



# JavaScript

LABORATORIO DE COMPUTACIÓN III

UTN-FRA

# Javascript - Características

Javascript es un lenguaje orientado a documento que ha surgido en un principio para extender las capacidades propias del lenguaje html.

Javascript es un lenguaje interpretado que se embebe en una página web.

El navegador es quien analiza e interpreta estas instrucciones en el momento que deben ser ejecutas.

Orientado a objetos y basado en prototipos

Es objetual, con tipificación débil y dinámica.

# Integración de código en el html

```
index.html x
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8      <script>
9          function Saludar(){
10              alert("Hola mundo");
11          }
12          Saludar();
13      </script>
14  </head>
15  <body>
16
17  <h1>Esto es un titulo h1</h1>
18
19  </body>
20  </html>
```

En el head

# Integración de código en el html II

```
index.html x
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7      <title>Document</title>
8    </head>
9    <body>
10     <h1>Esto es un titulo h1</h1>
11
12     <script>
13       function Saludar() {
14         alert("Hola mundo");
15       }
16       Saludar();
17     </script>
18   </body>
```

En el body

# Integración de código en el html III

```
index.html x JS scripts.js
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <script src="../js/scripts.js"></script>
8     <title>Document</title>
9   </head>
10  <body>
11    <h1>Esto es un titulo h1</h1>
12    <script>
13      Saludar();
14    </script>
15  </script>
16 </body>
17 </html>
```

En un archivo externo

```
index.html JS scripts.js x
1 function Saludar() {
2   alert("Hola mundo");
3 }
4
```

# Instrucciones de Salida

```
document.write("Hola mundo");
```

```
document.write("<h1>Hola mundo</h1>");
```

```
window.alert("Hola mundo");
```

```
document.getElementById("miParrafo").innerHTML = "Hola mundo";
```

```
document.getElementById("miDiv").innerHTML = "<h1>Hola mundo</h1>";
```

```
console.log("Texto en consola");
```

# Operadores de tipos

```
index.html JS scripts.js x
1
2  var cadena = "UTN FRA"; // string
3  var numero = 1234; // 1234
4  var bool = true; // boolean
5  var lista = [1, 2, 3, 4]; // object
6  var objeto = {nombre: "Juan", apellido: "Perez"}; // object
7  var fecha = new Date; // object
8  var miFuncion = function unaFuncion(){ alert("Hola");}; // function
9  var sinInicializar; // undefined
10 var vacia = null; // object
11
12 alert( typeof cadena);
```



# Funciones -- Sintaxis

```
index.html  JS scripts.js  x
1  function nombreFuncion( x, y ){
2      // desarrollo de la función
3  }
```

```
index.html  JS scripts.js  x
1  function Saludar( nombre ){
2      alert("Hola " + nombre);
3  }
4  var x = Saludar;
5  alert(x);
6  x("Jose");
7
```

Se puede asignar una función a  
Una variable y ejecutarla a través de esta



# Funciones Anónimas

```
JS script.js x
1 function (a, b) {
2
3     return a * b;
4 }
5
```

```
<> index.html JS script.js x
1
2
3 (function (){ alert("Hola mundo");})();
4
5
6
```

Función anónima auto invocada

# Función Constructora

JS script.js x

```
1
2  var miFuncion = new Function ( "a", "b", "return a*b");
3
4  var rta = miFuncion(20, 5);
5
6  alert(rta);  // 100
7
8
```

# Parámetros y Argumentos

- Una función puede o no recibir parámetros
- Parámetros son los nombres que aparecen en la definición de una función
- Argumentos son los valores que se le pasan a una función en la llamada
- No se especifican los tipos de los parámetros
- No se verifican los tipos de los argumentos
- No se comprueba el número de los argumentos recibidos

**Parámetros por defecto:** Si se llama a una función con menos argumentos de los declarados. Los valores que faltan no están definidos

```
<> index.html    JS script.js    x
1  function sumar(a, b){
2
3      if(!b){
4          b = 0;
5      }
6      return a + b;
7  }
8
9  alert(sumar(4));
10
```

# Parámetros y Argumentos II

```
<> index.html  JS script.js  x
1  function miFuncion( ){
2
3      alert( arguments.length);
4
5  }
6  miFuncion(4,5,6,3,2);    // 5
7
8
```

**Objeto arguments:** Es un objeto que guarda los argumentos recibidos en la llamada a la función

```
<> index.html  JS script.js  x
1  function miFuncion( ){
2
3      console.log( arguments );
4
5  }
6  miFuncion(4,5,6,3,2);    // [object Arguments]: {0: 4, 1: 5, 2: 6, 3: 3, 4: 2...}
7
8
```

# Objetos

```
<> index.html  JS script.js  x
1  // Datos Primitivos
2  var cadena = "Juan";
3  var sueldo = 10000.50;
4  var bool = true;
5  var x = undefined;
6  var y = null;
7
8  // Objetos Nativos (No dependen del navegador)  Objetos de Alto Nive (Dependen del navegador)
9  /*
10 - String                                - Window
11 - Number                                - Screen
12 - Boolean                                - Navigator
13 - Data                                  - Location
14 - Math                                  - History
15 - RegExp (Expresiones regulares)        - document
16 - Array
17 - Function
18 - Object
19 */
```

# Objetos II

```
<> index.html  JS script.js  x
1
2  var superHeros = { nombre: "Superman", editorial: "DC Comics", origen: "Krypton"};
3
4  /*      -Propiedades: Se definen en modo nombre : valor
5          -Metodos: Acciones que se pueden ejecutar sobre un objeto
6  */
7
```

# Objetos Definición

```
<> index.html    JS script.js    x
1  // - Objeto Literal
2  var persona1 ={};
3  var persona2 = {nombre:"Juan", apellido:"Perez", edad: 30};
4
5  // - Objeto Simple utilizando new
6  var persona2 = new Object();
7  persona2.nombre = "Juan";
8  persona2.apellido = "Perez";
9  persona2.edad = 30;
10
11 // -Definir un constructor de objeto
12 function Persona (nombre, apellido, edad){
13     this.nombre = nombre;
14     this.apellido = apellido;
15     this.edad = edad;
16 }
17
18 var p1 = new Persona("Juan", "Perez", 30);
19 var p2 = new Persona("Ana", "Gomez", 25);
20
```



# Objetos Propiedades


```
<> index.html    JS script.js    x
1  /* Sintaxis de acceso a las propiedades de un objeto
2      - objeto.propiedad;
3      - objeto["propiedad"];
4      - objeto[expresion];                                */
5
6  var persona1 = {nombre: "Juan", apellido: "Perez", edad: 30};
7
8  alert(persona1.nombre);
9
10 alert(persona1["apellido"]);
11
12 var prop = "edad";
13 alert(persona1[prop]);
14
```

# Objetos Propiedades II - bucle for in

Se utiliza para movernos por las propiedades de un objeto

```
<> index.html    JS script.js    x
1  /* Recorrer las propiedades de un objeto
2      for(<nombre_variable> in <nombre_objeto>){
3          |      <nombre_objeto>[<nombre_variable>];
4          |      }
5  */
6  var superHeroe = {nombre: "Superman", editorial: "DC Comics", origen: "Krypton"};
7
8  for(propiedad in superHeroe){
9      |      alert(superHeroe[propiedad]);
10 }
```

# Objetos – Métodos

```
<> index.html  JS script.js  x  
```

```
1  /* Sintaxis de creación de métodos de un objeto
2      <nombre_metodo> : function(){
3          Sentencias
4      }
5  */
6  var p1 = { nombre:"Juan",
7              apellido:"Perez",
8              Saludar: function(){
9                  return "Hola, soy " + this.nombre + " " + this.apellido;
10             }
11          };
12
13  alert(p1.Saludar());
14
```

# Objetos – Agregar propiedad

```
<> index.html  JS script.js  x
1
2  function Persona (nombre, apellido, edad){
3      this.nombre = nombre;
4      this.apellido = apellido;
5      this.edad = edad;
6  }
7  var p1 = new Persona("Juan", "Perez", 30);
8  var p2 = new Persona("Ana", "Gomez", 25);
9
10 // Añadir propiedad a un objeto
11 p1.pais = "Argentina";
12 alert(p1.pais); // Argentina
13 alert(p2.pais); // undefined
14
```

La propiedad se agrega solo a la instancia

# Objetos – Prototype

<> index.html

JS script.js

×

```
1
2  function Persona (nombre, apellido, edad){
3      this.nombre = nombre;
4      this.apellido = apellido;
5      this.edad = edad;
6  }
7  var p1 = new Persona("Juan", "Perez", 30);
8  var p2 = new Persona("Ana", "Gomez", 25);
9
10 // Añadir propiedad al prototipo
11 // <nombre_objeto>.prototype.<nombre_propiedad>
12
13 Persona.prototype.nacionalidad = "Argentina";
14
15 alert(p1.nacionalidad);
16 alert(p2.nacionalidad);
17
```

# Manejador de eventos en línea

```
<> index.html x JS script.js
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <script src="./script.js"></script>
9      <title>Manejador de eventos</title>
10 </head>
11
12 <body>
13     <!-- Cada elemento XHTML tiene sus posibles eventos como propiedades
14         Puede tener un evento de cada tipo.
15         El nombre del evento es "on" seguidodel nombre de la acción -->
16
17     <!-- Manejador como atributo de una etiqueta XHTML -->
18
19     <h3 id="mih3" onclick="this.innerHTML = 'Hiciste click'"
20         onmouseover="this.style.color = 'blue';"
21         onmouseout=" this.style.color = 'black'">
22         Esto es un titulo h3
23     </h3>
24 </body>
25
26 </html>
```



# Manejador con funciones externas

```
<> index.html x JS script.js
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <script src="./script.js"></script>
8   <title>Manejador de eventos</title>
9 </head>
10 <body>
11   <!-- Manejador con funciones externas -->
12
13   <h3 id="mih3" onclick="cambiarTexto(this);">Esto es un titulo h3</h3>
14 </body>
15 </html>
```

```
<> index.html JS script.js x
1
2 function cambiarTexto(e){
3   e.innerHTML = "Hiciste click";
4 }
5
```



# Manejador Tradicional

<> index.html x JS script.js

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Manejador de eventos</title>
8 </head>
9 <body>
10   <!-- Manejador Tradicional -->
11   <h3 id="mih3">Esto es un titulo h3</h3>
12
13   <script src="./script.js"></script>
14 </body>
15 </html>
```

<> index.html

JS script.js

x

```
1 var tituloH3 = document.getElementById("mih3");
2
3 tituloH3.onclick = cambiarTexto;
4
5 function cambiarTexto(){
6   tituloH3.innerHTML = "Me cambiaron el texto";
7 }
8
```

# Manejador Tradicional 2

```
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script src="._/script.js"></script>
  <title>Manejador de eventos</title>
</head>
<body>
  <!-- Manjeador Tradicional 2-->
  <h3 id="mih3">Esto es un titulo h3</h3>
</body>
</html>
```

```
window.onload = function (){
    document.getElementById("mih3").onclick = cambiarTexto;
}

function cambiarTexto(){
    document.getElementById("mih3").innerHTML = "Me cambiaron el texto";
}
```

# Manejador del w3c

```
<> index.html    JS script.js    x
1  var tituloH3;
2
3  window.addEventListener("load", inicializarEventos);
4
5  function inicializarEventos(){
6      tituloH3 = document.getElementById("mih3");
7      tituloH3.addEventListener("click", cambiarTexto);
8      tituloH3.addEventListener("click", cambiarColor);
9  }
10 function cambiarTexto(){
11     tituloH3.innerHTML = "Me cambiaron el texto";
12 }
13 function cambiarColor(){
14     tituloH3.style.color = "blue";
15 }
16
17 // tituloH3.removeEventListener("click", cambiarTexto);
18
```

# Manejador anónimo del w3c

```
<> index.html  JS script.js  x
1  var tituloH3;
2
3  window.addEventListener("load", inicializarEventos);
4
5  function inicializarEventos(){
6      tituloH3 = document.getElementById("mih3");
7      tituloH3.addEventListener("click", function(){
8          this.innerHTML = "Me cambiaron el texto";
9      });
10
11  }
12
```



# Obtener información de un evento

```
var enlace;  
  
window.addEventListener('load', inicializarEventos);  
  
function inicializarEventos(){  
    enlace = document.getElementById("enlace");  
    enlace.addEventListener('click', manejador);  
    enlace.addEventListener('mouseover', manejador);  
    enlace.addEventListener('mouseout', manejador);  
}  
  
function manejador(e){  
    console.log(e.type);  
    console.log(e.target.id);  
}
```