



CENTRO DE ESTUDIOS TECNOLOGICOS

Industrial y de Servicios No. 66

Cuadernillo de Aprendizajes Esenciales

Semestre febrero - julio 2021

Submodulo 2.- Desarrolla aplicaciones que se ejecutan en el cliente.

Semestre: 4

Maestro: María del Socorro Juárez Soto.

Nombre del alumno: _____

Grupo _____ **Especialidad** _____ **Turno** _____

Teléfono del alumno(a): _____

Correo del alumno: _____

San Pedro Garza García, N. L. a febrero de 2021.

Asignatura o submódulo:		Semestre:	Carrera:
Módulo III.- DESARROLLA APLICACIONES WEB Submodulo II.- Desarrolla aplicaciones que se ejecutan en el cliente.		4°	Programación
Competencias Genéricas y atributos	G4.1 Expresa ideas y conceptos mediante representaciones lingüísticas, matemáticas o gráficas. G5.2 Ordena información de acuerdo a categorías, jerarquías y relaciones. 8. Trabaja en forma colaborativa. G8.3 Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta dentro de distintos equipos de trabajo.		
Competencias Disciplinares	C1 Identifica, ordena e interpreta las ideas, datos y conceptos explícitos e implícitos en un texto, considerando el contexto en el que se generó y en el que se recibe. -Expresa ideas y conceptos en composiciones coherentes y creativas, con introducción, desarrollo y conclusiones claras. -Utiliza las TIC's para investigar, resolver problemas, producir materiales y transmitir información.		
Competencias profesionales	Identifica los elementos de JavaScript para manejar eventos		

CALENDARIO DE ACTIVIDADES			
PRIMERA EVALUACIÓN PARCIAL			
Aprendizajes Esenciales	Estrategias de aprendizaje y Producto a evaluar	Fecha entrega de actividad	Ponderación
Comprende:	Desarrolla aplicaciones que se ejecutan en el cliente: Después de consultar la documentación proporcionada por el docente dentro de este cuadernillo, realiza las siguientes actividades en la libreta, toma fotografía, colócalas en un documento Word que contenga portada: nombre del plantel, nombre de la asignatura, nombre de la actividad, nombre del maestro, nombre del alumno, carrera, grupo y semestre, realiza la entrega en el plantel o envíalas a través de la plataforma Edmodo., código de acceso: fg522n Resuelve la evaluación diagnostica en tu cuaderno de trabajo.		
Contenido Apertura: Evaluación Diagnóstica	Realiza un mapa mental en el cuaderno de trabajo con la información proporcionada en la documentación adjunta.	16 de Febrero 2021	25%
Contenido Desarrollo: Unidad 1 1.1 Qué es JavaScript 1.2 Características y funcionamiento del lenguaje JavaScript 1.3 Clientes y Navegadores 1.4 URL y Recurso	Revisa la información del editor para la construcción del código o bien identifica en tu computadora el bloc de notas que incluye Windows. Elabora un informe escrito en su libreta, de la herramienta que va a utilizar para la escritura del código.	18 de Febrero 2021	25%
Unidad 2 - Equipo y Software a utilizar	Realiza un resumen en tu cuaderno de trabajo de la Introducción a JavaScript	25 de Febrero 2021	25%
Unidad 3 1.5 Introducción a JavaScript (Tipos, objetos y valores)		04 de Marzo 2021	25%

<p>1.6 Operadores y expresiones 1.7 Sobrecarga de operadores 1.8 Conversión de tipos en expresiones 1.9 Operador typeof 1.10 Operadores JavaScript 1.11 Comentarios</p> <p>Contenido Cierre: Unidad 4</p> <p>1.12 Práctica Guiada Aplicación WEB utilizando HTML, CSS y JavaScript</p>	<p>Realiza práctica guiada 1 escrita en tu libreta con el fin de identificar la sintaxis del lenguaje JavaScript y la estructura del desarrollo web básico. Emplea HTML, CSS e Inserta un Script utilizando las etiquetas <script> y </script></p>  <pre>function nombre_funcion(parametros){ instrucciones; }</pre>	<p>11 de Marzo 2021</p>	<p>25%</p>
---	--	-------------------------	------------

EVALUACION DIAGNOSTICA JAVASCRIPT

- 1- ¿En qué lugar se ejecuta generalmente el código JavaScript?
- Servidor
 - Cliente (en el propio navegador de internet)
- 2- ¿Cuáles de estas son marcas para la inserción del código JavaScript en las páginas HTML?
- <javascript_code> y </javascript_code>
 - <script> y </script>
 - <?script> y <script?>
- 3- La llamada al código Javascript debe colocarse en:
- La sección Body de la página
 - Antes de la etiqueta HTML
 - Puede colocarse en la sección Head o en Body
- 4- En JavaScript, para darle el nombre a una variable, objeto o función, debemos tener en cuenta que:
- No se pueden usar mayúsculas
 - JavaScript no distingue entre mayúsculas y minúsculas
 - JavaScript diferencia entre mayúsculas y minúsculas
- 5- ¿Cuál es la instrucción usada para devolver un valor en una función de JavaScript?
- Send
 - Return
 - Value

UNIDAD 1



Es uno de los lenguajes de programación más importantes y según datos, **lo utilizan un 80% de los desarrolladores y un 95% de todos los sitios web**. Las ventajas de [JavaScript](#) se sitúan en el lado del front-end, y varios frameworks que soporta como React y Angular JS tienen un gran potencial para mejorar la experiencia del usuario en la web. **Se trata en definitiva de un lenguaje ligero, multiplataforma, estructurado y orientado a objetos y eventos.**

- Es un lenguaje de programación **seguro y fiable**.
- De **fácil uso y muy completo**.
- **Es ligero** y permite la elaboración de múltiples aplicaciones web.
- Es compatible con la mayoría de navegadores.

Brendan Eich

Creador del lenguaje
Java Script

Brendan Eich

Brendan Eich, foto oficial de Mozilla Foundation,
21 de agosto de 2012

Información personal	
Nacimiento	1961 Pittsburgh, Pensilvania, EE.UU.
Nacionalidad	Estadounidense
Educación	
Educado en	Universidad de Illinois en Urbana-Champaign Universidad Santa Clara
Información profesional	
Ocupación	Informático teórico, programador y director de tecnología ✓
Área	Programador ✓
Conocido por	JavaScript
Cargos ocupados	Director de tecnología de Corporación Mozilla (2005-2014) Director ejecutivo de Corporación Mozilla (2014) Director ejecutivo de Brave Software (desde 2015) ✓
Empleador	Corporación Mozilla ✓
Obras notables	JavaScript ✓

A Brendan cuando trabajaba en la empresa Netscape en 1995, le encomendaron crear un nuevo lenguaje de programación en 10 días porque tenía que salir con la nueva versión del navegador Netscape.

El objetivo de dicho lenguaje era que las páginas web pudieran aprovechar el poder de procesamiento de los ordenadores para poder hacer la navegación por Internet más rápida.



1.1

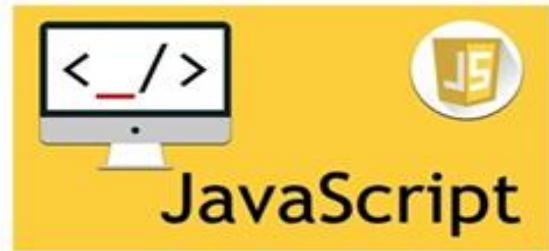


JavaScript es un robusto lenguaje de programación que se puede aplicar a un documento **HTML** y usarse para crear interactividad dinámica en los sitios web.

Fue inventado por Brendan Eich, cofundador del proyecto Mozilla

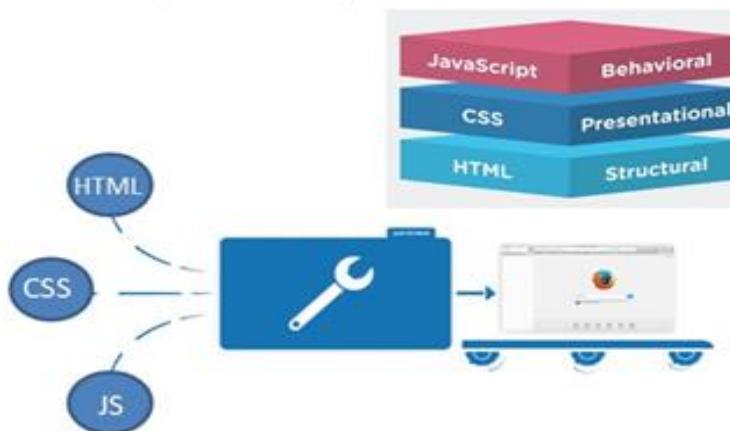
Puedes hacer casi cualquier cosa con JavaScript. Puedes empezar con pequeñas cosas como carruseles, galerías de imágenes, diseños fluctuantes, y respuestas a las pulsaciones de botones. Con más experiencia, serás capaz de crear juegos, animaciones 2D y gráficos 3D, aplicaciones integradas basadas en bases de datos ¡y mucho más!

1.2



Cuando cargas una página web en tu navegador, estás ejecutando tu código (HTML, CSS y JavaScript) dentro de un entorno de ejecución (la pestaña del navegador). Esto es como una fábrica que toma materias primas (el código) y genera un producto (la página web).

Es el lenguaje de programación más popular utilizado hoy en la web. Los sitios que visitas con frecuencia, los juegos que disfrutas, las aplicaciones en las que confías, no existirían si no fuera por JavaScript.



JavaScript MODIFICA dinámicamente HTML y CSS para actualizar la interfaz de usuario

JavaScript

- ◆ Es un lenguaje de programación de lado cliente. El término «lado cliente» significa que se ejecuta en nuestro navegador web sin necesidad de un servidor web.
- ◆ Es una puerta de acceso a otras tecnologías como AJAX, jQuery, node.js. Y un requerimiento indispensable para cualquier diseñador web que quiera hacer mejores webs
- ◆ Junto con html y css es uno de los 3 pilares del diseño web.



Creadores y navegadores



◆ **Creadores de acceso a Internet más importantes**

- PCs, portátiles, tabletas, teléfonos inteligentes

◆ **Navegador (browser) cliente Web de acceso a servidores**

- Utilizando: **URL, HTTP, HTML, CSS y JS**
 - p.e. Chrome, Firefox, Internet Explorer, Opera, Safari, ...



◆ **Tiendas de aplicaciones**

- Instalan aplicaciones en móviles y tabletas
 - Las aplicaciones usan las normas de la Web (URL, HTTP,)

Un **cliente** es un ordenador o software que accede a un servidor y recupera servicios especiales o datos de él.

Función principal del cliente:

- Estandarizar las solicitudes
- Transmitirlas al servidor
- Procesar los datos obtenidos para que puedan visualizarse en un dispositivo de salida como una pantalla.

Un **cliente no ejecuta tareas de servidor, el cliente es un elemento intermedio.**

- Los clientes que usas de forma cotidiana son los navegadores web y los clientes de correo electrónico.

Ejemplos de Clientes que se utilizan en diversos ámbitos cotidianamente:

- ❖ **Sistemas operativos:** acceso al servidor a través de la línea de comandos utilizando el protocolo Telnet
- ❖ **Navegador web:** La comunicación entre el servidor y el navegador se realiza a través del protocolo [HTTP](#). El navegador finalmente evalúa los documentos [HTML](#) recibidos o las aplicaciones [JavaScript](#).
- ❖ **Clientes de correo electrónico:** Incluso la recuperación de correos electrónicos desde un servidor se realiza con una función de cliente. Los protocolos comunes son POP3, SMTP o IMAP.
- ❖ **MMPORG:** En los juegos de rol online, el software instalado actúa como un cliente, que recupera y proporciona la información necesaria para el juego desde un servidor.
- ❖ **Clientes ligeros:** Se utilizan para aplicaciones que se ejecutan sólo en un servidor y requieren un hardware mínimo. Las soluciones en la nube serían un ejemplo de ello.
- ❖ **Clientes DNS:** Estos clientes trabajan automáticamente en segundo plano y obtienen la dirección IP apropiada a una URL desde el servidor DNS apropiado.
- ❖ **Aplicaciones basadas en la web:** como herramientas de análisis web como [Google Analytics](#), trabajando con clientes.
- ❖ **Clientes VPN:** Estos clientes establecen una conexión segura entre un servidor y un PC a través de una VPN (Virtual Private Network).

1.4

URL y Recurso



◆ URL (Uniform Resource Locator)

- Inicialmente se diseño como dirección de un recurso (página Web)
 - Se generalizó como dirección de acceso a un servicio o recurso en Internet

◆ Recurso

- Contenido digital de interés para un usuario
 - página Web, foto, película, fichero o parte de el, registro de una BD,

◆ URL tiene 3 componentes básicos

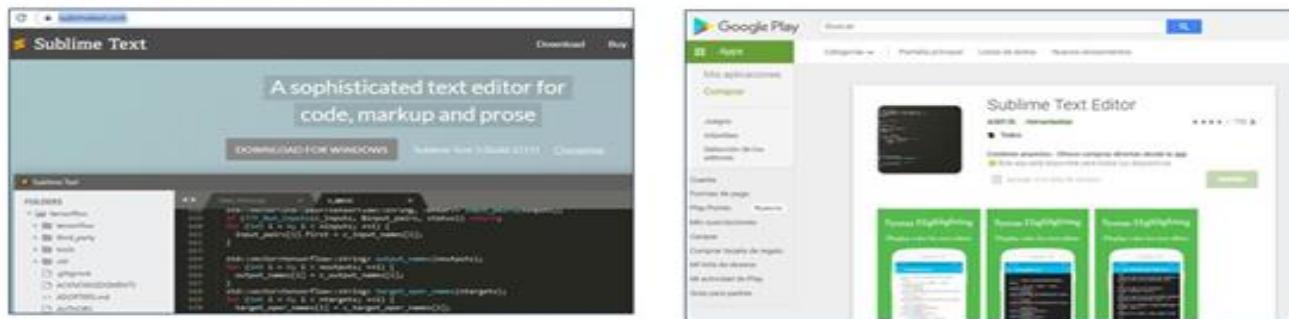
- **protocolo:** protocolo de acceso (**http**)
- **servidor:** dirección del servidor en Internet (**google.com**)
- **camino:** identificador del fichero en servidor (**/picture.com**)

UNIDAD 2

Equipos y Software a utilizar

- PC o Dispositivo móvil (Celular o tablet)
- Editor de código Sublime-Text <https://www.sublimetext.com/>
- O aplicación para el celular Sublime Text Editor

- ✓ En caso de contar con una señal inestable de internet, puedes utilizar el editor bloc de notas incluido en windows y tener instalado en tu pc o en tu celular un navegador como Chrome o Mozilla
- ✓ Las aplicaciones no requieren forzosamente la conectividad permanente con internet



UNIDAD 3

1.5

Tipos, objetos y valores

◆ Tipos de JavaScript

- **number**

- Literales de números: 32, 1000, 3.8

- **boolean**

- Los literales son los valores `true` y `false`

- **string**

- Los literales de string son caracteres delimitados entre comillas o apóstrofes
 - = "Hola, que tal", 'Hola, que tal',
 - Internacionalización con Unicode: 'Γεια σου, ισως', '嗨, 你好吗'

- **undefined**

- `undefined`: representa indefinido **UNDEFINED**



Introducción a JavaScript

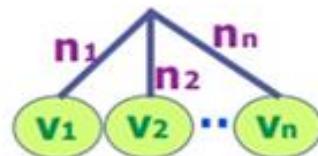
FALSE



◆ Objetos: agregaciones estructuradas de valores

- Se agrupan en **clases**: Object, Array, Date, ...

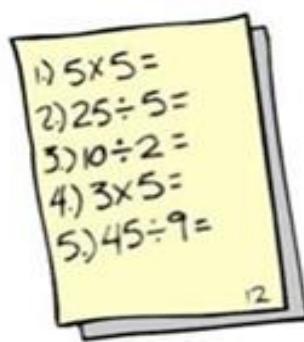
- Objeto `null`: valor especial que representa objeto nulo



1.6

Operadores y expresiones

- ◆ JavaScript incluye **operadores** de tipos y objetos
 - Los **operadores** permiten formar **expresiones**
 - Componiendo **valores** con los operadores
 - Que Javascript evalua hasta obtener un resultado
- ◆ Por ejemplo, con las operaciones aritméticas +, -, *, /
 - podemos formar expresiones numéricas
 - Expresiones con sintaxis errónea abortan la ejecución del programa



13 + 7	=>	20	// Suma de números
13 - 7	=>	6	// Resta de números
(8*2 - 4)/3	=>	4	// Expresión con paréntesis
8 / * 3	=>	Error_de_ejecución	// Ejecución se interrumpe
8 3	=>	Error_de_ejecución	// Ejecución se interrumpe

1.7

Sobrecarga de operadores

- ◆ Algunos operadores tienen varias semánticas diferentes
- ◆ Por ejemplo, el operador + tiene 3 semánticas diferentes
 - **Suma de enteros** (operador binario)
 - **Signo de un número** (operador unitario)
 - **Concatenación de strings** (operador binario)



13 + 7	=>	20	// Suma de números
+13	=>	13	// Signo de un número
"Hola " + "Pepe"	=>	"Hola Pepe"	// Concatenación de strings



1.8

Conversión de tipos en expresiones

- ◆ JavaScript realiza conversión automática de tipos
 - cuando hay ambigüedad en una expresión
 - utiliza las prioridades para resolver la ambigüedad
- ◆ La expresión "13" + 7 es ambigua
 - porque combina un **string** con un **number**
 - JavaScript asigna más prioridad al **operador +** de strings, convirtiendo 7 a string
- ◆ La expresión +"13" también necesita conversión automática de tipos
 - El **operador +** solo está definido para **number**
 - JavaScript debe convertir el **string "13"** a **number** antes de aplicar operador +

13 + 7 => 20

"13" + "7" => "137"

"13" + 7 => "137"

+ "13" + 7 => 20

1.9

Operador `typeof`

- ◆ El operador `typeof` permite conocer el tipo de un valor
 - Devuelve un string con el nombre del tipo
 - "number", "string", "boolean", "undefined", "object" y "function"

<code>typeof 7</code>	=> "number"	
<code>typeof "hola"</code>	=> "string"	
<code>typeof true</code>	=> "boolean"	
<code>typeof undefined</code>	=> "undefined"	
<code>typeof null</code>	=> "object"	
<code>typeof new Date()</code>	=> "object"	
<code>typeof new Function()</code>	=> "function"	

1.10

<code>[]</code>	Acceso a propiedad o invocar método; índice a array
<code>new</code>	Crear objeto con constructor de clase
<code>()</code>	Invocación de función/método o agrupar expresión
<code>++ --</code>	Pre o post auto-incremento; pre o post auto-decremento
<code>! ~</code>	Negación lógica (NOT); complemento de bits
<code>+ -</code>	Operador unitario, números. signo positivo; signo negativo
<code>delete</code>	Borrar propiedad de un objeto
<code>typeof void</code>	Devolver tipo; valor indefinido
<code>* / %</code>	Números. Multiplicación; división; modulo (o resto)
<code>+</code>	Concatenación de string
<code>+ -</code>	Números. Suma; resta
<code><< >> >>></code>	Desplazamientos de bit
<code>< <= > >=</code>	Menor; menor o igual; mayor; mayor o igual
<code>instanceof in</code>	¿Objeto pertenece a clase?; ¿Propiedad pertenece a objeto?
<code>== != === !==</code>	Igualdad; desigualdad; identidad; no identidad
<code>&</code>	Operación y (AND) de bits
<code>^</code>	Operación ó exclusivo (XOR) de bits
<code> </code>	Operación ó (OR) de bits
<code>&&</code>	Operación lógica y (AND)
<code> </code>	Operación lógica o (OR)
<code>?:</code>	Asignación condicional
<code>=</code>	Asignación de valor
<code>OP=</code>	Asig. con operación: <code>+= -= *= /= %= <<= >>= >>>= &= ^= =</code>
<code>,</code>	Evaluación múltiple

Operadores JavaScript

`+"3" + 7 => 10`

"+" unitario tiene mas prioridad y se evalúa antes que "+" binario

1.11

Comentarios

- ◆ Los comentarios son mensajes informativos
 - Deben ser claros, concisos y explicar todo lo importante del programa
 - Incluso el propio autor los necesita con el tiempo para recordar detalles del programa
- ◆ En JavaScript hay 2 tipos de comentarios
 - Monolinea: Delimitados por `//` y **final de línea**
 - Multilinea: Delimitados por `/*` y `*/`
 - **OJO!** Los comentarios multi-linea tienen problemas con las expresiones regulares

```
/* Ejemplo de comentario multilinea que ocupa 2 líneas
 -> al tener ambigüedades, se recomienda utilizarlos con cuidado */
```

```
var x = 1; // Comentario monolínea que acaba al final de esta linea
```

UNIDAD 4

1.12

Práctica guiada 1

Aplicación Web: HTML, CSS y JavaScript

◆ HTML

- Lenguaje de marcado

◆ CSS

- Estilo la visualización

◆ JavaScript

- Lenguaje de programación

The screenshot shows a browser window titled "Ejemplo" displaying the content of a file named "02-date_CSS.html". The page contains the following code:

```
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">

  <style type="text/css">
    body {color: blue;}
  </style>
</head>

<body>
  <h3>Fecha y hora</h3>

  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```

Below the browser window, there are three icons representing the technologies: HTML 5, CSS 3, and JavaScript 5.

Fecha y hora

Sun Sep 29 2013 17:11:38 GMT+0200 (CEST)

LISTA DE COTEJO				
Asignatura: Desarrolla aplicaciones que se ejecutan en el cliente.	1° PARCIAL		Fecha:	
Nombre del Docente: Lic. María del Socorro Juárez Soto.				
Temas	Unidad 3			
	1.5 Introducción a JavaScript (Tipos, objetos y valores) 1.6 Operadores y expresiones 1.7 Sobrecarga de operadores 1.8 Conversión de tipos en expresiones 1.9 Operador typeof 1.10 Operadores JavaScript 1.11 Comentarios			
Unidad 1				
1.1 Qué es JavaScript 1.2 Características y funcionamiento del lenguaje JavaScript 1.3 Clientes y Navegadores 1.4 URL y Recurso				
Unidad 2	Unidad 4			
- Equipo y Software a utilizar	1.12 Práctica Guiada Aplicación WEB utilizando HTML, CSS y JavaScript			
Nombre del Alumno:	Semestre:		Grupo:	
Competencias: C1 Identifica, ordena e interpreta las ideas, datos y conceptos explícitos e implícitos en un texto, considerando el contexto en el que se generó y en el que se recibe. -Expresa ideas y conceptos en composiciones coherentes y creativas, con introducción, desarrollo y conclusiones claras. -Utiliza las TIC's para investigar, resolver problemas, producir materiales y transmitir información.				
Atributos: G5.2 Ordena información de acuerdo a categorías, jerarquías y relaciones. 8. Trabaja en forma colaborativa. G8.3 Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta dentro de distintos equipos de trabajo.				
Aspectos Observables / Valor	Excelente 25 pts.	Bueno 15 pts.	Regular 10 pts.	Insuficiente 5 pts.
Define claramente los conceptos				
El contenido del trabajo es completo y conciso.				
La acentuación es correcta				
Entrega el trabajo en tiempo y forma				
Limpieza y presentación				
TOTAL				
Observaciones:				