

# Bootcamp Front End, ReactJs: Lectura 1

## Lectura 2

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Febrero-2022

# Javascript

Javascript es un lenguaje de programación diferente a JAVA ya que está diseñado para ser interpretado por el navegador (al igual que html y css), por lo que no hay que descargar nada en el computador para usarlo.

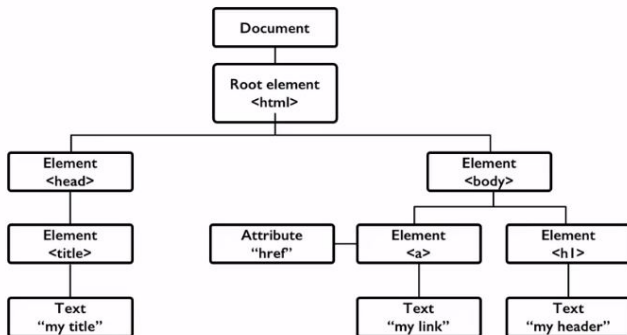
Javascript permite manejar los objetos del DOM de manera interactiva para hacer páginas web más profesionales y de mejor diseño.

La funcionalidad de javascript se establece a partir de **localizar elementos** y **establecer funciones**.

# DOM

El DOM es una metodología por la cual un documento de html se estructura para que los navegadores puedan interpretar sus elementos.

La metodología hace que se interprete al documento de html según un diagrama de árbol.



# API

Las interfaces de programación (APIs) son construcciones disponibles en un lenguaje de programación que permiten crear funcionalidades de manera simple. En palabras sencillas, son un conjunto de funciones predefinidas que se pueden llamar sin necesidad de usar todo un proceso para realizar la función.

Javascript funciona con APIs para poder controlar objetos en la página web, por lo que no se necesita hacer códigos "difíciles" para lograr un buen resultado.

Con Javascript tenemos APIs que permiten leer datos del DOM, manipular estilos CSS y modificar el HTML. Por ejemplo al mostrar una ventana emergente, estamos recurriendo a la API de javascript para ello. A este grupo de funciones se le conoce como API del navegador, ya que están conectadas al navegador.

# Vinculación

Al ser archivos propios de JavaScript, tienen que vincularse con el archivo html, que será el que leerá primero el navegador. Por lo tanto, lo fundamental es realizar una vinculación.

Existen 2 formas de vincularlo: Explícitamente, dentro de las etiquetas `<script></script>` dentro del head o dentro del body, y mediante un hipervínculo, dentro de la marca de entrada de la etiqueta `<script>`:

```
<script src="archivo.js"></script>
```

# Console.log()

La ventana de consola, dentro de un navegador, es una de las cosas más importantes para verificar problemas y manejar situaciones que se presentan en el lenguaje de programación.

para poder usar una respuesta del navegador, JavaScript implementa dentro de su API la función **console.log()**. Esta función permite escribir en la consola del navegador, números, expresiones, variables, tipo de objetos, etc.

Ejemplo de sintaxis:

```
console.log("Hola, bienvenido");
```

# Alert y Prompt

De vez en cuando, necesitaremos mostrar mensajes de alerta al cliente o recibir datos que deben ingresar.

Para poder escribir un mensaje de alerta al navegador, hacemos uso de la función **alert**. Esta función permite transmitir un mensaje al usuario, en un cuadro desplegable. Probemos la sintaxis:

```
alert("mensaje");
```

Para poder recibir un dato externo del cliente, podemos usar la función **prompt**. Por ejemplo la sintaxis:

```
prompt("nombre");
```

# Declaración de variables

Para declarar variables en JavaScript, tenemos que tener claro qué tipo de variable queremos definir.

En el 2012 todos los navegadores aceptaban la forma:

```
var nombre = ...
```

En el 2015, sin embargo, se añadieron las formas de declaración **let** y **const**.

La diferencia entre **let** y **var**, radica en la seguridad de redeclaración que no se encontraba con **var**, lo que generaba confunciones en código colaborativo.

**Const**, por contraparte, se usa para declarar elementos constantes.



# getElementById

Esta función del API de Javascript, permite obtener un objeto del DOM para que se pueda usar como variable. Miremos la sintaxis:

```
var elemento = document.getElementById("demo");
```

Al llamar un elemento del DOM podemos acceder a sus atributos, a sus estilos y a su contenido.

## inner.HTML

El contenido de un elemento de html, se podrá acceder mediante el método **inner.HTML**, dentro podemos escribir texto o definir una estructura de html que se visualizará. Ejemplo de sintaxis:

```
document.getElementById("demo").innerHTML = 5 + 6;
```

# Asignación vs Comparación

Las variables dentro de javascript se pueden asignar o comparar. Aunque estas palabras, suenen parecido, son muy distintas.

**Asignar** se utiliza para definir, declarar o modificar el valor de una variable, por lo que tiene sentido, sintácticamente, declarar:

```
let x= 5;  
x = x+2;  
console.log(x)
```

Aunque matemáticamente, cause dudas la segunda línea, en el ámbito de la programación es sintácticamente correcto, por lo que el resultado devolverá 7.

# Asignación vs Comparación

Para comparar, en programación escribimos `==`, el doble igual, que significa estrictamente **comparar**. En este sentido, al escribir:

```
let x = 5;  
console.log(x == x+2);
```

El resultado en la consola, nos devolverá un valor de False, ya que es falso que 5 sea igual a 7.

En el caso de javascript, existe la comparación fuerte, que es `===`, el triple igual, debido a que los tipos de variables pueden cambiar.

# Operadores

En JavaScript los operadores matemáticos, se encuentran en esta lista:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation ( <a href="#">ES2016</a> )
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement