



Programación Web Básica.



U
N
Q



Prof. Jorge Nicolás Terradillos
Prof. Fernando Blanco

En la clase de hoy...

-Introducción a JavaScript

-Sintáxis básica de JS

-Tipos de datos

-Variables

-Arrays

-Funciones

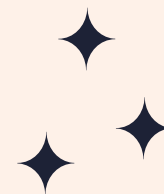




01

Introducción a JavaScript





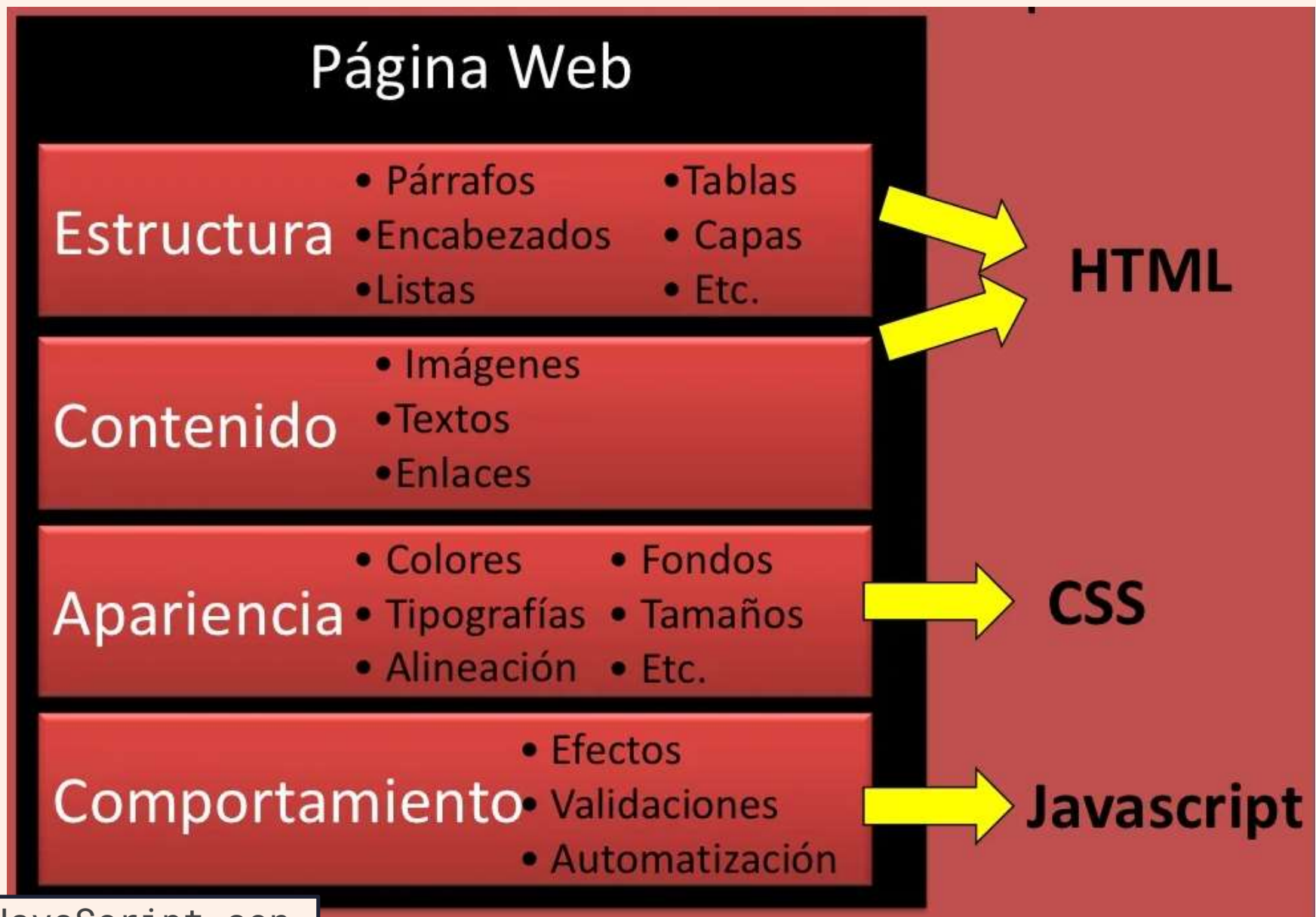
¿Qué es JavaScript?

Es un lenguaje de programación que permite modificar el contenido de una pagina web de forma dinámica

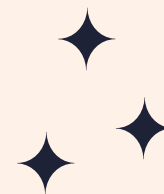
JS



Se suele decir que es la tercera “capa” del front-end de una página: con HTML creamos la estructura de una página, con CSS diseñamos el aspecto y con JavaScript se programa cómo reacciona la pagina a las interacciones, su comportamiento₄

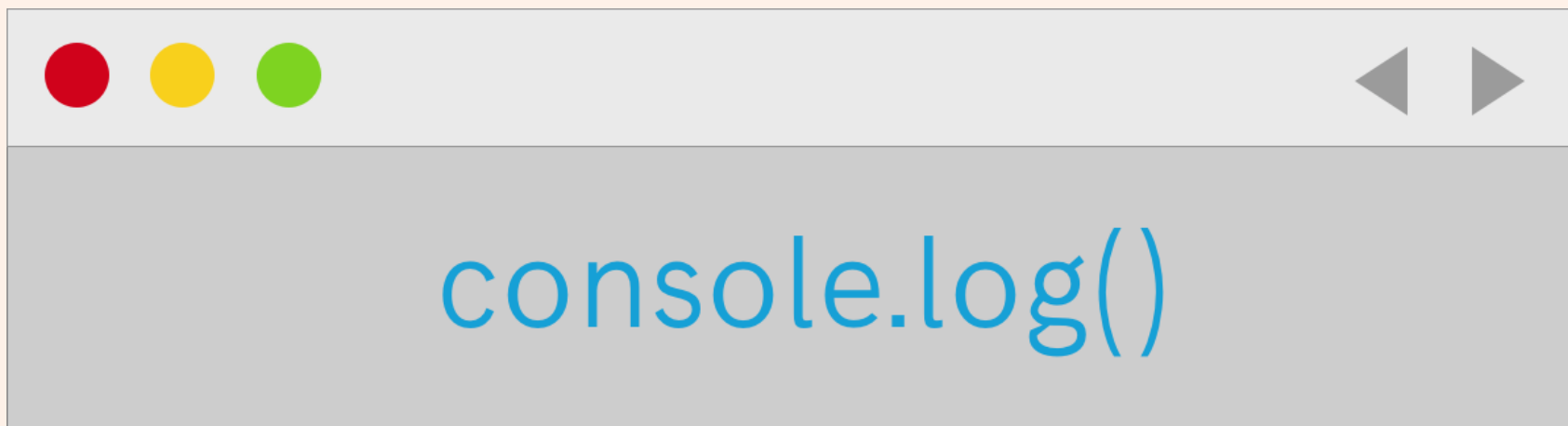


Nota: Java y JavaScript son distintos. Java es para desarrollo de software

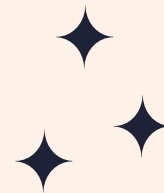
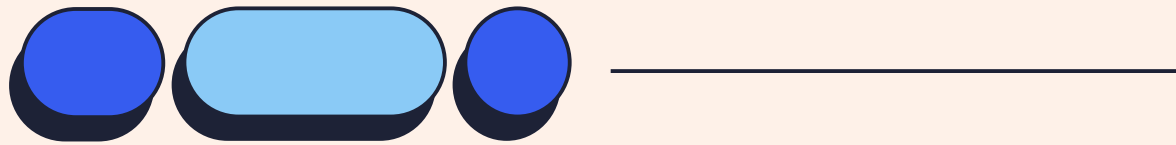


Consola

Para revisar la ejecución de algo en JavaScript, se suele acceder a la consola



Se usa el comando **console.log()** que hace visible en la salida o “output” del programa todo lo que coloquemos entre los parentesis “imprimiendolo”

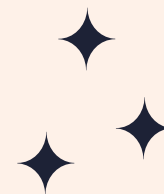


Elemento script

Comenzamos creando script (un programa simple) dentro de una página HTML, con el elemento **<script>**

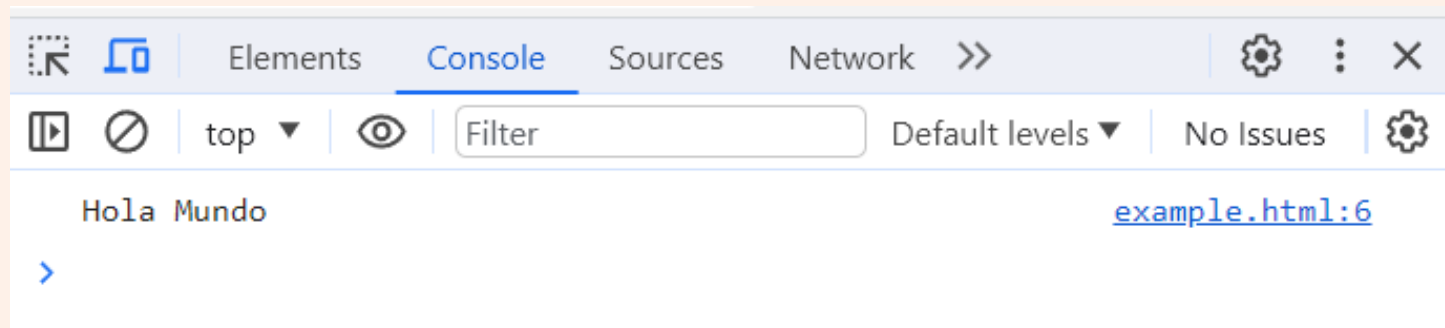
```
<script>  
    console.log("Hola Mundo!");  
</script>
```

Se puede colocar tanto el head como en body

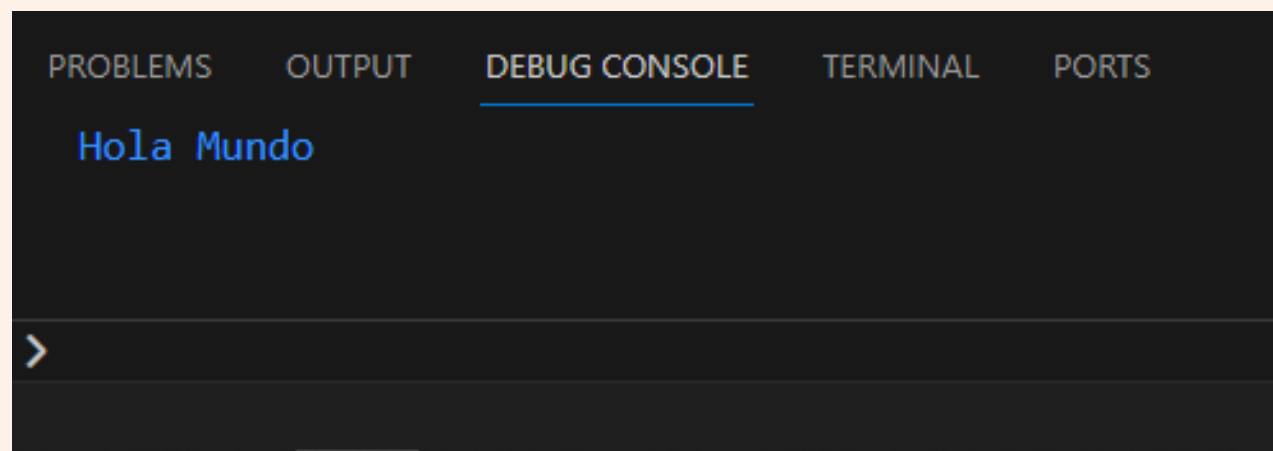


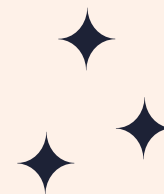
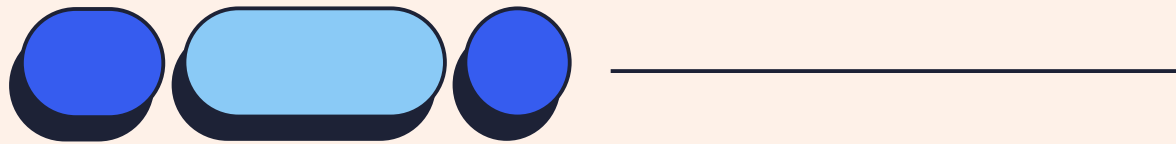
Ejecución

Al ejecutar algo en consola con JavaScript, nos aparecerá en la consola del programa que usamos (como el navegador)



En Visual Studio
abriremos la consola con
Ctrl+J y hacemos correr
el código con F5





Otra alternativa: Hay un editor de texto online en la web de freecodecamp. Nos logueamos con Google y vamos a la parte que dice 'Algoritmos de Javascript y Estructura de Datos':

(<https://www.freecodecamp.org/espanol/learn/javascript-algorithms-and-data-structures/>)



Comentarios

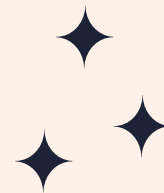
Igual que en HTML, para 1 línea usamos la doble barra

```
//este es un comentario
```

Comentario multilínea

```
/*este es un comentario  
multilínea*/
```

```
<script>  
  
    // Esto es un comentario  
  
    /*Esto también es  
    un comentario*/  
  
</script>
```

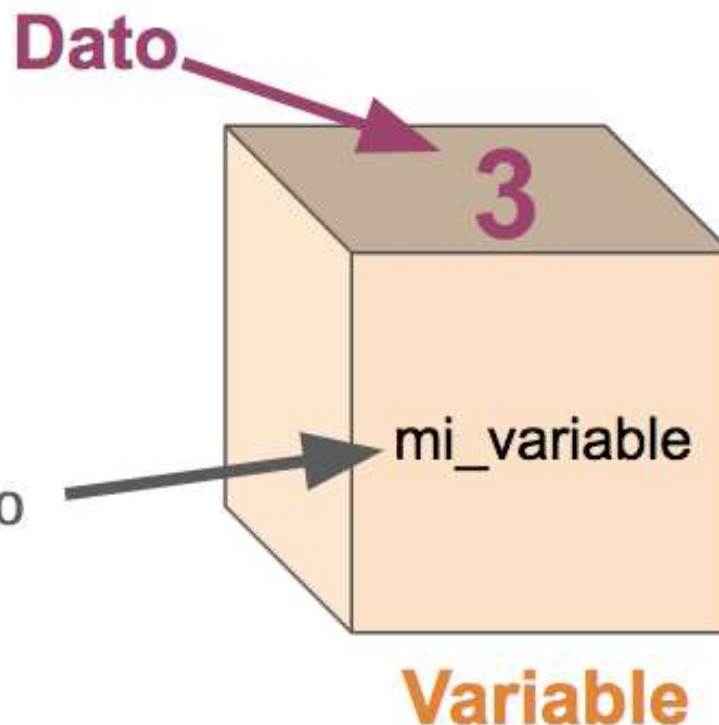


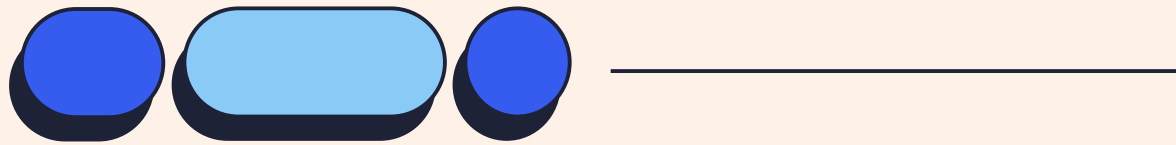
Variables

Las variables se pueden pensar como una caja donde podemos guardar un valor.

Variables JS

Cada variable tiene un nombre, de modo que podamos acceder a ese dato siempre que necesitemos.



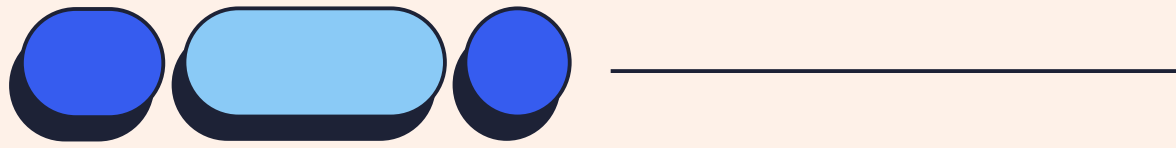


Sintaxis de una variable

En Javascript, uno de los tipos de variables disponibles es **“let”**

```
let miVariable = “curso de  
programación”;
```

El “=” es el operador de asignación, le asigna un valor a una variable.



Como ya lo tengo guardado , lo puedo usar en consola:

```
let miVariable = true;
```

o

```
let miVariable = 9;
```

```
console.log(miVariable);
```

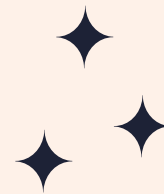
Se recomienda NOMBRES DESCRIPTIVOS, y VAMOS A USAR NOTACION
camel case (mayusculas para separar palabras)



Tipos de datos

En JS veremos datos de tipo number (números), string (letras), boolean (valores lógicos), object, null y undefined

```
let message = 'hello';  
message = 123456;  
let isChecked = true;  
let age = null;  
let x; //undefined
```



Inicializar variable

El primer paso para usar una variable es inicializarla

```
let a;           //creo la variable  
a = 2;           //le asigno un valor
```

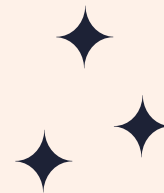
En un paso:

```
let a = 2;
```

Para usarlas :

```
console.log(a)
```





Asignar variable

El operador de asignación, como vimos, es «=». El valor por defecto de toda variable es undefined. También podemos asignar el valor de una variable a otra o sobrescribirla

```
let a = 19;
```

```
let b = a;
```

Aquí le asigno el valor que tiene la variable 'a' a la variable 'b'; la segunda variable valdrá 19

```
let x = 12;
```

//en este momento inicializo la variable x con el valor 12

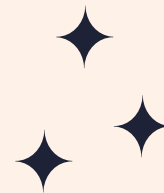
```
let miIdioma = "castellano";
```




02

Operaciones básicas





Aritmética

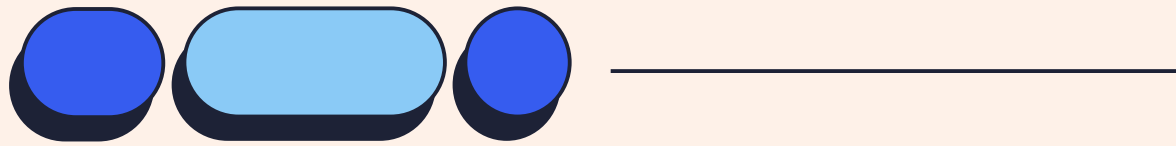
Varias operaciones matemáticas básicas son primitivas en JS

```
let suma = 7 + 12;    //declaramos e inciamos variable  
console.log(suma);
```

```
let resta = 15 - 6;  
console.log(resta);
```

Multiplicación:

```
let producto = 5 * 3;    //15  
//producto con 0  
let producto = 10 * 0;  
console.log(producto);
```



```
let cociente = 20 / 2;    //división
```

```
console.log(cociente);    //10
```

```
cociente = 3 / 0;    //infinity
```

```
//tambien podemos crear numeros decimales
```

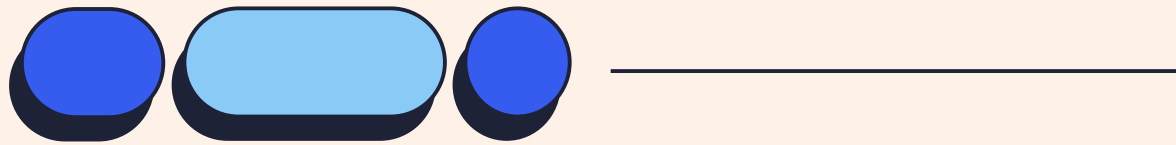
```
let miNumeroDecimal = 23.4;
```

```
let resto = 15 % 5;
```

```
//si dividimos 15 en 5, el cociente es 3 y el resto es 0
```

```
resto = 15 % 2
```

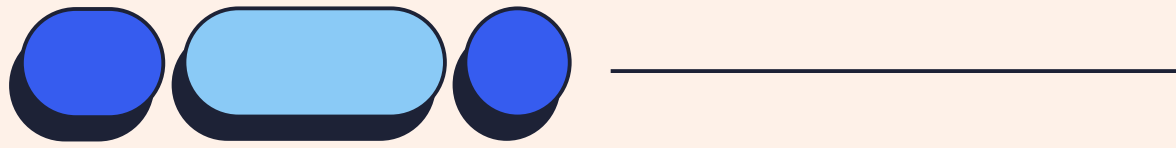
```
console.log(resto) // 1
```



Incrementar y disminuir

Supongamos una variable que guarda cuantos libros compré y quiero ir sumandole más:

```
let librosComprados = 112;  
console.log(librosComprados); //112  
librosComprados = librosComprados + 1; //opcion 1  
librosComprados ++ ; //opcion 2: asigna +1 al valor  
console.log(librosComprados);
```



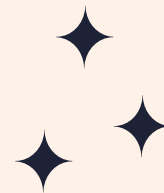
Así como incrementamos con «++», disminuimos con «--»

```
let numeroDeEstudiantes = 186;
```

```
numeroDeEstudiantes = numeroDeEstudiantes - 1; //opcion 1
```

```
numeroDeEstudiantes --; //opcion 2
```

```
console.log(numeroDeEstudiantes);
```



Asignar operaciones

```
let a = 23;
```

```
console.log(a); //23
```

```
//para incrementar el valor en 5 por ejemplo
```

```
a += 5;
```

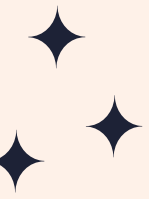
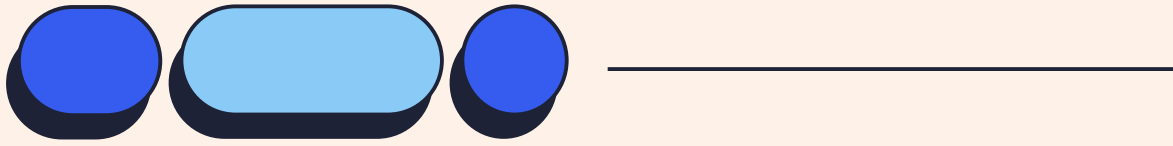
```
//esto incrementa en 5 la variable y luego la asigna. es como
```

```
//hacer a = a +5;
```

```
console.log(a); //28
```

```
let totalVentas = 1345.69;
```

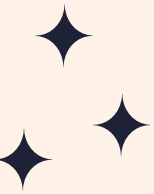
```
totalVentas += 345.29 ;
```



Podemos asignar resta y multiplicación de la misma forma

```
let totalDeudas = 1345.69;  
totalDeudas -= 210.11;
```

```
let c = 11;  
console.log(c); //11  
//para multiplicar el valor por 5  
c *= 5; //esto multiplica por 5 la variable y luego la  
asigna. es como hacer c = c * 5;  
console.log(c); //55
```



03

Strings





Strings

Los strings representan texto y se usan con comillas

```
let nombre = "Sergio";
```

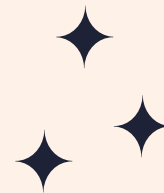
//todos los caracteres son parte de la variable:

guiones, espacios, sumas, números

```
let apellido = "Perez";
```

```
let nombreConError = 'luis';
```

//esto al compilar marca un error por las comillas.



Concatenar

Concatenar es unir varios strings juntos

```
let nombre = "Linus" + "Torvalds";
```

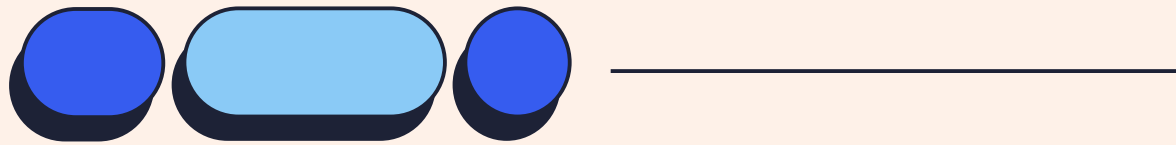
```
console.log(nombre); //LinusTorvalds
```

//para agregar espacio lo puedo poner en el apellido

```
nombre = "Linus" + " Torvalds";
```

//otra manera es agregar un string vacio

```
nombre = "Linus" + " " + "Torvalds";
```

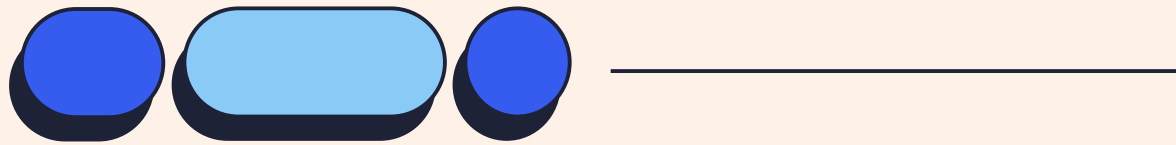


Concatenar variables

Entre las unidades relativas está el porcentaje y la unidad «em»

```
let mensaje = "Estoy aprendiendo a programar ";  
let parteFinal = "en la Bernalesa";  
//¿Como los concateno de forma rapida?
```

```
mensaje += parteFinal;  
console.log(mensaje);
```



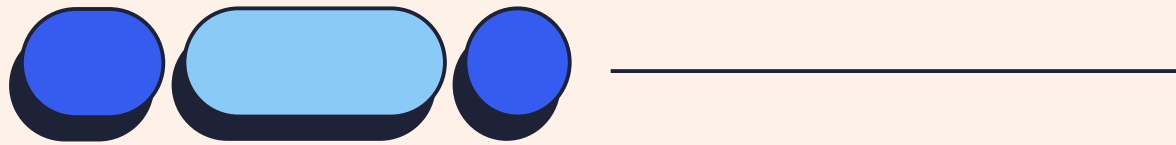
Longitud de strings

También puedo saber que tan largo es un string

```
let miCadena = "A";  
console.log(miCadena.length); //nos da 1
```

```
miCadena = "A B";  
console.log(miCadena.length); //los espacios cuentan  
//como caracter
```

```
miCadena = "Estoy aprendiendo a programar!";  
console.log(miCadena.length);
```



Notación strings

Para entender los strings, podemos verlos como un listado de espacios numerados

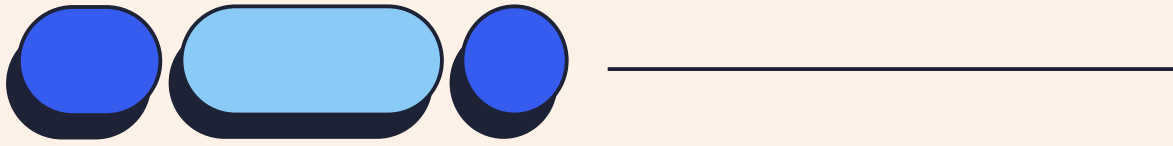
```
let lenguajeDeProgramacion = "Javascript";
```

```
/* J a v a s c r i p t
```

```
0 1 2 3 4 5 6 7 8 9    */
```

```
console.log(lenguajeDeProgramacion[0]); //imprime J
```

//dentro de los corchetes el índice que quieres acceder



```
let lenguajeDeProgramacion = "Javascript";
```

```
/*  J      a      v      a      s      c      r      i      p      t
    0      1      2      3      4      5      6      7      8      9      */
```

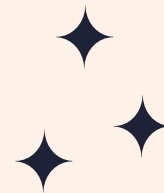
```
//como puedo accederlos?
```

```
console.log(lenguajeDeProgramacion[2]); //v
```

```
console.log(lenguajeDeProgramacion[3]); //a
```

```
console.log(lenguajeDeProgramacion[4]); //s
```

```
console.log(lenguajeDeProgramacion[10]); //undefined
```



Inmutabilidad

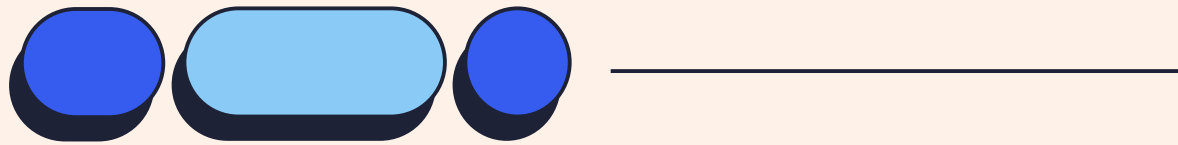
```
let lenguajeDeProgramacion = "Javascript";
```

```
//no podemos cambiar sus elementos individuales  
//si hago :
```

```
let saludo = "Jola, Mundo";
```

```
saludo[0] = "H"; //muestra error porque no puedo  
                // cambiar el caracter
```

```
let saludo = "Hola, Mundo"; //cambio todo el string
```



De derecha a izquierda, del final al principio

```
let miCadena = "Javascript";
```

//el penultimo es el la longitud - 2 porque comenzamos a contar desde 0.

```
console.log(miCadena[miCadena.length - 2]); //p
```

```
console.log(miCadena[miCadena.length - 3]); //i
```


Actividad

-Punto 1: Crear las siguientes variables

```
let miSustantivo = "perro";
let miAdjetivo = "negro";
let miVerbo = "corrio";
let miAdverbio = "rapidamente";
```

-Punto 2: Armar frase

```

    'el perro negro corrió rápidamente a la tienda' o
    'la bicicleta pequeña rodó lentamente a la avenida';
    probarla //console.log(palabrasEnFrase);

```

-Punto 3 Imprimir la longitud del mensaje

Guardar en una variable el texto completo. Imprimir en pantalla la longitud del texto completo





04

Arrays

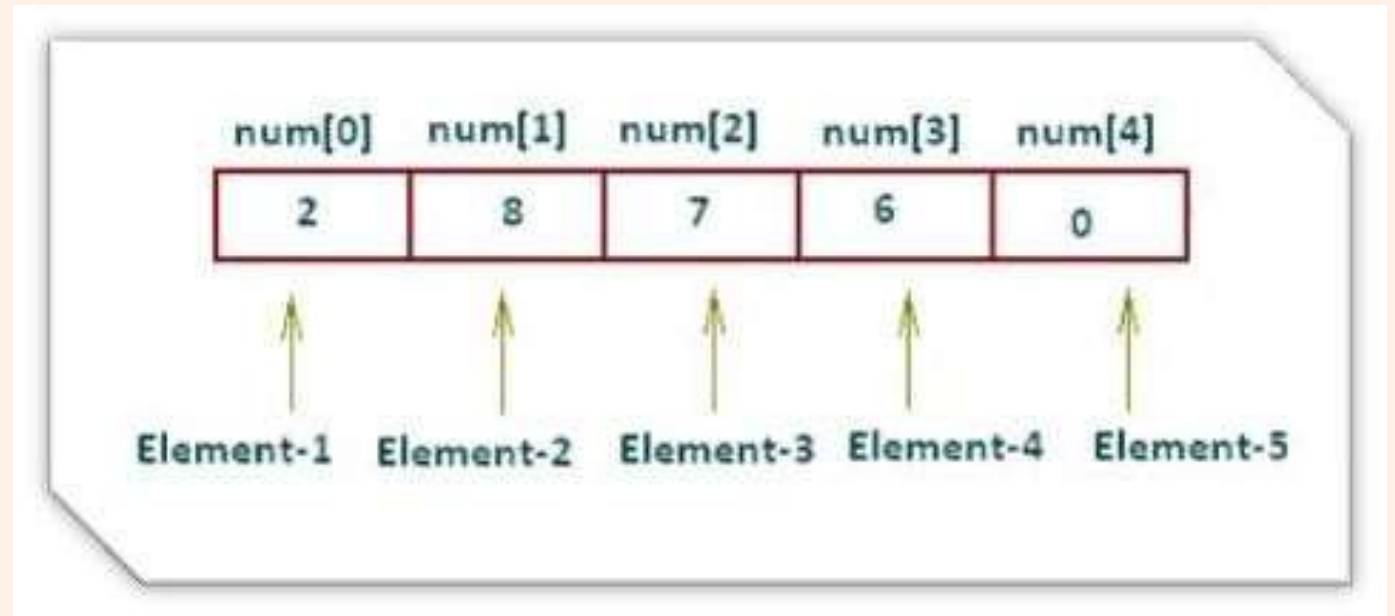


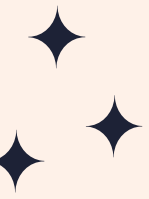
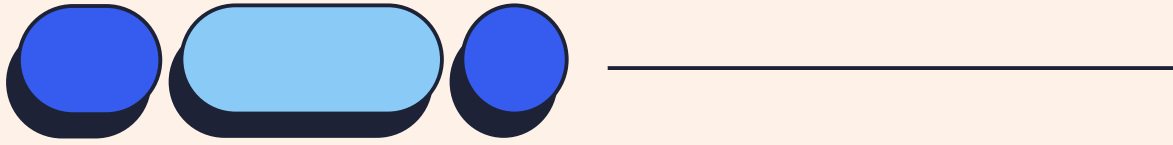


Arrays/arreglos

Un array es una estructura de datos que guarda varios datos como si fuera una lista. Tienen una extensión que se puede ampliar

Cada dato es un elemento dentro del array



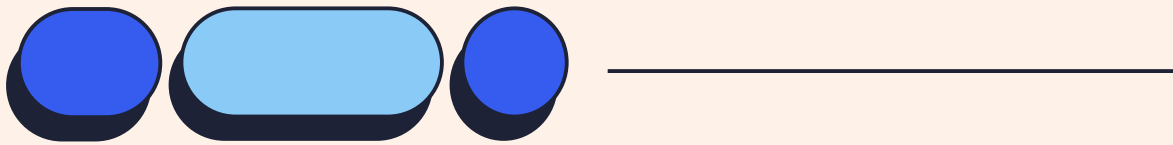


```
let miArray = ["Esteban", 24];  
console.log(miArray);
```

```
let estudiantes = ["juan", "candela", "matías"];  
console.log(estudiantes);
```

```
let notasDeExamen = [95, 75, 99];  
console.log(notasDeExamen);
```

Un array puede contener datos de distinto tipo



Podemos guardar los arreglos dentro de otro arreglo que los contenga:

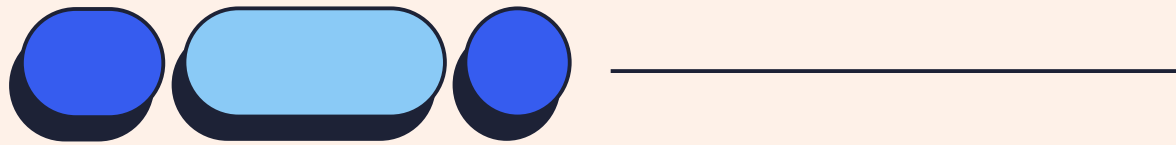
```
let listaDeEstudiantes = [ ["juan", 87], ["yesica", 79] ];
```

Si hago una lista de productos con precio y articulos:

```
let listaDeProductos = [ ["zapatos", 245.22, "s135"], ["celular", 5500, "s212"] ] ;
```

//cada array dentro del array principal describe un producto.

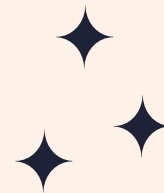
```
console.log(listaDeProductos);
```



Corchetes en arrays

Podemos acceder a los elementos de un array como hacemos con las letras en un string:

```
let miArreglo = [10, 20, 30];  
/* índice           0  1    2  */  
//para acceder al 10, usamos el índice 0  
console.log(miArreglo[0]); //10  
console.log(miArreglo[1]); //20  
console.log(miArreglo[2]); //30
```



Corchetes en arrays

Podemos acceder a los elementos de un array como hacemos con las letras en un string:

```
let miArreglo = [10, 20, 30];
```

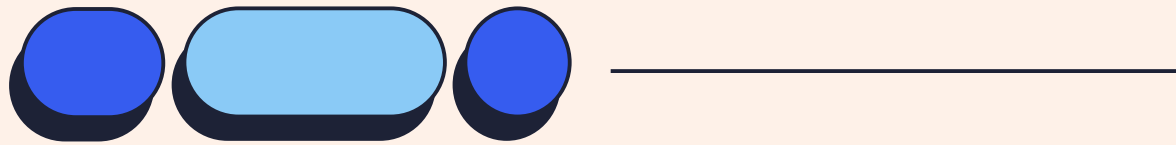
```
/* índice           0  1    2  */
```

```
//para acceder al 10, usamos el índice 0
```

```
console.log(miArreglo[0]); //10
```

```
console.log(miArreglo[1]); //20
```

```
console.log(miArreglo[2]); //30
```



Modificar arrays

En un array SI puedo modificar cada elemento individualmente

```
let miArreglo = [10,20,30,50];
```

//modificar es muy similar a acceder al elemento

```
miArreglo[0] = 15;
```

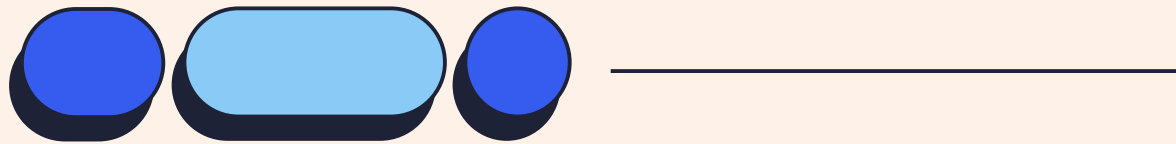
```
console.log(miArreglo); //resultado: [15,20,30,50]
```


Actividad

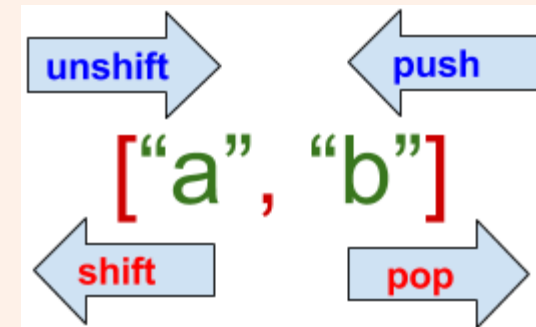
-Punto 1: Operar en un array

¿Cómo sumar los elementos del anterior array y mostrarlos en consola de forma que nos de 60, sin modificar ningún elemento del array?



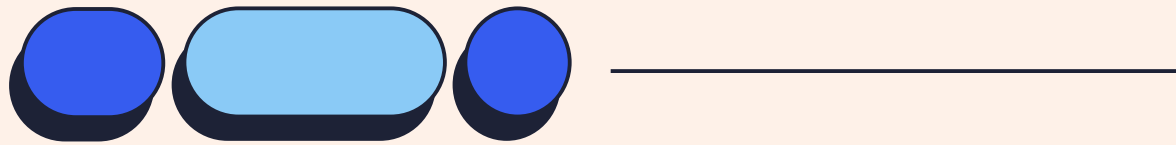


Operaciones en arrays: push



Los arrays también se modifican con ciertas operaciones o metodos. En caso de quere agregar un elemento usamos el metodo push

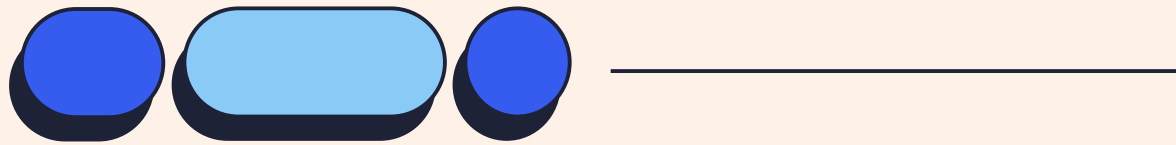
```
let estaciones = ["otoño", "invierno", "primavera"];
estaciones.push("verano");
console.log(estaciones);
//resultado: ["otoño", "invierno", "primavera", "verano"]
```



Pop

En los arreglos podemos borrar elementos. Vamos a usar el metodo **pop**

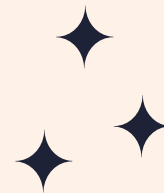
```
let estaciones = ["otoño", "invierno",  
"primavera", "verano"];  
estaciones.pop(); //borra el último elemento  
console.log(estaciones);
```



Shift

Para borrar el primer elemento usamos el método **shift**

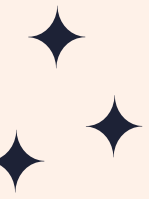
```
let estaciones = ["otoño", "invierno",  
"primavera", "verano"];  
estaciones.shift(); //borra el primer elemento  
console.log(estaciones);
```



Unshift

Para agregar un elemento al principio usamos `unshift`

```
let estaciones = ["otoño", "invierno",  
"primavera"];  
estaciones.unshift("verano");  
//el método agrega un elemento al principio  
console.log(estaciones);
```



Splice

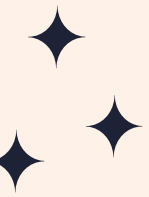
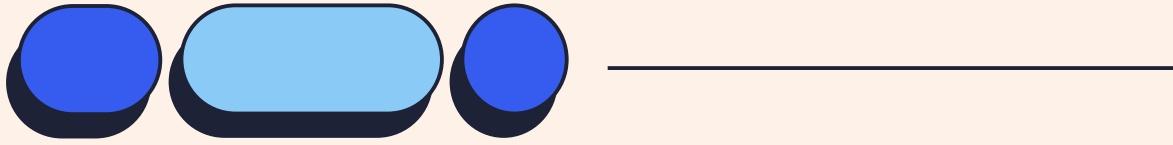
Si queremos eliminar solo ciertos elementos específicos, tenemos splice, un metodo que requiere varios datos

```
let estaciones = ["otoño", "invierno",  
"primavera", "verano"];
```

```
estaciones.splice(1);
```

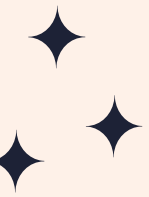
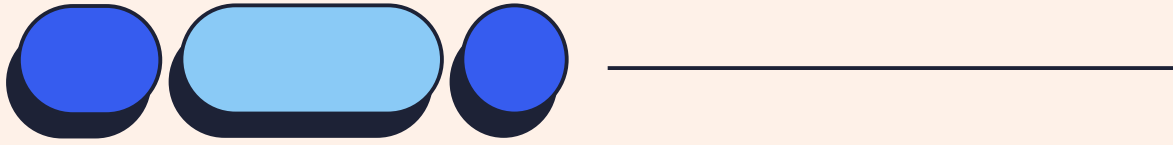
//aquí le pasamos el índice a partir del cual va a eliminar elementos

```
console.log(estaciones); //["otoño"]
```



Si queremos que elimine cierta cantidad de elementos, le pasamos un segundo número

```
let dias = ["lunes", "martes", "miercoles", "jueves", "viernes",  
"sabado", "domingo"];  
dias.splice(3, 2);  
/*empieza a eliminar a partir del índice 3 y quita 2  
elementos*/  
console.log(dias);  
/*["lunes", "martes", "miercoles", "sabado",  
"domingo"]*/
```



Finalmente, podemos usar splice para agregar elementos en un lugar determinado

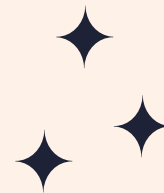
```
let dias = ["lunes", "martes", "miercoles", "jueves", "viernes",  
"sabado", "domingo"];  
dias.splice(6, 1, "osvaldo");  
/*elimina a partir del índice 6, quita 1 elemento y lo  
reemplaza con "osvaldo"*/  
console.log(dias);  
/*["lunes", "martes", "miercoles", "jueves",  
"viernes", "sabado", "osvaldo"]*/
```




05

Funciones en JavaScript

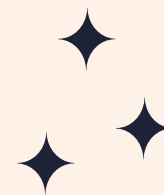




Concepto de funciones

Es un conjunto de instrucciones y comandos que, tomando ciertos valores de entrada, devuelve algún valor de salida o realiza cierta acción

```
function nombre_funcion(parametros){  
    instrucciones;  
}
```

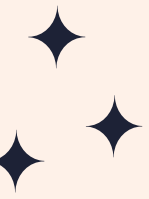
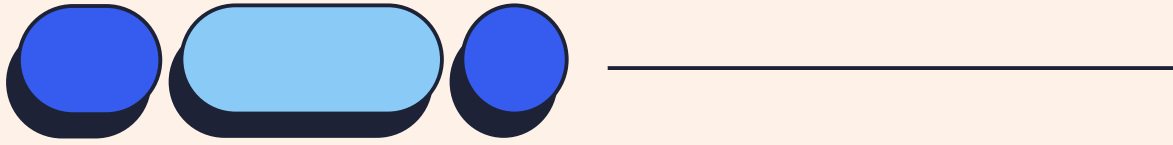


Sintaxis

Toda función tiene un nombre, seguido de ciertos parametros entre parentesis y luego su cuerpo o bloque de instrucciones entre llaves

```
function addNumbers(a, b) {  
    return a + b;  
}
```

NAME points to `addNumbers`
PARAMETERS points to `(a, b)`
BODY points to `return a + b;`



Para estructurar nuestro programa vamos a usar funciones

```
function mostrarMensaje() {  
    console.log("hola Mundo!");  
}
```

//como funciona. solamente haciendo:

```
mostrarMensaje();
```

```
mostrarMensaje();    //le decimos 'llamar' a la función.
```



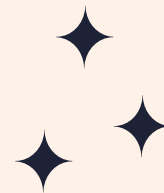
Parametros

Vamos a crear funciones que reciban valores (también llamados “parametros”)

```
function sumar(numero1, numero2) {  
    let suma = numero1 + numero2;  
    console.log("El resultado es " + suma);  
}
```

```
sumar(5, 3); //el resultado que se mostrará es 8;  
//al llamar a la función, cambiamos los parametros por valores concretos
```

Podemos crear variables dentro de la misma función para usarlas en el cuerpo de la misma



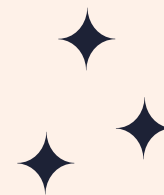
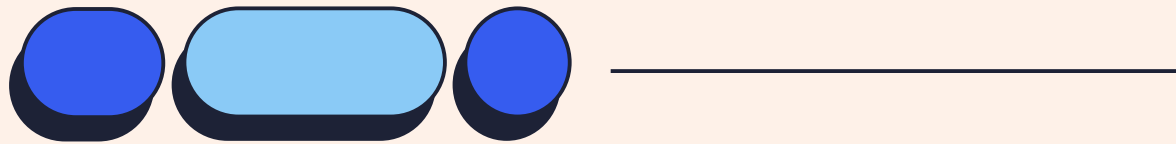
Argumentos

Los valores concretos que pasamos a una función en lugar de los parametros se llaman **argumentos**

```
sumar(5, 3);  
/*el 5 y el 3 son los  
argumentos en esta  
llamada*/
```

También puedo pasar variables como argumentos:

```
let x = 6;  
let y = 3;  
sumar(x, y); //resultado: 9
```

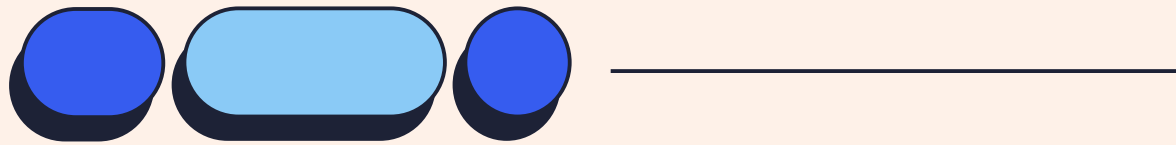


Retornar valores

Para mostrar un valor sin usar la consola en la función, usamos la palabra clave `return`

```
function sumar (a, b) {  
    return a + b;  
}
```

```
console.log(sumar(5, 3)); //8  
sumar(5, 3); //si la llamo muestra undefined
```



```
function retornarCadenasConMeta(lenguaje){  
  return ("mi meta es aprender "+ lenguaje);  
}
```

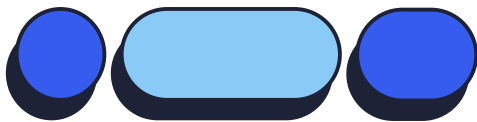
//podemos guardar funciones que retornan valores en variables

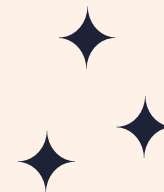
```
let miMeta = retornarCadenasConMeta("Javascript");  
console.log(miMeta);
```


Actividad

-Punto 1: Funciones sobre un array

Creemos una función llamada “elementoN” a la cuál le pasamos un array y un número, y nos retorna el elemento de ese array que se encuentra en el índice del número que damos. Por ej: si llamo a la función “elementoN([2,3,4,5], 2)” entonces quiero saber cual elemento está en la dirección “2” del arreglo.





Resumiendo

Sintaxis

Aprendimos lo básico para usar JavaScript: usar la consola y las variables "let"

Aritmética

Uso de cuentas básicas: suma, resta, división, multiplicación y resto, incrementar y disminuir

Strings

También usamos los datos en forma de texto y aprendimos como concatenar

Arrays

Listas ordenadas de datos podemos crear y modificar con metodos como push, pop, shift, etc.

Funciones

Para reutilizar codigos creamos funciones que pueden tener parametros y retornar valores





Eso es todo

Se escuchan dudas, quejas y
propuestas

F
i
n
d
e
l
a
c
l
a
s
e

