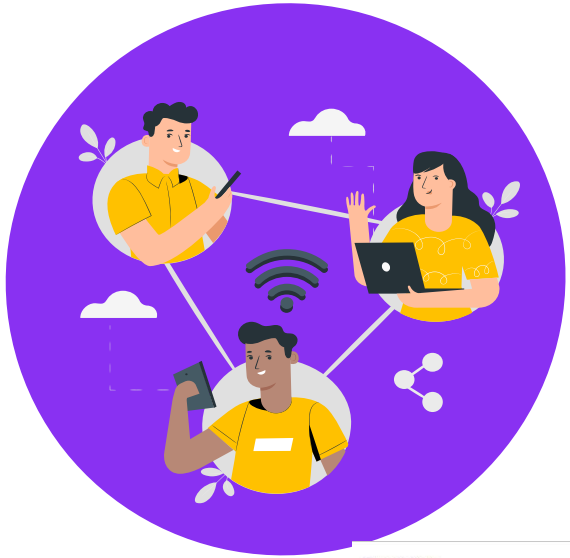
Maurisandev @MauriDeveloper maurisan4011@gmail.com

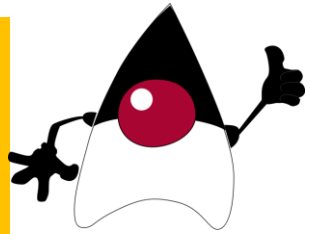


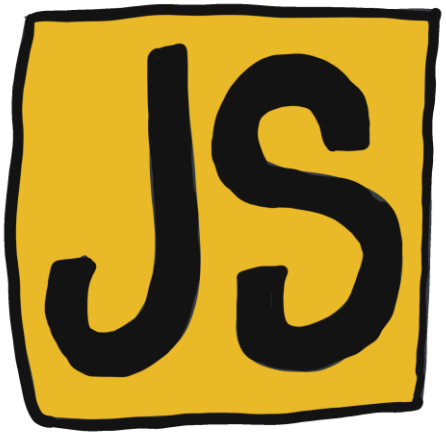
Curso Java FullStack

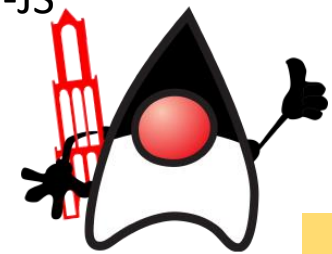
Codo a Codo 4.0

Clase-14

BOOTSTRAP-JS



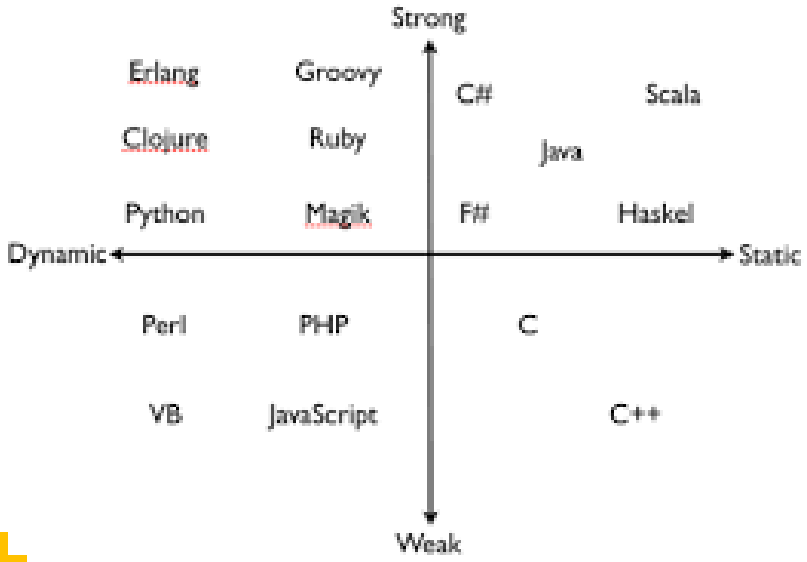




01

AULA VIRTUAL

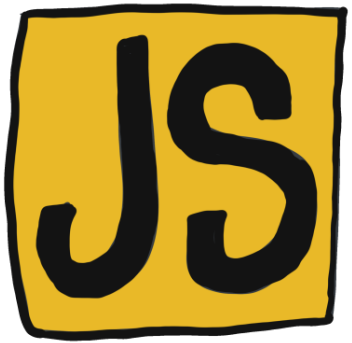
Tipado Dinámico



JavaScript cuenta con lo que se conoce como “dynamic tiping”, el cual nos permite a nosotros los programadores declarar variables sin indicar el tipo de dato que almacerá.

Al ejecutarse, JS resolverá automáticamente qué tipo de dato debe asignarle a cada variable, dependiendo de su valor.

SINTAXIS Y CÓDIGO



CÓDIGO JAVASCRIPT

JavaScript tiene sus **propias reglas** para la sintaxis, aunque respeta los estándares de muchos lenguajes de programación lógicos.

Existen dos maneras de escribir código en JavaScript.

¿CÓMO ESCRIBIR CÓDIGO JAVASCRIPT?

- **En un archivo individual con extensión .js**
  Ejemplo: mi-archivo.js

  Recuerda no utilizar espacios ni mayúsculas en los nombres de archivo.

JS - Formas de insertar código

- **Dentro de un archivo html, entre medio de las etiquetas <script>**

  Ejemplo:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> JAVA FULLSTACK CODO A CODO 4.0 </title>
</head>
<body>
    <h1>Agregamos JS dentro del documento HTML</h1>
    <script>
        console.log("Hola Mundo Alumnes CaC 4.0 ")
    </script>
</body>
</html>
```

SINTAXIS: REGLAS BÁSICAS

- **No se tienen en cuenta** los espacios en blanco y las nuevas líneas (al igual que HTML)
- **Se distinguen** las mayúsculas y minúsculas.
- **No se define** el tipo de las variables: numero, texto, etc
- Se pueden **incluir comentarios**:

```
<script>
    // Comentario simple: una linea
    /*Comentario de más de una linea*/
</script>
```

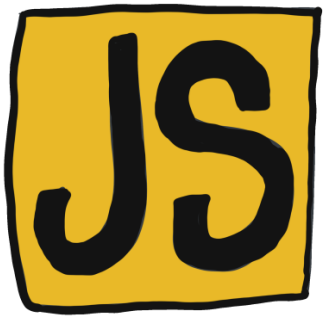
JS

SINTAXIS: PALABRAS RESERVADAS

- Palabras reservadas: son las palabras (en inglés) que se utilizan para **construir las sentencias** de JavaScript y que por tanto **no pueden ser utilizadas libremente**. Las palabras actualmente reservadas por JavaScript son:

```
break, case, catch, continue, default, delete,
do, else, finally, for, function, if, in,
instanceof, new, return, switch, this, throw,
try, typeof, var, void, while, with.
```

PROMPT, CONSOLA Y ALERT

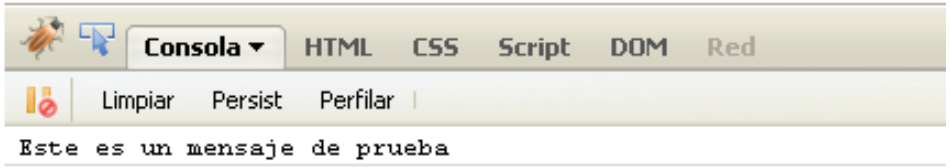


CONSOLA

La sentencia console.log() **muestra** el **mensaje** que pasemos como parámetro a la **llamada en la consola** JavaScript del Navegador web.

```
<script>
  console.log("Mensaje de prueba")
</script>
```

EJEMPLO DE console.log



En Chrome, la consola del navegador está disponible accediendo mediante:

Botón derecho sobre alguna parte de la web > Inspeccionar > Consola

ALERT

La sentencia alert() mostrará **una ventana** sobre la página web que estemos accediendo **mostrando** el **mensaje** que se pase como parámetro a la llamada.

```
<script>
    alert("Mensaje de prueba");
</script>
```

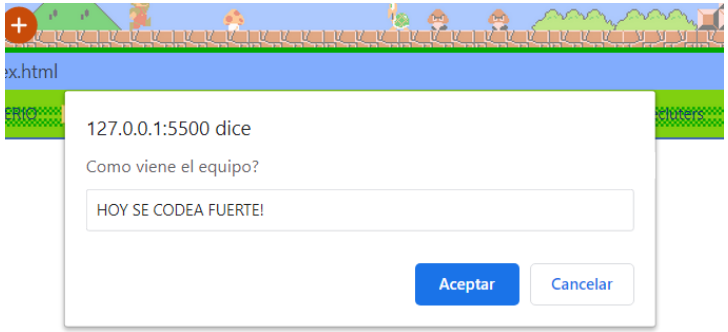
PROMPT

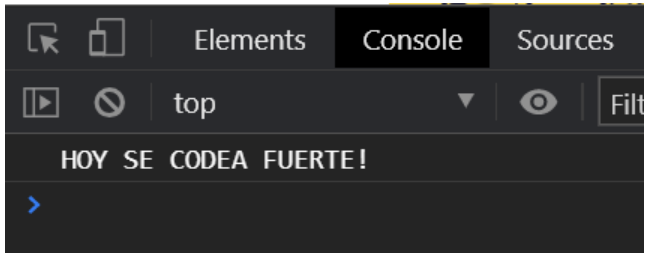
El mensaje mostrará un **cuadro de diálogo** para que el usuario **ingrese un dato**. Se puede proporcionar un mensaje que se colocará sobre el campo de texto. El valor que **devuelve** es una cadena que representa **lo que el usuario ingresó** en el formulario.

```
<script>
    var estadoTeam = prompt("Como viene el equipo?");
    console.log(estadoTeam)
</script>
```

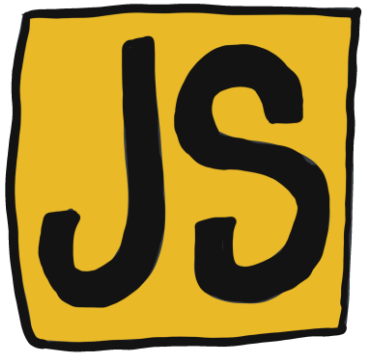
EJEMPLODEPROMPT

En la pantalla del navegador, el usuario verá una **ventana** sobre la web **solicitándole** un dato, tambien quiero que muestre en el console.log el texto ingresado.





VARIABLES



Variables

Una variable es un contenedor para almacenar un valor para utilizarlo múltiples veces en el código. Ocupa un espacio en la memoria de nuestra máquina.

Variables

Las variables se usan como nombres simbólicos para valores en nuestra aplicación. Los nombres de las variables se rigen por ciertas reglas: tienen que comenzar por una letra, un guion bajo o el símbolo de $, los valores subsiguientes pueden ser números, JavaScript diferencia entre mayúsculas y minúsculas, por lo tanto las letras incluyen desde la "A a la "Z" y desde la "a" a la "z".

Variables

Para asignarle un nombre a las variables o constantes (llamados también identificadores), deben cumplir las siguientes reglas:

- ✓ El nombre debe contener solo letras, dígitos o los símbolos $y _.
- ✓ El primer carácter no debe ser un número.

Nombres reservados:

- ✓ ver acá

Reglas de nombres

No podemos declarar una variable que comience con:

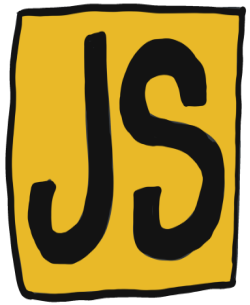
- Números
- Caracteres especiales
- Palabras reservadas de JS

```
> var 3test = 1;
Uncaught SyntaxError: Invalid or unexpected token
```

```
> var @miVariable = "Juan";
Uncaught SyntaxError: Invalid or unexpected token
```

Debemos hacerlo con:

- Guión bajo
- Signo dólar
- Cualquier palabra que no rompa las dos primeras reglas



Variables

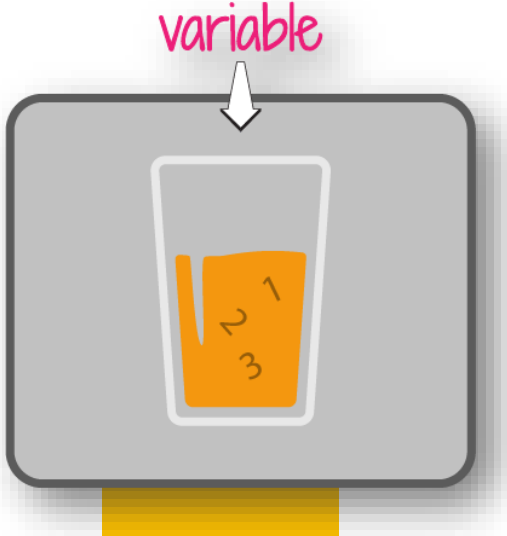
Hay 3 tipos de variables en JavaScript:

- ✓ **var:** declara una variable, iniciándola opcionalmente a un valor. Podrá cambiar el mismo y su scope es global o de función.
- ✓ **let:** declara una variable en un bloque de ámbito, iniciándola opcionalmente a un valor. Podrá cambiar su valor.
- ✓ **const:** declara una variable de sólo lectura en un bloque de ámbito. No será posible cambiar su valor mediante la asignación.

Var

var: las variables se hacen visibles en el ámbito global, es decir, que sin importar donde se definan, puede ser accedida desde cualquier parte del documento y permite que su valor pueda ser reasignado. El uso de ésta, puede dar a resultados inesperados, por eso, hay que tener cuidado de cómo se usa.

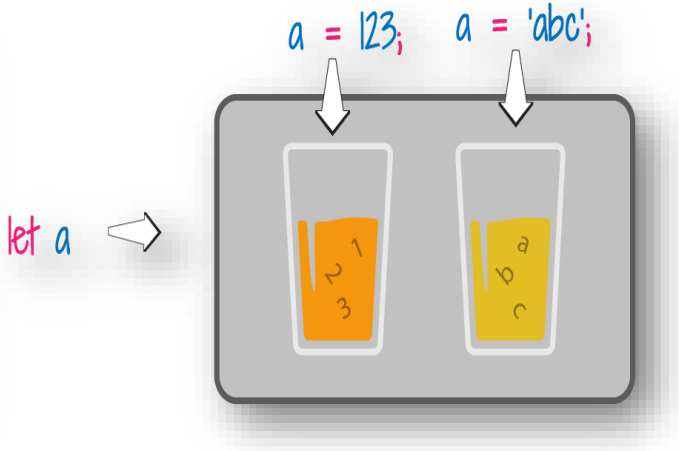
```
var nombreVariable;
var a;
var b;
```



Let

let: el alcance de estas variables, es que solo pueden ser accedidas dentro del bloque donde se definen. También, permiten que su valor pueda ser reasignado.

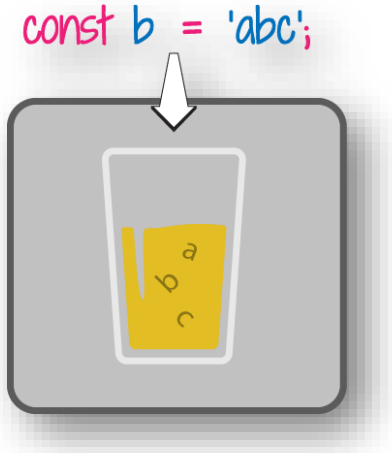
```
let nombreVariable = 'texto';
let a = 'abc';
a = 123;
let b = 1;
b = 5;
```



Const

const: estas variables (al igual que "let") solo pueden ser accedidas dentro del bloque donde están definidas, pero no permite que su valor sea reasignado, es decir, la variable se vuelve inmutable.

```
const nombreVariable = 'texto';
const a = 'Hola Mundo';
const b = 'abc';
const c = 123;
```



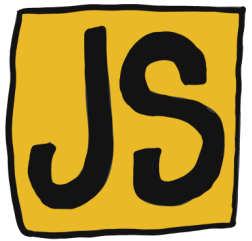
Ámbito de una variable

Cuando declaramos una variable fuera de una función se la denomina variable global. Cuando declaramos una variable dentro de una función se la denomina variable local, porque está disponible solo dentro de esa función donde fue creada.

Las variables en JavaScript pueden hacer referencia a una variable declarada más tarde. Este concepto se lo conoce como **hoisting**. Las variables son "elevadas" a la parte superior de la función, las variables que no se han inicializado todavía devolverán un valor **undefined**.

Tipos de Datos

- ✓ **String:** Secuencia de caracteres que representan un valor.
- ✓ **Number:** Valor numérico, entero, decimal, etc.
- ✓ **Boolean:** Valores true o false.
- ✓ **Null:** Valor nulo.
- ✓ **Undefined:** Valor sin definir.
- ✓ **Symbol:** Tipo de dato cuyos casos son únicos e inmutables.



challenge

https://www.bitdegree.org/learn/javascript-console-log

JS

Maurisandev @MauriDeveloper maurisan4011@gmail.com

MUCHAS GRACIAS!

****

NOS VEMOS EL mañana!!

<codoa codo/>