



Control de Flujo y Funciones en JavaScript

Clase 6

KOICA

IGU HANDONG GLOBAL
UNIVERSITY



UNA

OBJETIVOS DE LA CLASE 6

- Repasar los conceptos de estructuras repetitivas.
- Manejar los conceptos de funciones.
- Implementar los códigos de ejemplos propuestos en clase.



Estructura Repetitiva (For)

- Cualquier problema que requiera una estructura repetitiva se puede resolver empleando la estructura while. Pero hay otra estructura repetitiva cuyo planteo es más sencillo en ciertas situaciones.
- Esta estructura se emplea en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones. Ejemplo: cargar 10 números, ingresar 5 notas de alumnos, etc. Conocemos de antemano la cantidad de veces que queremos que el bloque se repita.

Estructura Repetitiva (For)

- Por último, hay que decir que la ejecución de la sentencia break dentro de cualquier parte del bucle provoca la salida inmediata del mismo.

Sintaxis:

```
for (<Inicialización> ; <Condición> ; <Incremento o Decremento>)
```

```
{
```

```
    <Instrucciones>
```

```
}
```

Estructura Repetitiva (For)

- Esta estructura repetitiva tiene tres argumentos: variable de inicialización, condición y variable de incremento o decremento.
- Este tipo de estructura repetitiva se utiliza generalmente cuando sabemos la cantidad de veces que deseamos que se repita el bloque.

Funcionamiento

- 1 - Primero se ejecuta por única vez el primer argumento <Inicialización>. Por lo general se inicializa una variable.
- 2 - El segundo paso es evaluar la (Condición), en caso de ser verdadera se ejecuta el bloque, en caso contrario continúa el programa.
- 3 - El tercer paso es la ejecución de las instrucciones.
- 4 - El cuarto paso es ejecutar el tercer argumento (Incremento o Decremento).
- 5 - Luego se repiten sucesivamente del Segundo al Cuarto Paso.

Ejemplo

— — —

Mostrar por pantalla los números del 1 al 10.

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>

<body>

  <script>
    for (let f = 1; f <= 10; f++) {
      document.write(f + " ");
    }
  </script>

</body>

</html>
```

Estructura Repetitiva (For)

— — —

- Podemos definir la variable `f` directamente dentro del `for` si no se la requiere a dicha variable después del `for`.
- Como la condición se verifica como verdadera ($1 \leq 10$) se ejecuta el bloque del `for` (en este caso mostramos el contenido de la variable `f` y un espacio en blanco).
- Luego de ejecutar el bloque pasa al tercer argumento del `for` (en este caso con el operador `++` se incrementa en uno el contenido de la variable `f`, existe otro operador `--` que decrementa en uno una variable), hubiera sido lo mismo poner `f=f+1` pero este otro operador matemático nos simplifica las cosas.

Estructura Repetitiva (For)

- Importante: Tener en cuenta que no lleva punto y coma al final de los tres argumentos del for.
- El disponer un punto y coma provoca un error lógico y no sintáctico, por lo que el navegador no avisará.

Acotaciones

Acá ya podemos mostrar una ventaja de utilizar la palabra clave 'let' para definir la variable en lugar de la sintaxis antigua de JavaScript de utilizar la palabra clave 'var'.

Si utilizamos la palabra clave 'var' en principio el resultado es lo mismo:

```
<script>

    for (var f = 1; f <= 10; f++) {

        document.write(f + " ");

    }

</script>
```

Acotaciones

Pero si nos equivocamos y volvemos a mostrar la variable 'f' fuera del for:

```
<script>
  for (var f = 1; f <= 10; f++) {
    document.write(f + " ");
  }
  document.write(f + " ");
</script>
```

Acotaciones

Como podemos comprobar la variable 'f' sigue existiendo y tiene almacenado el valor '10' (gracias a dicho valor sale del for).

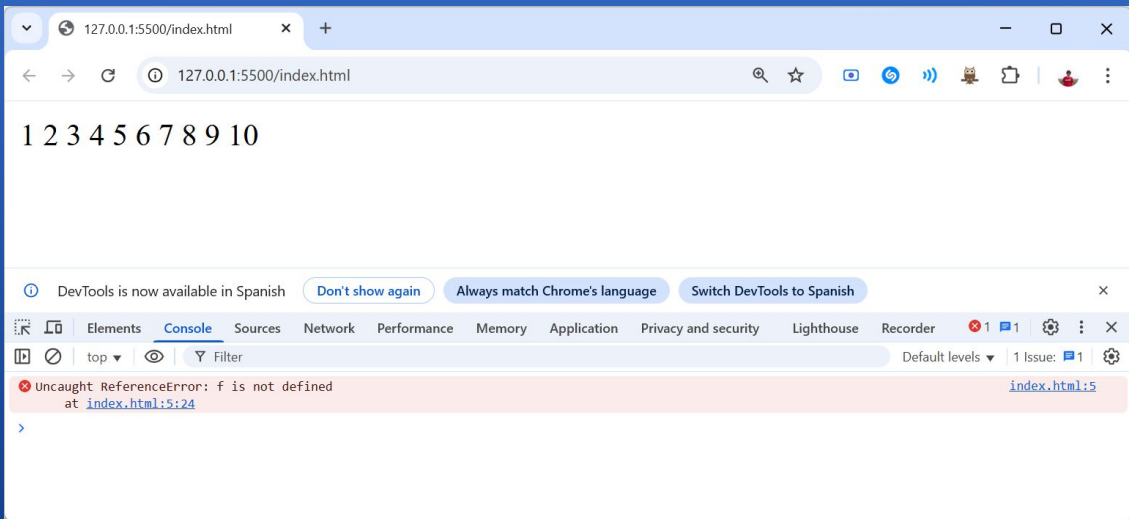
Si utilizamos la palabra clave 'let' la variable 'f' tiene existencia dentro del ámbito donde se la definió, en nuestro caso solo existe dentro del for, luego si codificamos:

```
<script>
  for (let f = 1; f <= 10; f++) {
    document.write(f + " ");
  }
  document.write(f + " ");
</script>
```

Acotaciones

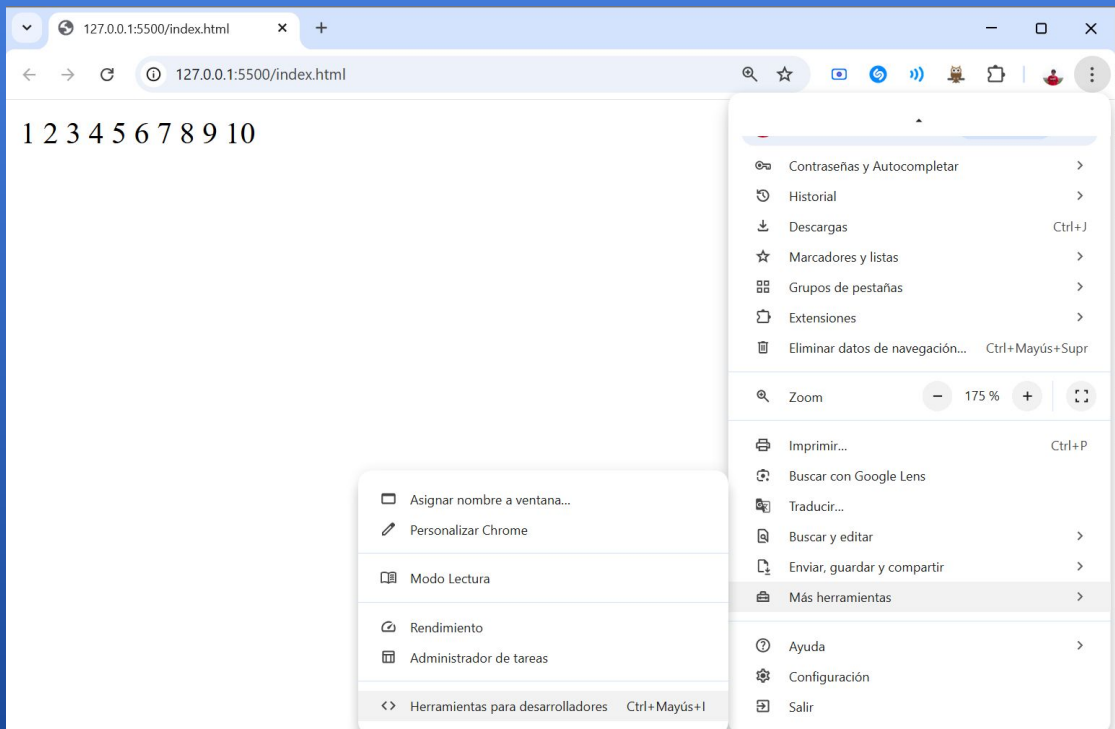
— — —

Podemos ver que el intérprete de JavaScript detecta como un error el acceso a la variable 'f' fuera del for:



Acotaciones

La ventana de "Herramientas para desarrolladores" es de fundamental importancia para detectar los errores sintácticos y lógicos de nuestra aplicación. La podemos abrir presionando la tecla 'F12' o desde el menú de opciones:



Acotar el ámbito para el acceso a una variable es una muy buena práctica que nos va a evitar dolores de cabeza cuando tenemos programas grandes (cientos o miles de líneas)

Problema

Confeccionar un programa que lea 3 pares de datos, cada par de datos corresponde a la medida de la base y la altura de un triángulo. El programa deberá informar:

a) De cada triángulo la medida de su base, su altura y su superficie.

b) La cantidad de triángulos cuya superficie es mayor a 12.

```
<script>

    let contal = 0;
    for (let f = 1; f <= 3; f++) {
        let base = parseInt(prompt( 'Ingrese la
base:'));

        let altura = parseInt(prompt( 'Ingrese
la altura:'));

        let superficie = base * altura / 2;
        if (superficie > 12) {
            contal++;
        }
        document.write( 'Triángulo nro:' + f +
'<br>');

        document.write( 'Base:' + base +
'<br>');

        document.write( 'Altura:' + altura +
'<br>');

        document.write( 'Superficie:' +
superficie + '<br>');
    }
    document.write( 'Cantidad de triángulos con
superficie mayor a 12:' + contal);
</script>
```

Problema

— — —

Desarrollar un programa que solicite la carga de 10 números e imprima la suma de lo últimos 5 valores ingresados.

```
<script>
    /* Desarrollar un programa que solicite
    la carga de 10 números
        e imprima la suma de lo últimos 5
valores ingresados.*/
    let suma = 0;
    for (let f = 1; f <= 10; f++) {
        let valor =
parseInt(prompt('Ingrese un nro:'));
        if (f > 5) {
            suma = suma + valor;
        }
    }
    document.write('La suma de los últimos
cinco valores ingresados es: ' + suma);
</script>
```


Problema

— — —

Desarrollar un programa que muestre la tabla de multiplicar del 5 (del 5 al 50).

```
<script>
    //Desarrollar un programa que muestre
    la tabla de multiplicar del 5 (del 5 al 50)
    let tabla = 5;
    for (let f = 1; f <= 10; f++) {
        document.write('5 x ' + f + ' = ' +
        tabla + '<br>');
        tabla = tabla + 5;
    }
</script>
```

Problema

— — —

Confeccionar un programa que permita ingresar un valor del 1 al 10 y nos muestre la tabla de multiplicar del mismo (los primeros 12 términos)

Ejemplo: Si ingreso 3 deberá aparecer en pantalla los valores 3, 6, 9, hasta el 36.

```
<script>
    /*Confeccionar un programa que permita
    ingresar un valor del 1 al 10 y
        nos muestre la tabla de
multiplicar del mismo (los primeros 12
términos)

        Ejemplo: Si ingreso 3 deberá
aparecer en pantalla los valores 3, 6, 9, hasta
el 36.*/

    let tabla = parseInt(prompt('Ingrese un
valor del 1 al 10:', ''));
    let conta = tabla;
    for (let f = 1; f <= 12; f++) {
        document.write(tabla + 'x ' + f + '
= ' + conta + '<br>');
        conta = conta + tabla;
    }
</script>
```

Problema

— — —

Realizar un programa que lea los lados de 4 triángulos, e informar:

a) De cada uno de ellos, qué tipo de triángulo es: equilátero (tres lados iguales), isósceles (dos lados iguales), o escaleno (ningún lado igual)

b) Cantidad de triángulos de cada tipo.

```
<script>

let cant1 = 0, cant2 = 0, cant3 = 0;
for (let f = 1; f <= 4; f++) {
    let lado1 = parseInt(prompt('Ingrese primer lado:'));
    let lado2 = parseInt(prompt('Ingrese segundo lado:'));
    let lado3 = parseInt(prompt('Ingrese tercer lado:'));
    if (lado1 == lado2 && lado1 == lado3) {
        document.write('Triángulo equilátero.<br>');
        cant1++;
    } else {
        if (lado1 == lado2 || lado1 == lado3 || lado2 == lado3) {
            document.write('Triángulo isósceles.<br>');
            cant2++;
        } else {
            document.write('Triángulo escaleno.<br>');
            cant3++;
        }
    }
}

document.write('<br>');
document.write('Cantidad de triángulos equiláteros:' + cant1 + '<br>');
document.write('Cantidad de triángulos isósceles:' + cant2 + '<br>');
document.write('Cantidad de triángulos escalenos:' + cant3 + '<br>');

</script>
```

Problema

— — —

Escribir un programa que pida ingresar coordenadas (x,y) que representan puntos en el plano. Informar cuántos puntos se han ingresado en el primer, segundo, tercer y cuarto cuadrante. Al comenzar el programa se pide que se ingrese la cantidad de puntos a procesar.

```
<script>

let cuad1 = 0, cuad2 = 0, cuad3 = 0, cuad4 = 0;
let cant = parseInt(prompt('Cuántos puntos procesará:'));
cant = parseInt(cant);
for (let f = 1; f <= cant; f++) {
    let x = parseInt(prompt('Ingrese coordenada x:'));
    let y = parseInt(prompt('Ingrese coordenada y:'));
    if (x > 0 && y > 0) {
        cuad1++;
    } else {
        if (x < 0 && y > 0) {
            cuad2++;
        } else {
            if (x < 0 && y < 0) {
                cuad3++;
            } else {
                if (x > 0 && y < 0) {
                    cuad4++;
                }
            }
        }
    }
}

document.write('Cantidad de puntos ingresados en el primer cuadrante:' + cuad1 + '<br>');
document.write('Cantidad de puntos ingresados en el segundo cuadrante:' + cuad2 +
'<br>');

document.write('Cantidad de puntos ingresados en el tercer cuadrante:' + cuad3 + '<br>');
document.write('Cantidad de puntos ingresados en el cuarto cuadrante:' + cuad4 + '<br>');

</script>
```

Problema

— — —

Se realiza la carga de 10 valores enteros por teclado. Se desea conocer:

- a) La cantidad de valores negativos ingresados.
- b) La cantidad de valores positivos ingresados.
- c) La cantidad de múltiplos de 15.
- d) El valor acumulado de los números ingresados que son pares.

```
<script>

let cantnegativos = 0;
let cantpositivos = 0;
let mult15 = 0;
let sumapares = 0;
for (let f = 1; f <= 10; f++) {
    let valor = parseInt(prompt('Ingrese un valor:'));
    if (valor < 0) {
        cantnegativos++;
    } else {
        if (valor > 0) {
            cantpositivos++;
        }
    }
    if (valor % 15 == 0 && valor != 0) {
        mult15++;
    }
    if (valor % 2 == 0) {
        sumapares = sumapares + valor;
    }
}

document.write('Cantidad de valores negativos:' + cantnegativos + '<br>');
document.write('Cantidad de valores positivos:' + cantpositivos + '<br>');
document.write('Cantidad de múltiplos de 15:' + mult15 + '<br>');
document.write('Suma de los valores pares ingresados:' + sumapares + '<br>');

</script>
```

Problema

Se cuenta con la siguiente información:

Las edades de 5 estudiantes del turno mañana.

Las edades de 6 estudiantes del turno tarde.

Las edades de 11 estudiantes del turno noche.

Las edades de cada estudiante deben ingresarse por teclado.

a) Obtener el promedio de las edades de cada turno (tres promedios).

b) Imprimir dichos promedios (promedio de cada turno).

c) Mostrar por pantalla un mensaje que indique cual de los tres turnos tiene un promedio de edades mayor.

```
<script>

    let suma1 = 0;
    for (let f = 1; f <= 5; f++) {
        let edad = parseInt(prompt( 'Edad de estudiante del turno mañana:' ));
        suma1 = suma1 + edad;
    }
    let suma2 = 0;
    for (let f = 1; f <= 6; f++) {
        let edad = parseInt(prompt( 'Edad de estudiante del turno tarde:' ));
        suma2 = suma2 + edad;
    }
    let suma3 = 0;
    for (let f = 1; f <= 11; f++) {
        let edad = parseInt(prompt( 'Edad de estudiante del turno noche:' ));
        suma3 = suma3 + edad;
    }
    let promedio1 = suma1 / 5;
    let promedio2 = suma2 / 6;
    let promedio3 = suma3 / 11;
    document.write( 'Promedio de edades de alumnos del turno mañana:' + promedio1
+ '<br>' );
    document.write( 'Promedio de edades de alumnos del turno tarde:' + promedio2
+ '<br>' );
    document.write( 'Promedio de edades de alumnos del turno noche:' + promedio3
+ '<br>' );
    if (promedio1 > promedio2 && promedio1 > promedio3) {
        document.write( 'El turno mañana tiene un promedio mayor de edades' );
    } else {
        if (promedio2 > promedio3) {
            document.write( 'El turno tarde tiene un promedio mayor de edades' );
        } else {
            document.write( 'El turno noche tiene un promedio mayor de edades' );
        }
    }
}

</script>
```

Funciones

- En programación es muy frecuente que un determinado procedimiento de cálculo definido por un grupo de sentencias tenga que repetirse varias veces, ya sea en un mismo programa o en otros programas, lo cual implica que se tenga que escribir tantos grupos de aquellas sentencias como veces aparezca dicho proceso.
- La herramienta más potente con que se cuenta para facilitar, reducir y dividir el trabajo en programación, es escribir aquellos grupos de sentencias una sola y única vez bajo la forma de una FUNCIÓN.

Funciones

- Un programa es una cosa compleja de realizar y por lo tanto es importante que esté bien ESTRUCTURADO y también que sea inteligible para las personas. Si un grupo de sentencias realiza una tarea bien definida, entonces puede estar justificado el aislar estas sentencias formando una función, aunque resulte que sólo se le llame o use una vez.
- Hasta ahora hemos visto cómo resolver un problema planteando un único algoritmo.

Funciones

- Con funciones podemos segmentar un programa en varias partes. Frente a un problema, planteamos un algoritmo, éste puede constar de pequeños algoritmos.
- Una función es un conjunto de instrucciones que resuelven una parte del problema y que puede ser utilizado (llamado) desde diferentes partes de un programa.
- Consta de un nombre y parámetros. Con el nombre llamamos a la función, es decir, hacemos referencia a la misma. Los parámetros son valores que se envían y son indispensables para la resolución del mismo. La función realizará alguna operación con los parámetros que le enviamos. Podemos cargar una variable, consultarla, modificarla, imprimirla, etc.

Funciones

- Incluso los programas más sencillos tienen la necesidad de fragmentarse. Las funciones son los únicos tipos de subprogramas que acepta JavaScript. Tienen la siguiente estructura:

```
function <nombre de función>(argumento1, argumento2, ..., argumento n)
{
    <código de la función>
}
```

Funciones

- Debemos buscar un nombre de función que nos indique cuál es su objetivo (Si la función recibe un string y lo centra, tal vez deberíamos llamarla `centrarTitulo`). Veremos que una función puede variar bastante en su estructura, puede tener o no parámetros, retornar un valor, etc.

Funciones

- Ejemplo: Mostrar un mensaje que se repita 3 veces en la página con el siguiente texto:

'Cuidado'

'Ingrese su documento correctamente'

'Cuidado'

'Ingrese su documento correctamente'

'Cuidado'

'Ingrese su documento correctamente'

La solución sin emplear funciones es:

— — —

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de JavaScript </title>
  <meta charset="UTF-8">
</head>

<body>

  <script>
    document.write( "Cuidado<br>" );
    document.write( "Ingrese su documento
correctamente<br>" );
    document.write( "Cuidado<br>" );
    document.write( "Ingrese su documento
correctamente<br>" );
    document.write( "Cuidado<br>" );
    document.write( "Ingrese su documento
correctamente<br>" );
  </script>

</body>
</html>
```

Empleando una función:

— — —

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de JavaScript </title>
  <meta charset="UTF-8">
</head>

<body>

  <script>
    function mostrarMensaje() {
      document.write( "Cuidado<br>" );
      document.write( "Ingrese su documento
correctamente<br>" );
    }

    mostrarMensaje();
    mostrarMensaje();
    mostrarMensaje();
  </script>

</body>
</html>
```

Recordemos que

- JavaScript es sensible a mayúsculas y minúsculas. Si fijamos como nombre a la función `mostrarMensaje` (es decir la segunda palabra con la primer letra en mayúscula) debemos respetar este nombre cuando la llamemos a dicha función.
- Es importante notar que para que una función se ejecute debemos llamarla desde fuera por su nombre (en este ejemplo: `mostrarMensaje()`).
- Cada vez que se llama una función se ejecutan todas las líneas contenidas en la misma. Si no se llama a la función, las instrucciones de la misma nunca se ejecutarán.
- A una función la podemos llamar tantas veces como necesitemos. Las funciones nos ahorran escribir código que se repite con frecuencia y permite que nuestro programa sea más entendible.

Acotaciones

- Hemos estado haciendo uso de funciones existentes por defecto en JavaScript como son:

`parseInt`

`parseFloat`

Problema

Confeccionar una función que solicite el ingreso de un entero y nos muestre la tabla de multiplicar de dicho valor. Llamar luego una vez a la función.

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>

<body>

  <script>
    function calcularTabla() {
      let valor = parseInt(prompt("Ingrese qué tabla de
multiplicar quiere obtener"))
      for (let f = 1; f <= 10; f++) {
        let tabla = f * valor;
        document.write(f + " * " + valor + " = " + tabla +
'<br>')

      }
    }

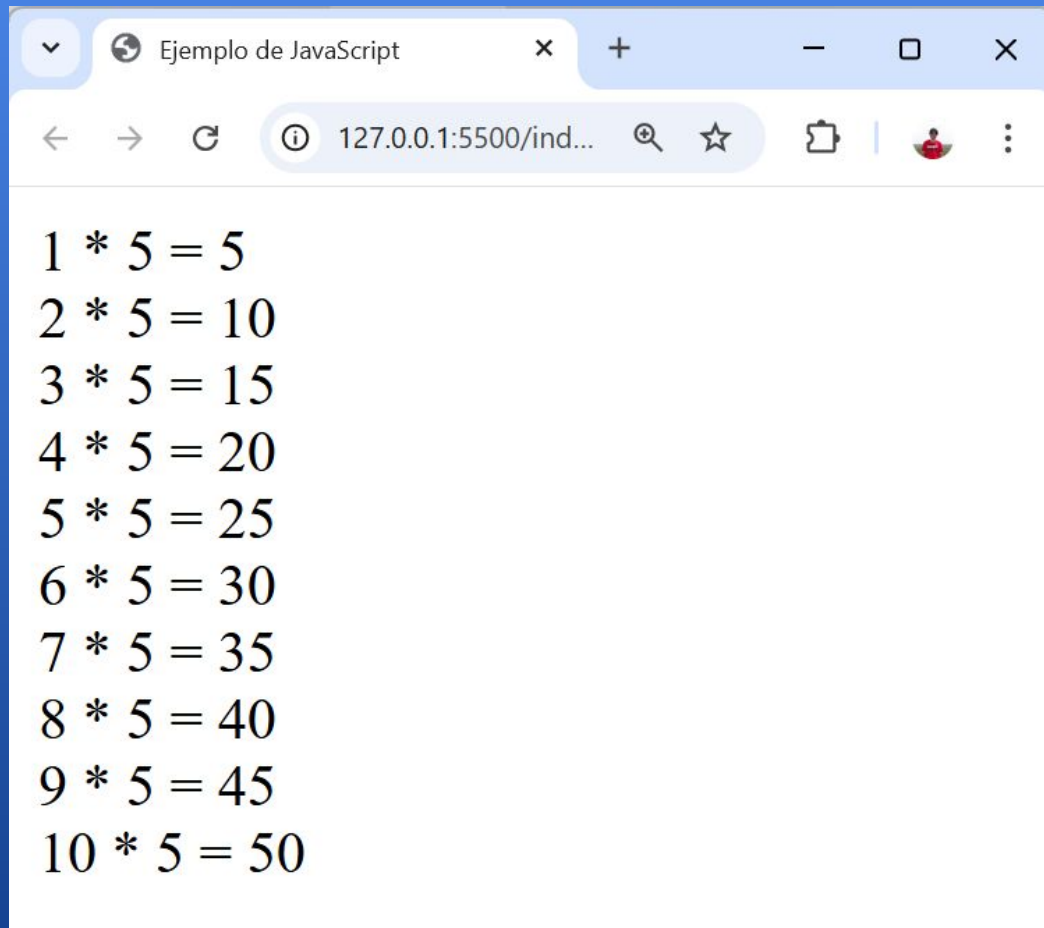
    calcularTabla();
  </script>

</body>
</html>
```

Acotaciones

Dentro de una función podemos definir variables locales como 'valor' y 'f'. También podemos ingresar datos por teclado.

Luego de ingresar el valor 5 por teclado tenemos como resultado en pantalla:



Problema

Confeccionar una función que solicite la carga de dos enteros (primero el menor y luego el mayor) y nos muestre desde el menor hasta el mayor de uno en uno.

Por ejemplo si ingresamos los valores 4 y 10 luego se debe mostrar por pantalla:

4 5 6 7 8 9 10

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>

<body>

  <script>
    function mostrarRango() {
      let inicio = parseInt(prompt('Ingrese valor inicial'))
      let fin = parseInt(prompt('Ingrese valor final'))
      for (let f = inicio; f <= fin; f++) {
        document.write(f + " - ")
      }

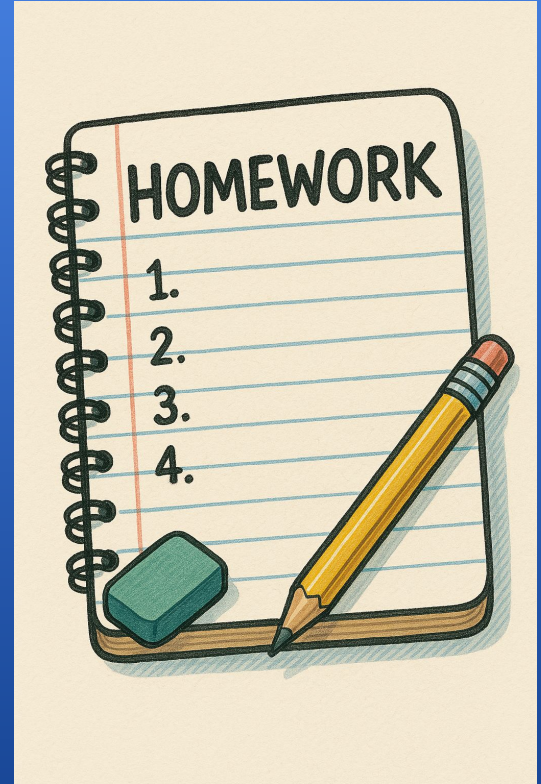
      mostrarRango();
    }
  </script>

</body>

</html>
```

TAREA 3 (GRUPAL)

Resolver los siguientes ejercicios en **GRUPOS DE 3 PERSONAS** y mandar los archivos de soluciones y un archivo de integrantes dentro de una carpeta comprimida para entregar en Educa:



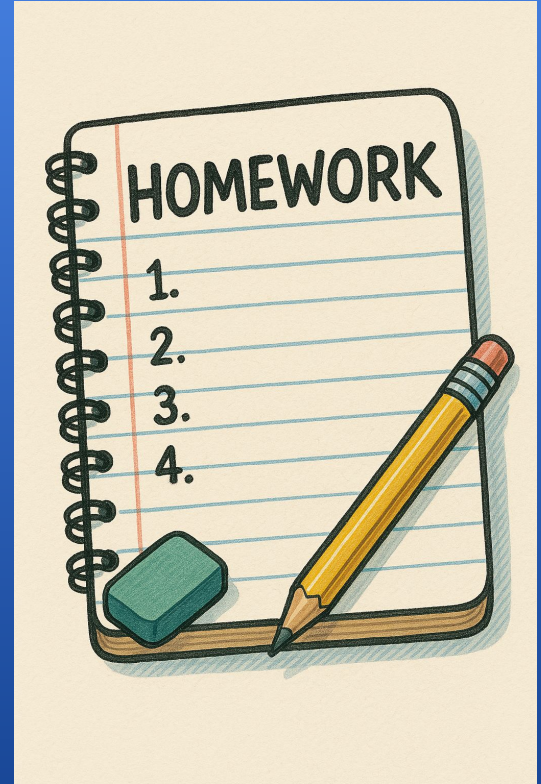
TAREA 3 (GRUPAL)

Problema 1: Calculadora de consumo eléctrico mensual

Enunciado:

Una familia quiere calcular cuánto gastó en electricidad en el mes. Saben cuántos kWh consumen por día y que el mes tiene 30 días. La tarifa por kWh es de ¢650. Se desea calcular:

- Consumo total del mes.
- Monto a pagar.
- Ahorro si se reduce 10% el consumo diario.



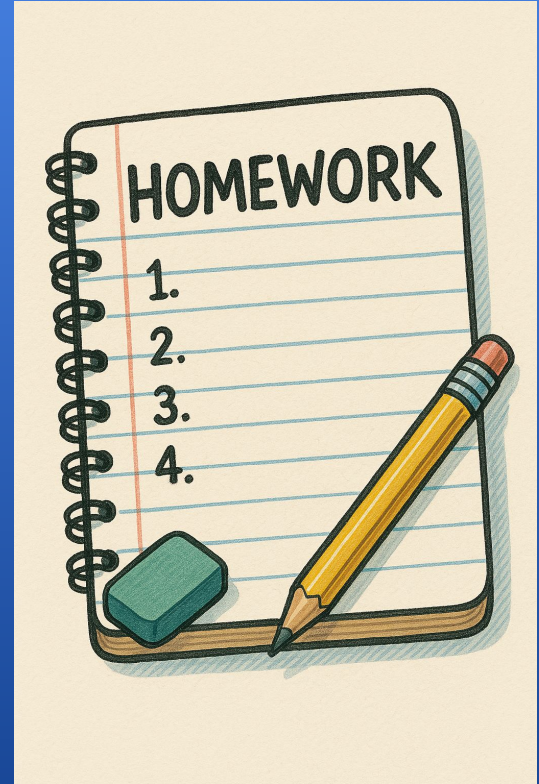
TAREA 3 (GRUPAL)

Problema 2: Costo total de imprimir apuntes

Enunciado:

Una imprenta cobra \$200 por cada página impresa. Un estudiante imprime 5 apuntes diferentes durante la semana, cada uno con una cantidad distinta de páginas. Se desea:

- Calcular el total gastado.
- Calcular cuánto hubiera gastado si imprimía todos a doble cara (mitad de páginas).
- Mostrar cuánto se ahorra con doble cara.



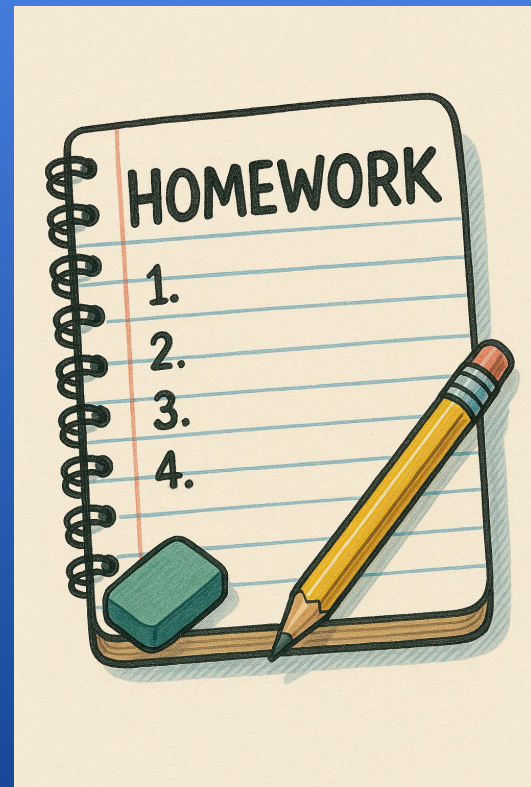
TAREA 3 (GRUPAL)

Problema 3: Cálculo de viáticos para docentes

Enunciado:

Un docente viaja a una sede 3 veces por semana. Cobra \$20.000 por día de traslado. Si va durante 4 semanas, ¿cuánto cobra en total?

Extra: ¿Cuánto cobraría si se aumentara el viático un 15% en la última semana?



EJERCICIOS ADICIONALES PROPUESTOS

— — —





**¡MUCHAS
GRACIAS!**