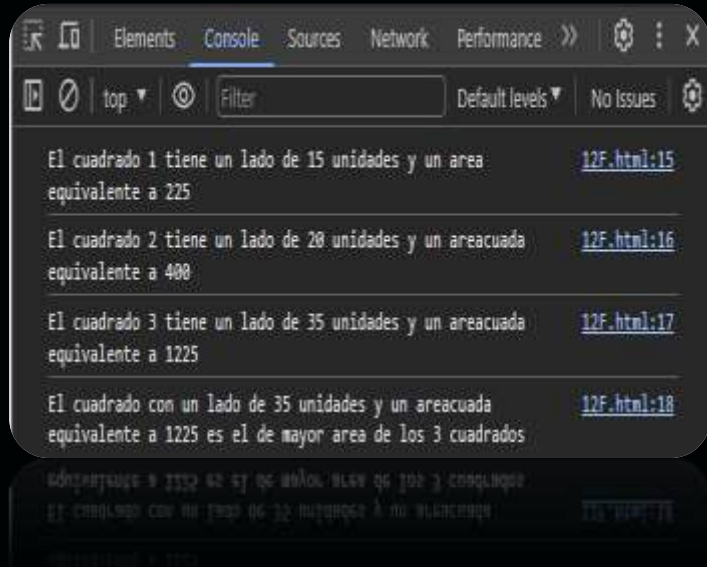
Esta función toma tres números como parámetros (pcu1, pcu2 y pcu3). Luego, compara estos números para determinar cuál es el mayor de los tres. Si los tres números son iguales, devuelve un mensaje de error. De lo contrario, compara los números entre sí para encontrar el mayor y lo devuelve.

VARIABLE	TIPO
cu1	INT
cu2	INT
cu3	INT

FUNCION: CALCULAR EL ÁREA DE 3 CUADRADOS E IMPRIMIR CUÁL TIENE EL MAYOR ÁREA.

DESCRIPCION: El código calcula el área de tres cuadrados dados sus lados y determina cuál de ellos tiene el área más grande.

VARIABLE	TIPO
lado1	INT
lado2	INT
lado3	INT
cu1	INT
cu2	INT
cu3	INT
lado	INT
cuadrado	INT



FUNCION: CALCULAR EL ÁREA DE 3 CUADRADOS E IMPRIMIR CUÁL TIENE EL MAYOR ÁREA

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
  <script src="./12F.js"></script>
</head>
<body>
  <script>
    let lado1 = 15
    let lado2 = 20
    let lado3 = 35

    console.log("El cuadrado 1 tiene un lado de "+lado1+"
unidades y un area equivalente a "+areacuad(lado1));
    console.log("El cuadrado 2 tiene un lado de "+lado2+"
unidades y un areacuada equivalente a "+areacuad(lado2));
    console.log("El cuadrado 3 tiene un lado de "+lado3+"
unidades y un areacuada equivalente a "+areacuad(lado3));
    console.log("El cuadrado con un lado de
"+may(lado1,lado2,lado3)+" unidades y un areacuada
equivalente a
"+may(areacuad(lado1),areacuad(lado2),areacuad(lado3))+" es
el de mayor area de los 3 cuadrados");
  </script>
</body>
</html>
```

JS

```
function areacuad(plado){
  let lado=plado;
  let cuadrado=lado*lado;
  return cuadrado;
}

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;
      }
    }
  }
}
```

FUNCION: CALCULAR 3 EDADES, SU PROMEDIO Y SI ES MAYOR DE EDAD O NO

Funcion: eda

Propósito: Esta función toma un parámetro pfecNa que representa el año de nacimiento de una persona. Luego, calcula la edad de la persona restando el año de nacimiento del año actual.

Finalmente, devuelve la edad calcula

VARIABLE	TIPO
fecNa	INT
fecAc	INT
edad	INT

Funcion: pro

:Esta función toma tres parámetros peda1, peda2 y peda3, que representan las edades de tres personas. Luego, calcula el promedio de estas edades y devuelve el resultado.

VARIABLE	TIPO
eda1	INT
eda2	INT
eda3	INT
prom	FLOAT

Funcion: may

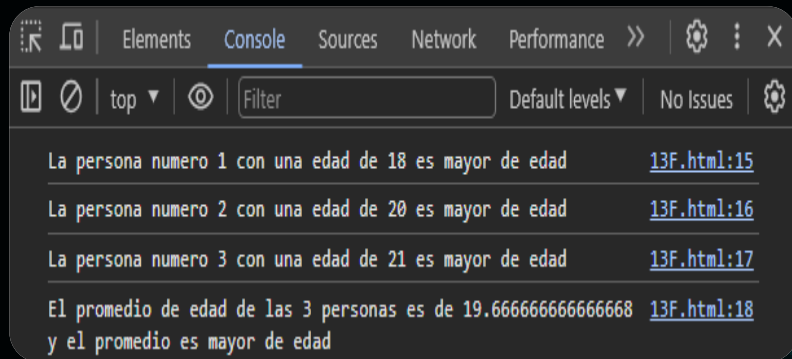
Esta función toma un parámetro ped que representa una edad. Luego, verifica si la edad es mayor que 17 y devuelve un mensaje indicando si la persona es mayor o menor de edad.

VARIABLE	TIPO
ed	INT

FUNCION: CALCULAR 3 EDADES, SU PROMEDIO Y SI ES MAYOR DE EDAD O NO

DESCRIPCION: Este código calcula la edad de tres personas a partir de sus fechas de nacimiento, determina si cada una es mayor o menor de edad, y calcula el promedio de sus edades.

VARIABLE	TIPO
fecNa1	INT
fecNa2	INT
fecNa3	INT
fecNa	INT
fecAc	INT
edad	INT
eda1	INT
eda2	INT
eda3	INT
ed	INT
prom	FLOAT



FUNCION: CALCULAR EL ÁREA DE 3 CUADRADOS E IMPRIMIR CUÁL TIENE EL MAYOR ÁREA

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
  <script src="./13F.js"></script>
</head>
<body>
  <script>
    let fecNa1 = 2006
    let fecNa2 = 2004
    let fecNa3 = 2003

    console.log("La persona numero 1 con una edad de
"+eda(fecNa1)+" "+may(eda(fecNa1)));
    console.log("La persona numero 2 con una edad de
"+eda(fecNa2)+" "+may(eda(fecNa2)));
    console.log("La persona numero 3 con una edad de
"+eda(fecNa3)+" "+may(eda(fecNa3)));
    console.log("El promedio de edad de las 3 personas
es de "+pro(eda(fecNa1),eda(fecNa2),eda(fecNa3))+" y el
promedio "+may(pro(eda(fecNa1),eda(fecNa2),eda(fecNa3))));
  </script>
</body>
</html>
```

JS

```
function eda(pfecNa){
  let fecNa = pfecNa;
  let fecAc = 2024;
  let edad;
  edad = fecAc - fecNa;
  return edad;
}

function pro(peda1, peda2, peda3){
  let eda1 = peda1;
  let eda2 = peda2;
  let eda3 = peda3;
  let prom;
  prom = (eda1+eda2+eda3)/3;
  return prom;
}

function may(ped){
  let ed = ped;
  if (parseInt(ed)>17) {
    return "es mayor de edad";
  }
  else{
    return "es menor de edad";
  }
}
```

FUNCION: CALCULAR LA NOTA DE UNA PERSONA

Funcion: pornot

Esta función toma dos parámetros pnot y pporc, que representan una nota y un porcentaje respectivamente. Luego, calcula la nota ponderada multiplicando la nota por el porcentaje y devuelve el resultado.

VARIABLE	TIPO
not	INT
porce	INT
rnot	INT

Funcion: Suma

Esta función toma tres parámetros pnumUno, pnumDos y pnumTre, que representan tres números. Luego, suma estos números y devuelve el resultado.

VARIABLE	TIPO
numUno	INT
numDos	INT
numTre	INT
rnot	INT

Funcion: may

Esta función toma un parámetro psuma, que representa una suma de notas. Luego, verifica el valor de la suma y devuelve un mensaje indicando si la nota final es superior, buena, media o mala.

VARIABLE	TIPO
suma	INT

FUNCION: CALCULAR LA NOTA DE UNA PERSONA

calcular el porcentaje de tres notas y determinar la calidad total de esas notas en función de su suma.

VARIABLE	TIPO
not1	NUMBER
porcet1	NUMBER
not2	NUMBER
porcet2	NUMBER
not3	NUMBER
porcet3	NUMBER
p1	NUMBER
p2	NUMBER
p3	NUMBER
not	NUMBER
porce	NUMBER
not	NUMBER
numUno	NUMBER
numDos	NUMBER
numTre	NUMBER
suma	NUMBER



FUNCION: CALCULAR LA NOTA DE UNA PERSONA

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="./15F.js"></script>
</head>
<body>
  <script>
    let not1 = 5
    let pornot1 = 15
    let not2 = 7
    let pornot2 = 15
    let not3 = 9
    let pornot3 = 70

    let p1 = pornot1/100;
    let p2 = pornot2/100;
    let p3 = pornot3/100;

    console.log("La nota 1 es "+not1+" la cual tiene un porcentaje de "+pornot1+"%");
    console.log("La nota 2 es "+not2+" la cual tiene un porcentaje de "+pornot2+"%");
    console.log("La nota 3 es "+not3+" la cual tiene un porcentaje de "+pornot3+"%");
    console.log("Su nota final es de "+suma(pornot(not1,p1),pornot(not2,p2),pornot(not3,p3)));
    console.log(may(suma(pornot(not1,p1),pornot(not2,p2),pornot(not3,p3))));
  </script>
</body>
```

FUNCION: CALCULAR LA NOTA DE UNA PERSONA

JS

```
function pnot(pnot, pporc) {
  let not = pnot;
  let porce = pporc;
  let rnot;
  rnot = not * porce;
  return rnot;
}

function suma(pnumUno, pnumDos, pnumTre) {
  let numUno = pnumUno;
  let numDos = pnumDos;
  let numTre = pnumTre;
  let sumar;
  sumar = numUno + numDos + numTre;
  return sumar;
}

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";
      }
    }
  }
}
```


FUNCION: CONTAR-WHILE

Funcion: con

Esta función toma un parámetro pinicio y simplemente lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
inicio	INT

Funcion: conExp

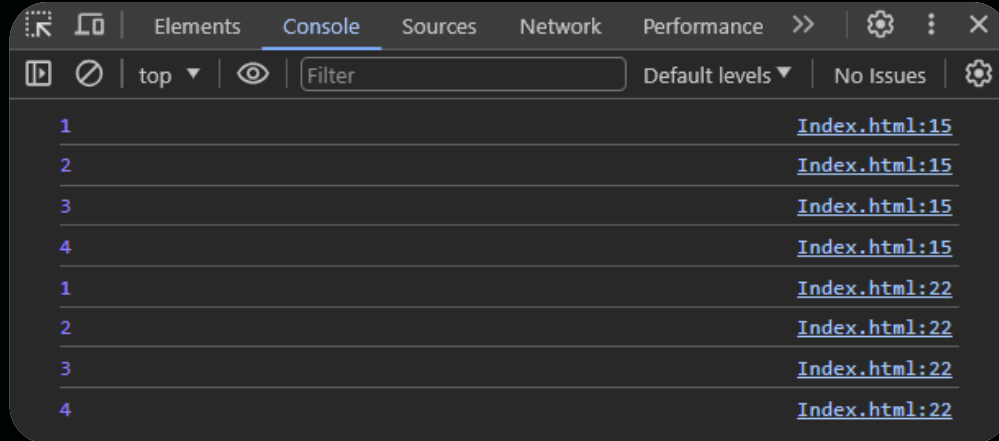
Al igual que la función con, esta función toma un parámetro pinicio y simplemente lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
inicio	INT

FUNCION: CONTAR-WHILE

DESCRIPCION: El código cuenta desde cero hasta el número tres utilizando un bucle while.

VARIABLE	TIPO
inicio	INT
fin	INT



FUNCION: CONTAR-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Contar while</title>
  <script src="./16F.js"></script>
</head>
<body>
  <script>
    let inicio=0;
    let fin=4;
    while(inicio<fin){
      inicio=inicio+1;
      console.log(con(inicio));
    }

    inicio=0;
    fin=4;
    while(inicio<fin){
      inicio=inicio+1;
      console.log(conExp(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function con(pinicio){
  let inicio=pinicio;
  return inicio;
}
```

```
const conExp=function(pinicio){
  let inicio=pinicio;
  return inicio;
}
```

FUNCION: CONTAR-FOR

Funcion: con

Esta función toma un parámetro pinicio y simplemente lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
inicio	INT

Funcion: conExp

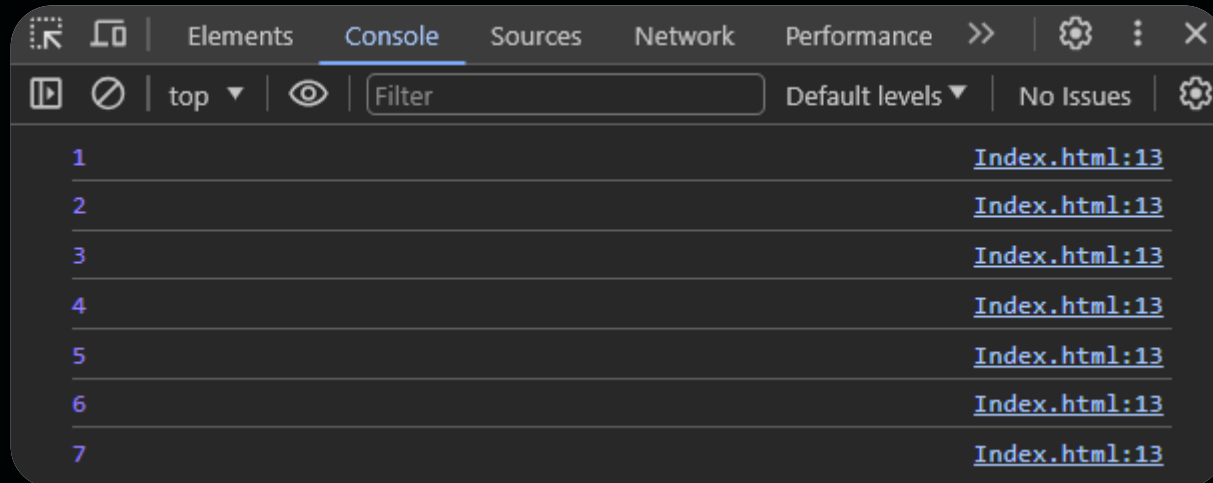
Al igual que la función con, esta función toma un parámetro pinicio y simplemente lo devuelve sin realizar ninguna operación adicional.

VARIABLE	TIPO
inicio	INT

FUNCION: CONTAR-FOR

DESCRIPCION: El código cuenta desde uno hasta el número siete utilizando un bucle for.

VARIABLE	TIPO
inicio	INT
fin	INT



FUNCION: CONTAR-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Contar hasta el 5 (for)</title>
  <script src="./17F.js"></script>
</head>
<body>
  <script>
    let fin=7;
    for(inicio=1;inicio<=fin;inicio++){
      console.log(con(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function con(pinicio){
  let inicio=pinicio;
  return inicio;
}

const conExp=function(pinicio){
  let inicio=pinicio;
  return inicio;
}
```

FUNCION: FACTORIAL-WHILE

Funcion: fac

Esta función toma un número pnun como parámetro y calcula su factorial. Utiliza un bucle while para multiplicar todos los números desde 1 hasta pnun.

VARIABLE	TIPO
multi	INT
cont	INT
fac	INT

Funcion: facExp

Al igual que la función fac, esta función toma un número pnun como parámetro y calcula su factorial utilizando un bucle while.

VARIABLE	TIPO
multi	INT
cont	INT
fac	INT

FUNCION: FACTORIAL-WHILE

DESCRIPCION: El código calcula el factorial de un número dado y luego imprime el resultado en la consola del navegador utilizando while.

VARIABLE	TIPO
num	INT
resultado	INT
multi	INT
cont	INT
fac	INT

```
1 JS17.js:11
2 JS17.js:11
6 JS17.js:11
24 JS17.js:11
120 JS17.js:11
720 JS17.js:11
5040 JS17.js:11
5040 Index17.html:13
```


FUNCION: FACTORIAL-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="JS17.js"></script>
</head>
<body>
  <script>
    let num = 7;
    let resultado = fac(num);
    console.log(resultado);
  </script>
</body>
</html>
```

JS

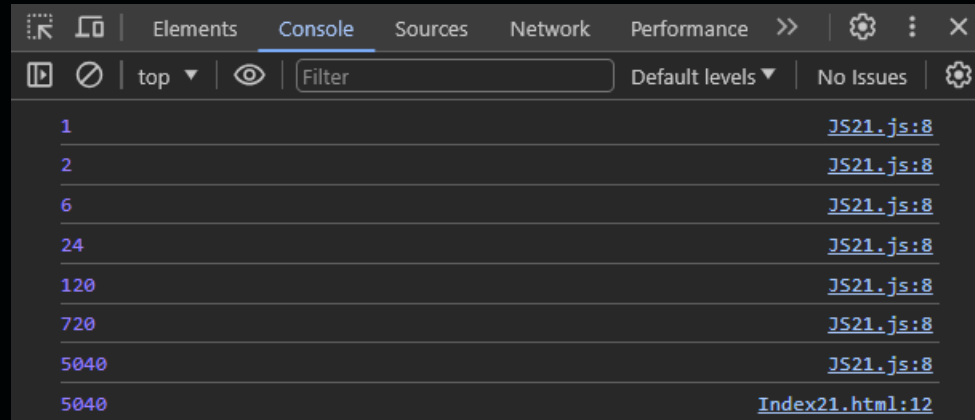
```
function fac(pnum){
  let multi = pnum;
  let cont = 0;
  let fac = 1;
  while(multi > cont){
    cont++;
    fac *= cont;
    console.log(fac);
  }
  return fac;
}

// Como Exp
const facExp = function(pnum){
  let multi = pnum;
  let cont = 0;
  let fac = 1;
  while(multi > cont){
    cont++;
    fac *= cont;
    console.log(fac);
  }
  return fac;
}
```

FUNCION: FACTORIAL-FOR

DESCRIPCION: El código calcula el factorial de un número dado y luego imprime el resultado en la consola del navegador utilizando for:

VARIABLE	TIPO
num	INT
con	INT
fac	INT



```
1 JS21.js:8
2 JS21.js:8
6 JS21.js:8
24 JS21.js:8
120 JS21.js:8
720 JS21.js:8
5040 JS21.js:8
5040 Index21.html:12
```

FUNCION: FACTORIAL-FOR

Funcion: fac

Calcula el factorial del número pnun dado utilizando un bucle for. La variable fact se multiplica por cada número desde 1 hasta pnun.

VARIABLE	TIPO
fact	INT
con	INT

Funcion: facExp

Al igual que la función fac, calcula el factorial del número pnun dado utilizando un bucle for.

VARIABLE	TIPO
fact	INT
con	INT

FUNCION: FACTORIAL-FOR

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title><script
src="JS21.js"></script>
</head>
<body>
  <script>
    let num = 7
    let fact = fac(num);
    console.log(fact);
  </script>
</body>
</html>
```

JS

```
function fac(pnum) {
  let fact = 1;
  for (let con = 1; con <= pnum; con++) {
    fact *= con;
    console.log(fact);
  }
  return fact;
}

// Como Exp
const facExp=function(pnum) {
  let fact = 1;
  for (let con = 1; con <= pnum; con++) {
    fact *= con;
    console.log(fact);
  }
  return fact;
}
```

FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR- WHILE

Funcion: con

Verifica si el número pinicio es par o impar. Si el resto de dividir inicio entre 2 es igual a 0, devuelve el mensaje "es par", de lo contrario, devuelve "es impar".

VARIABLE	TIPO
inicio	INT

Funcion: conExp

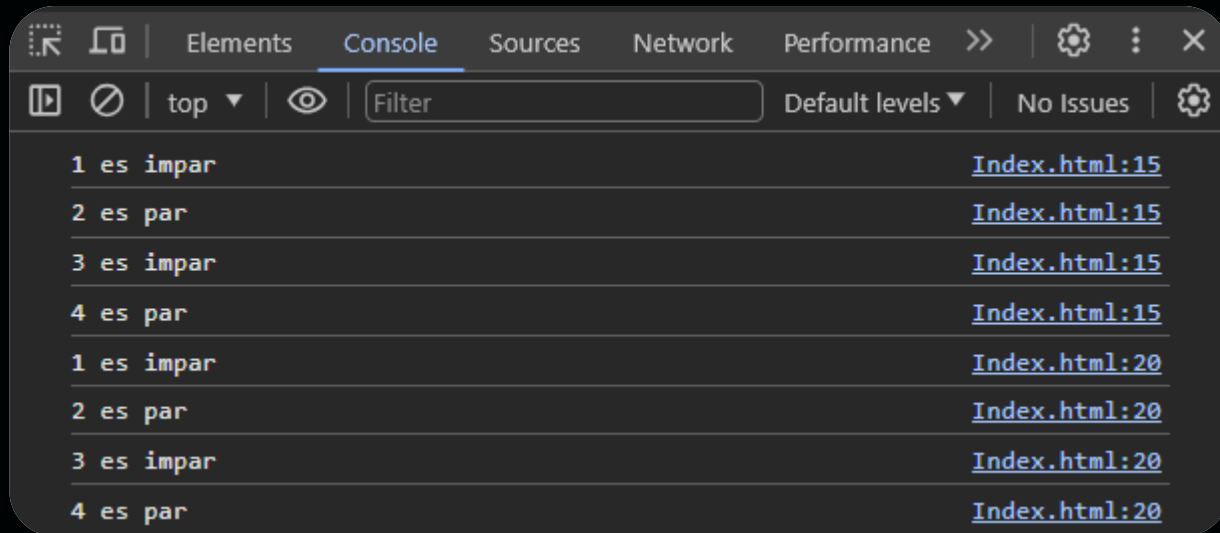
Al igual que la función con, verifica si el número pinicio es par o impar y devuelve un mensaje correspondiente.

VARIABLE	TIPO
inicio	INT

FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR- WHILE

DESCRIPCION: El código cuenta desde uno hasta el número cuatro y dice cuales son pares e impares utilizando un bucle while.

VARIABLE	TIPO
inicio	INT
fin	INT



FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>contar y cual es par e impar</title>
  <script src="./18F.js"></script>
</head>
<body>
  <script>
    let inicio=0;
    let fin=4;
    while(inicio<fin){
      inicio=inicio+1
      console.log(con(inicio));
    }
    inicio=0;
    while(inicio<fin){
      inicio=inicio+1;
      console.log(conExp(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function con(pinicio){
  let inicio=pinicio;
  if(inicio%2==0){
    return inicio+" es par";
  }else{
    return inicio+" es impar";
  }
}

const conExp=function(pinicio){
  let inicio=pinicio;
  if(inicio%2==0){
    return inicio+" es par";
  }else{
    return inicio+" es impar"
  }
}
```

FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR-FOR

Funcion: con

Verifica si el número pinicio es par o impar. Si el resto de dividir inicio entre 2 es igual a 0, devuelve el mensaje "es par", de lo contrario, devuelve "es impar".

VARIABLE	TIPO
inicio	INT

Funcion: conExp

Al igual que la función con, verifica si el número pinicio es par o impar y devuelve un mensaje correspondiente.

VARIABLE	TIPO
inicio	INT

FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR-FOR

DESCRIPCION: El código cuenta desde uno hasta el número cuatro y dice cuales son pares e impares utilizando un bucle for.

VARIABLE	TIPO
inicio	INT
fin	INT

```
1 impar Index.html:14
2 par Index.html:14
3 impar Index.html:14
4 par Index.html:14
1 impar Index.html:18
2 par Index.html:18
3 impar Index.html:18
4 par Index.html:18
```

FUNCION: CONTAR Y DECIR SI ES PAR O IMPAR-FOR

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>contar hasta el 5 (for)</title>
  <script src="/19F.js"></script>
</head>
<body>
  <script>
    let fin=4;

    for(inicio=1;inicio<=fin;inicio++){
      console.log(con(inicio));
    }

    for(inicio=1;inicio<=fin;inicio++){
      console.log(conExp(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function con(pinicio){
  let inicio=pinicio;
  if(inicio%2==0){
    return inicio+" es par";
  }else{
    return inicio+" es impar";
  }
}

const conExp=function(pinicio){
  let inicio=pinicio;
  if(inicio%2==0){
    return inicio+" es par";
  }else{
    return inicio+" es impar"
  }
}
```

FUNCION: TABLA DEL 7-WHILE

Funcion: tabla7

Multiplica el número pinicio por 7 y devuelve el resultado.

VARIABLE	TIPO
inicio	INT

Funcion: tabla7Exp

Al igual que la función tabla7, multiplica el número pinicio por 7 y devuelve el resultado.

VARIABLE	TIPO
inicio	INT

FUNCION: TABLA DEL 7-WHILE

DESCRIPCION: Este código genera las tablas de multiplicar del 7 usando dos funciones diferentes y las imprime en la consola también usando el while.

VARIABLE	TIPO
inicio	NUMBER
fin	NUMBER

```
7x1=7 Index.html:16
7x2=14 Index.html:16
7x3=21 Index.html:16
7x4=28 Index.html:16
7x5=35 Index.html:16
7x1=7 Index.html:22
7x2=14 Index.html:22
7x3=21 Index.html:22
7x4=28 Index.html:22
7x5=35 Index.html:22
```

FUNCION: TABLA DEL 7-WHILE

HTML

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>tabla del 5</title>
  <script src="./js/tabla.js"></script>
</head>
<body>
  <script>
    let inicio=0;
    let fin=5;
    while(inicio<fin){
      inicio=inicio+1;
      console.log("7x"+inicio+"="+tabla7(inicio));
    }

    inicio=0;
    while(inicio<fin){
      inicio=inicio+1;
      console.log("7x"+inicio+"="+tabla7Exp(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function tabla7(pinicio){
  let inicio=pinicio;
  inicio=inicio*7;
  return inicio;
}

const tabla7Exp=function(pinicio){
  let inicio=pinicio;
  inicio=inicio*7;
  return inicio;
}
```

FUNCION: TABLA DEL 7-FOR

Funcion: tabla7

Multiplica el número pinicio por 7 y devuelve el resultado.

VARIABLE	TIPO
inicio	INT

Funcion: tabla7Exp

Al igual que la función tabla7, multiplica el número pinicio por 7 y devuelve el resultado.

VARIABLE	TIPO
inicio	INT

FUNCION: TABLA DEL 7-FOR

DESCRIPCION: Este código genera las tablas de multiplicar del 7 usando dos funciones diferentes y las imprime en la consola también usando el for.

VARIABLE	TIPO
inicio	INT
fin	INT

```
7x1=7 Index.html:13
7x2=14 Index.html:13
7x3=21 Index.html:13
7x4=28 Index.html:13
7x5=35 Index.html:13
7x1=7 Index.html:17
7x2=14 Index.html:17
7x3=21 Index.html:17
7x4=28 Index.html:17
7x5=35 Index.html:17
```

FUNCION: TABLA DEL 7-FOR

HTML

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>tabla del 5</title>
  <script src="./js/tabla.js"></script>
</head>
<body>
  <script>
    let fin=5;
    for(inicio=1;inicio<=fin;inicio++){
      console.log("7x"+inicio+"="+tabla7(inicio));
    }

    for(inicio=1;inicio<=fin;inicio++){
      console.log("7x"+inicio+"="+tabla7Exp(inicio)
);
    }
  </script>
</body>
</html>
```

JS

```
function tabla7(pinicio){
  let inicio=pinicio;
  inicio=inicio*7;
  return inicio;
}

const tabla7Exp=function(pinicio){
  let inicio=pinicio;
  inicio=inicio*7;
  return inicio;
}
```


FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-WHILE

Funcion: tabla11

Multiplica el número pinicio por 11 y almacena el resultado en inicio. Luego verifica si inicio es par o impar y devuelve un mensaje indicando su paridad.

VARIABLE	TIPO
inicio	INT

Funcion: tabla11Exp

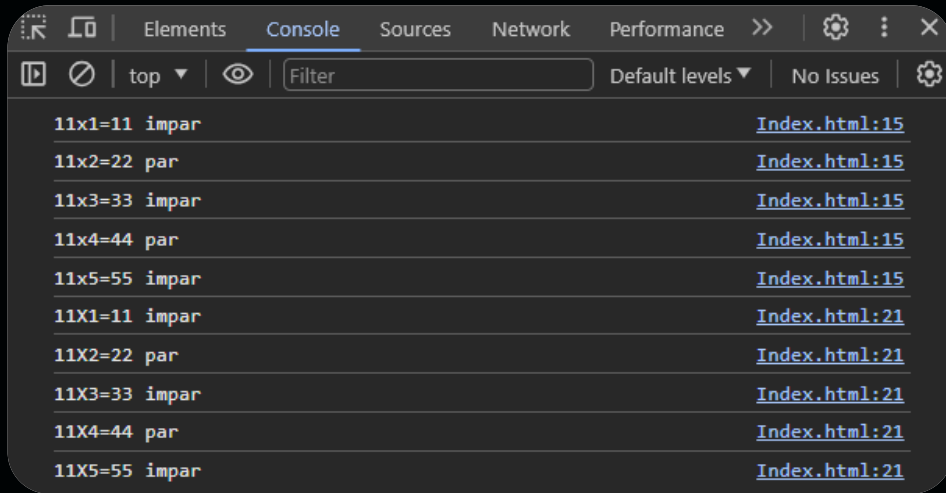
Al igual que la función tabla11, multiplica el número pinicio por 11 y almacena el resultado en inicio. Luego verifica si inicio es par o impar y devuelve un mensaje indicando su paridad.

VARIABLE	TIPO
inicio	INT

FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-WHILE

DESCRIPCION: Este código genera las tablas de multiplicar del 11 usando el WHILE y dice cual es par e impar.

VARIABLE	TIPO
inicio	INT
fin	INT



```
11x1=11 impar Index.html:15
11x2=22 par Index.html:15
11x3=33 impar Index.html:15
11x4=44 par Index.html:15
11x5=55 impar Index.html:15
11X1=11 impar Index.html:21
11X2=22 par Index.html:21
11X3=33 impar Index.html:21
11X4=44 par Index.html:21
11X5=55 impar Index.html:21
```

FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>tabla del 9 y cual es par o impar</title>
  <script src="./js/tabla11.js"></script>
</head>
<body>
  <script>
    let inicio=0;
    let fin=5;
    while(inicio<fin){
      inicio=inicio+1;
      console.log("11x"+inicio+"="+tabla11(inicio));
    }

    inicio=0;
    while(inicio<fin){
      inicio=inicio+1;
      console.log("11X"+inicio+"="+tabla11Exp(inicio));
    }
  </script>
</body>
</html>
```

JS

```
function tabla11(pinicio){
  let inicio=pinicio;
  inicio=inicio*11;
  if(inicio%2==0){
    return inicio+" par"
  }else{
    return inicio+" impar"
  }
}

const tabla11Exp=function(pinicio){
  let inicio=pinicio;
  inicio=inicio*11;
  if(inicio%2==0){
    return inicio+" par";
  }else{
    return inicio+" impar";
  }
}
```

FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-FOR

Funcion: tabla11

Multiplica el número pinicio por 11 y almacena el resultado en inicio. Luego verifica si inicio es par o impar y devuelve un mensaje indicando su paridad.

VARIABLE	TIPO
inicio	INT

Funcion: tabla11Exp

Al igual que la función tabla11, multiplica el número pinicio por 11 y almacena el resultado en inicio. Luego verifica si inicio es par o impar y devuelve un mensaje indicando su paridad.

VARIABLE	TIPO
inicio	INT

FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-FOR

DESCRIPCION: Este código genera las tablas de multiplicar del 11 usando el FOR y dice cual es par e impar.

VARIABLE	TIPO
inicio	INT
fin	INT

```
11x1=11 impar Index.html:13
11x2=22 par Index.html:13
11x3=33 impar Index.html:13
11x4=44 par Index.html:13
11x5=55 impar Index.html:13
11x1=11 impar Index.html:17
11x2=22 par Index.html:17
11x3=33 impar Index.html:17
11x4=44 par Index.html:17
11x5=55 impar Index.html:17
```

FUNCION: TABLA DEL 11 SI ES PAR O IMPAR-FOR

HTML

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>tabla del 9 y saber cual es par e
impar</title>
  <script src="./js/tabla11.js"></script>
</head>
<body>
  <script>
    let fin=5;
    for(inicio=1;inicio<=fin;inicio++){
      console.log("11x"+inicio+"="+tabla11(inicio))
    }

    for(inicio=1;inicio<=fin;inicio++){
      console.log("11x"+inicio+"="+tabla11Exp(inici
o));
    }
  </script>
</script>
</body>
</html>
```

JS

```
function tabla11(pinicio){
  let inicio=pinicio;
  inicio=inicio*11;
  if(inicio%2==0){
    return inicio+" par"
  }else{
    return inicio+" impar"
  }
}

const tabla11Exp=function(pinicio){
  let inicio=pinicio;
  inicio=inicio*11;
  if(inicio%2==0){
    return inicio+" par";
  }else{
    return inicio+" impar";
  }
}
```

FUNCION: HACER LAS TABLAS DE MULTIPLICAR QUE DESEE EL USUARIO HASTA DONDE EL USUARIO INDIQUE-WHILE

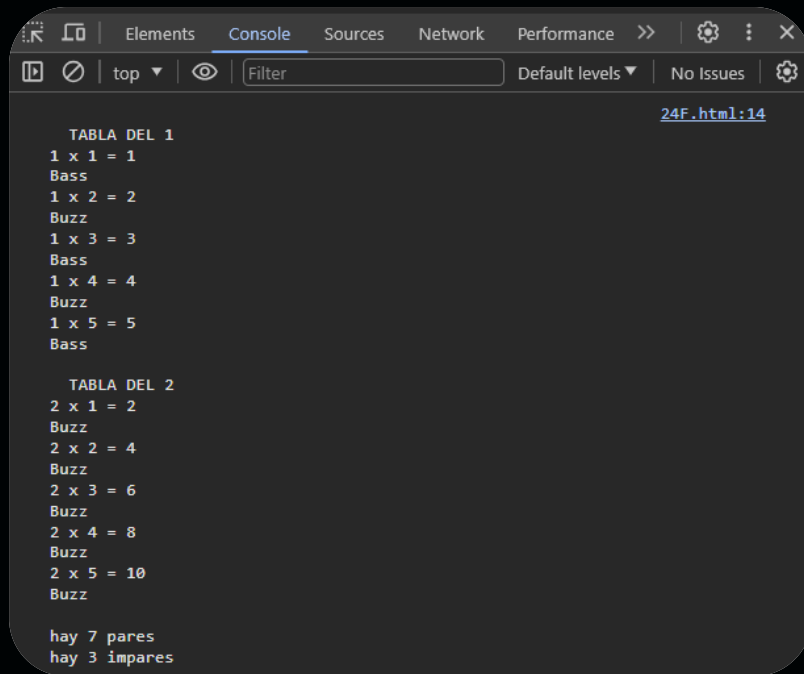
Funcion: mult

La función genera una tabla de multiplicar del 1 al pnum1 multiplicada por los números del 1 al pnum2. Para cada producto, verifica si es par o impar. Luego, cuenta la cantidad de productos pares e impares.

La variable res se actualiza con la tabla de multiplicar y los mensajes de pares e impares.

Finalmente, la función devuelve la tabla de multiplicar junto con la cantidad de productos pares e impares.

VARIABLE	TIPO
num1	INT
num2	INT
con1	INT
con2	INT
multi	INT
par	INT
imp	INT
res	STRING



```
TABLA DEL 1
1 x 1 = 1
Bass
1 x 2 = 2
Buzz
1 x 3 = 3
Bass
1 x 4 = 4
Buzz
1 x 5 = 5
Bass

TABLA DEL 2
2 x 1 = 2
Buzz
2 x 2 = 4
Buzz
2 x 3 = 6
Buzz
2 x 4 = 8
Buzz
2 x 5 = 10
Buzz

hay 7 pares
hay 3 impares

hay 3 pares
hay 3 impares

multi
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
```

FUNCION: HACER LAS TABLAS DE MULTIPLICAR QUE DESEE EL USUARIO HASTA DONDE EL USUARIO INDIQUE-WHILE

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
  <script src="./24F.js"></script>
</head>
<body>
  <script>
    let num1 = 2
    let num2 = 5

    console.log(mult(num1,num2));
  </script>
</body>
</html>
```

JS

```
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";
}
```


FUNCION: HACER LAS TABLAS DE MULTIPLICAR QUE DESEE EL USUARIO HASTA DONDE EL USUARIO INDIQUE-FOR

Funcion: mult

La función genera una tabla de multiplicar del 1 al pnun1 multiplicada por los números del 1 al pnun2. Para cada producto, verifica si es par o impar. Luego, cuenta la cantidad de productos pares e impares.

La variable res se actualiza con la tabla de multiplicar y los mensajes de pares e impares.

Finalmente, la función devuelve la tabla de multiplicar junto con la cantidad de productos pares e impares.

VARIABLE	TIPO
num1	INT
num2	INT
con1	INT
con2	INT
multi	INT
par	INT
imp	INT
res	STRING

[25F.html:14](#)

TABLA DEL 1

1 x 1 = 1

Bass

1 x 2 = 2

Buzz

1 x 3 = 3

Bass

1 x 4 = 4

Buzz

1 x 5 = 5

Bass

TABLA DEL 2

2 x 1 = 2

Buzz

2 x 2 = 4

Buzz

2 x 3 = 6

Buzz

2 x 4 = 8

Buzz

2 x 5 = 10

Buzz

hay 7 pares

hay 3 impares

FUNCION: HACER LAS TABLAS DE MULTIPLICAR QUE DESEE EL USUARIO HASTA DONDE EL USUARIO INDIQUE-FOR

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="./25F.js"></script>
</head>
<body>
  <script>
    let num1 = 2
    let num2 = 5

    console.log(mult(num1,num2));
  </script>
</body>
</html>
```

JS

```
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";
}
```