



/INTRODUCCIÓN A JAVASCRIPT

Curso de FrontEnd

Sábado 09 de Noviembre 2024





/AGENDA



/01

/bienvenida

/02

/revisión tarea n 3

/03

/¿Qué es JS?

/04

/Historia (Jquery / AJAX / Cronología)

/05

/¿Cómo usar JS?

/06

/Tipos de datos

/07

/Variables

/08

/Funciones

/09

/tarea n 4



¿Qué es JS?

A yellow square with the letters 'JS' in black, representing the JavaScript logo.

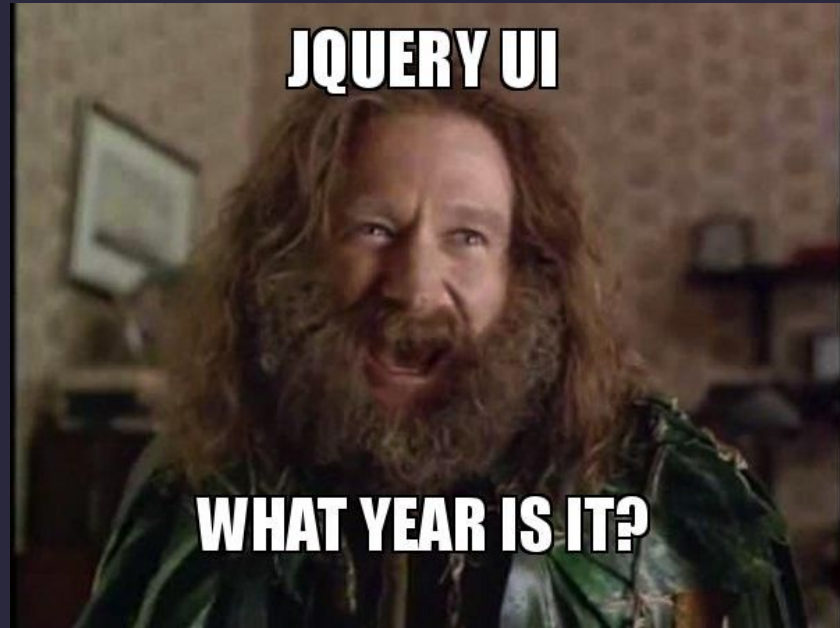
JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional).

- <https://developer.mozilla.org/es/docs/Web/JavaScript>

Historia: JQuery / Ajax / Cronología JS

jQuery es una librería desarrollada en 2006 por John Resig que permite añadir una capa de interacción AJAX entre la web y las aplicaciones que desarrollemos controlando eventos, creando animaciones y diferentes efectos para enriquecer la experiencia de usuario.

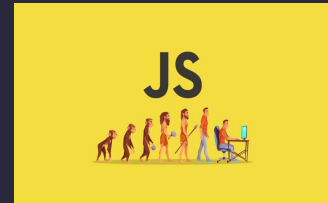
***Ajax** permite que un usuario de la aplicación web interactúe con una página web sin la interrupción que implica volver a cargar la página web



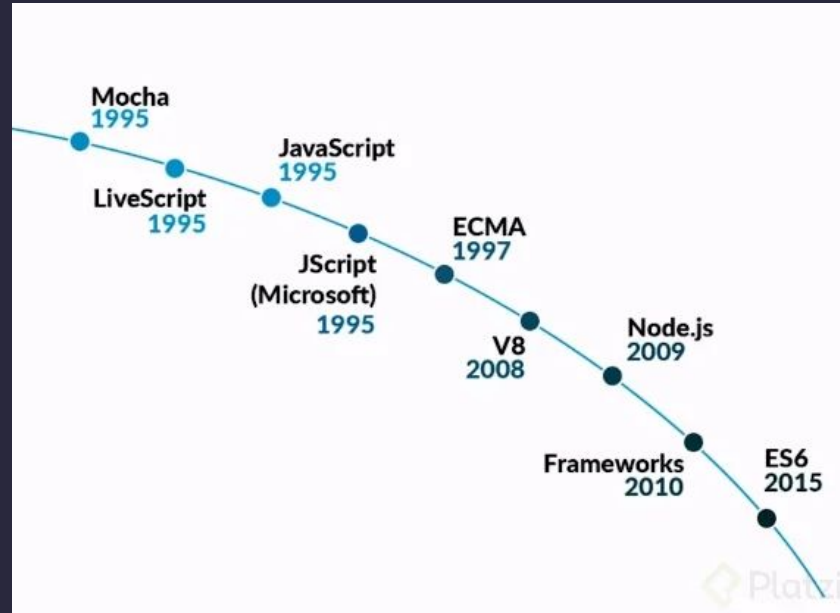
¿Sabías qué...

Javascript fue desarrollado en 10 días? 😱

La primera versión que se lanzó con el nombre de Mocha sí tardó 10 días en desarrollarse. Eso es porque *Netscape* presionó a Brendan en tener una primera versión funcional lo antes posible para poder enfrentarse al empuje de Microsoft para usar Visual Basic en la web.



Historia: JQuery / Ajax / Cronología JS



Pausa Activa



¿Cómo usar JS?



Ejemplo de script en línea

```
<html>
  <head>
    <title>Título de la página</title>
    <script>
      console.log("¡Hola!");
    </script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```

Ejemplo de script externo

```
<html>
  <head>
    <title>Título de la página</title>
    <script src="js/index.js"></script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```

<https://lenguajejs.com/javascript/introduccion/como-funciona/>

Tipos de datos en JS

Tipado dinámico

JavaScript es un lenguaje **débilmente tipado y dinámico**. Las variables en JavaScript no están asociadas directamente con ningún tipo de valor en particular, y a cualquier variable se le puede asignar (y reasignar) valores de todos los tipos:

```
let foo = 42;    // foo ahora es un número  
foo = 'bar';    // foo ahora es un string  
foo = true;     // foo ahora es un booleano
```

Tipos de datos en JS

Undefined: Una variable a la que no se le ha asignado un valor tiene el valor de undefined.

Boolean: Representa una entidad lógica y puede tener dos valores: true y false.

Number: Variable de tipo número.

Null: Tipo primitivo especial que tiene un uso adicional para su valor: si el objeto no se hereda, se muestra null.

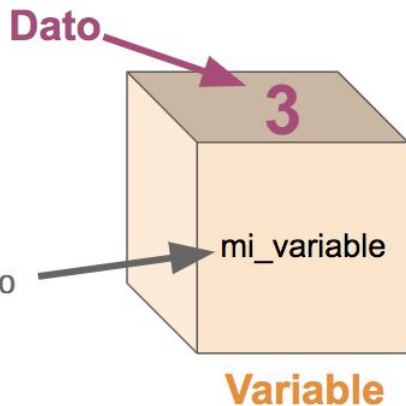
Object: Tipo estructural especial que no es de datos pero para cualquier instancia de objeto construido que también se utiliza como estructuras de datos: new Object, new Array, new Map, new Set, new WeakMap, new WeakSet, new Date y casi todo lo hecho con la palabra clave new

Function: Esta simplemente es una forma abreviada para funciones, aunque cada constructor de funciones se deriva del constructor Object

Variables en JS

Variables JS

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



Las variables son un concepto fundamental en cualquier lenguaje de programación. En JavaScript, puedes declarar variables usando las palabras clave **var**, **const** o **let**. Ejemplo:

```
let edad = 4
```

Variables en JS

¿Qué es el scope en JavaScript?

El **scope** puede definirse como **el alcance que una variable tendrá en tu código**. En otras palabras, el scope **decide a qué variables tienes acceso** en cada parte del código.




Qué es el Scope Local

Cuando puedes acceder a una variable únicamente en cierta parte del código, se dice que esa variable está declarada en el **scope local**.

Qué es el Scope Global

Se dice que una variable está en el **scope global** cuando está declarada fuera de una función o de un bloque.

Funciones en JS

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

Una función en JavaScript es similar a un procedimiento – un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como función, debe tomar alguna entrada y devolver una salida.

Para usar una función, debes definirla en algún lugar del ámbito desde el que deseas llamarla.

Funciones en JS

Declaración de función

Una definición de función (también denominada declaración de función o expresión de función) consta de la palabra clave `function`, seguida de

- El nombre de la función.
- Una lista de parámetros de la función, entre paréntesis y separados por comas.
- Las declaraciones de JavaScript que definen la función, encerradas entre llaves, `{ ... }`.

```
function square(number) {  
  return number * number;  
}
```

Lllamar una función

Definir una función no la ejecuta. Definirla simplemente nombra la función y especifica qué hacer cuando se llama a la función.

Lllamar a la función en realidad lleva a cabo las acciones especificadas con los parámetros indicados. Por ejemplo, si defines la función `square`, podrías llamarla de la siguiente manera:

```
square(5);
```

Tarea N°5

Importante Todas las tareas se deben subir en Github, no se acepta el envío de .zip mediante e-mail

Crear las funciones en JS que sean necesarias para:

Obtener el promedio de notas de un alumno considerando que la suma de notas debe ser el retorno de una función y el promedio el retorno de otra función. Las notas son: 6,8,9,2,5,10.

Crear un repositorio en GitHub y enviarlo mediante formulario:

<https://www.softwarelibrechile.cl/G24-S2-05-Tarea>

Fecha de entrega: Domingo 17 Noviembre 23:59 hrs

Rúbrica Tarea N°5



Rúbrica Tarea N°5- Evaluación para Tarea en JavaScript

Suma el porcentaje alcanzado en cada criterio para obtener una nota del 0 al 10.

Esta rúbrica refuerza la estructura modular de las funciones, la importancia de una entrega clara en GitHub, y el cumplimiento de la lógica de la tarea.

| Criterio | Descripción | Ponderación | Notas |
|-----------------------------------|--|-------------|---|
| Entrega en GitHub | La tarea fue correctamente subida en un repositorio en GitHub según las instrucciones, sin uso de archivos .zip. | 20% | 0: No se entrega en GitHub. 5-10: La tarea está en GitHub, pero con errores o incompleta. 15-19: Tarea subida correctamente con pequeños detalles. 20: Entrega en GitHub sin errores. |
| Función para Suma de Notas | Implementación de la función para obtener la suma de las notas. | 20% | 0: La función no está implementada. 5-10: La función está incompleta o con errores de ejecución. 15-19: La función realiza la suma correctamente, pero con detalles menores. 20: Función correcta y sin errores. |
| Función para Cálculo del Promedio | Implementación de la función para obtener el promedio de las notas usando el valor de la función de suma. | 25% | 0: No se implementa la función. 5-12: La función está incompleta o no calcula el promedio adecuadamente. 13-19: La función calcula el promedio, pero hay detalles a mejorar. 20-24: Función funcional con detalles menores. 25: Función correcta y bien estructurada. |
| Modularización del Código | Las funciones para suma y promedio están correctamente separadas y organizadas. | 15% | 0: Funciones no separadas o falta alguna de ellas. 5-10: Funciones creadas, pero la organización es confusa. 11-14: Modularización adecuada con detalles menores. 15: Funciones bien organizadas y separadas. |
| Comentarios y Legibilidad | Código bien comentado para facilitar la comprensión, con nombres de variables claros y legibilidad en el código. | 10% | 0: Sin comentarios y nombres confusos. 3-6: Pocos comentarios o falta de claridad en variables. 7-9: Bien comentado, con algunos detalles menores. 10: Excelente legibilidad y claridad. |
| Pruebas de Ejecución | Comprobación de que las funciones retornan los valores esperados (suma correcta y promedio correcto). | 10% | 0: Las funciones no producen los valores esperados. 3-6: Las funciones retornan valores correctos pero falta verificar casos límites. 7-9: Retornan correctamente pero con algún ajuste posible. 10: Funciones devuelven valores precisos en todas las pruebas. |

Programas

Visual Studio Code

www.softwarelibrechile.cl/vscode

Trello

www.trello.com

Codesandbox

codesandbox.io

Programa

- 01 - Fundamentos
- 02 - Estilos y diagramación
- 03 - Estilos y diagramación
- 04 - Introducción a JS
- 05 - Primeros pasos a JS
- 06 - Funciones, API's, Manejo de errores
- 07 - Programación JS
- 08 - ReactJS
- 09 a 12 - Práctico grupal

Curso FrontEnd: Contenidos

| CLASE | CAPÍTULO | CONTENIDOS |
|----------|-------------|--|
| SÁBADO 1 | FUNDAMENTOS | <ol style="list-style-type: none">1. BIENVENIDA AGI2. ORGANIZACIÓN DE TAREAS3. TRELLO4. TOMA DE REQUERIMIENTOS BASE5. METODOLOGÍAS DE TRABAJO6. HTML, CSS, JS7. BOOTSTRAP INSPECTOR ELEMENTOS.8. TAREA N° 1 |

Curso FrontEnd: Contenidos

| | | |
|-----------------|---|--|
| SÁBADO 2 | ESTILOS Y DIAGRAMACIÓN CSS PARTE 1 | <ol style="list-style-type: none">1. REVISIÓN TAREA N°12. BEM, SASS, FLEXBOX3. PREFIJOS, NAVEGADOR, MEDIAQUERYS4. Posicionamiento, layout5. Cajas (tipos, posición)6. Responsividad7. Tarea N° 2 |
| SÁBADO 3 | ESTILOS Y DIAGRAMACIÓN CSS PARTE 2 | <ol style="list-style-type: none">1. Revisión Tarea N° 22. GRID3. Introducción a GIT4. Visionamiento de código (GITHUB)5. Despliegue (GITHUB Pages)6. Tarea N° 3 |

Curso FrontEnd: Contenidos

| | | |
|-----------------|------------------------------|---|
| SÁBADO 4 | INTRODUCCION JS | <ol style="list-style-type: none">1. Revisión Tarea N° 32. ¿Qué es JS?3. ¿Cómo usar JS?)4. JQUERY, AJAX5. Tipos de datos6. Funciones7. Tarea N° 4 |
| SÁBADO 5 | PRIMEROS PASOS CON JS | <ol style="list-style-type: none">1. Revisión Tarea N° 42. ARRAYS, CALLBACK3. Ciclos4. Tarea N° 5 |

Curso FrontEnd: Contenidos

| | | |
|-----------------|--|---|
| SÁBADO 6 | FUNCIONES, API'S, MANEJO DE ERRORES | <ol style="list-style-type: none">1. Revisión Tarea N° 52. Tarea N° 6 |
| SÁBADO 7 | PROGRAMACIÓN JS | <ol style="list-style-type: none">1. Revisión Tarea N° 62. herencias3. Tarea N° 7 |
| SÁBADO 8 | INTRODUCCIÓN A REACTJS | <ol style="list-style-type: none">1. Revisión Tarea N° 72. Definición proyecto final3. Tarea N° 8 |
| SÁBADO 9 | PRÁCTICO (GRUPAL) | <ol style="list-style-type: none">1. Revisión Tarea N° 82. Proyecto final}3. Tarea N° 9 |

Curso FrontEnd: Contenidos

| | | |
|------------------|--|---|
| SÁBADO 10 | PRÁCTICO (GRUPAL) | <ol style="list-style-type: none">1. Revisión Tarea N° 92. Presentación y entrega |
| SÁBADO 11 | PROYECTO FINAL (GRUPAL) | <ol style="list-style-type: none">1. Presentación y entrega |
| SÁBADO 12 | ENTREGA CERTIFICADOS APROBACIÓN | <ol style="list-style-type: none">1. Presentación y entrega2. Entrega Certificados3. Cierre |

Canales de comunicación

Slack (principal)

www.softwarelibrechile.cl/slack

WhatsApp

www.softwarelibrechile.cl/whatsapp

/Porcentajes a cumplir en el curso

A continuación les dejamos información acorde al % de asistencia, entregas de tarea y proyecto final, todo esto de **carácter obligatorio** para obtener el certificado de aprobación.

| /CLASES | /TAREAS | /PROYECTO FINAL |
|--|--|--|
| ASISTENCIA 80% | ENTREGAS 80% | ENTREGA 100% |
| 11 clases | 8 tareas | 1 proyecto |
| Debe existir asistencia mínima de 9 clases | Deben entregarse mínimo 7 tareas o más | El proyecto final se hará de forma grupal, 2 personas, ambas obteniendo la misma nota. |

/APROBACIÓN Y ENTREGA CERTIFICADO



/TAREAS

El promedio de las notas de la tareas equivale a un

/50%



/PROYECTO FINAL

La nota en el proyecto final equivale a un

/50%

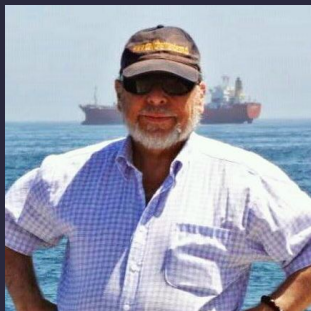


/NOTA FINAL

La nota debe ser igual o mayor a 6

/100%

Curso FrontEnd: Relatores



David Hernández

Director Asociación de
Informáticos UTE-USACH
A.G. | Paleo Informático
david.hernandezm@usach.cl
+56998246832

<https://www.linkedin.com/in/david-hern%C3%A1ndez-23205b3a>



Gonzalo Flemming

Tech Lead en Falabella
Financiero

<https://www.linkedin.com/in/gfleming-garrido/>



Sebastian Becerra

Gerente de Operaciones en
PropulsoW, Ingeniero
Informático, voluntario
Mozilla Chile ayudante en
taller Joomla dictado en la
Usach el 2014- 2015

<https://www.linkedin.com/in/sebaebc/>



Scarlet Melgarejo

Encargada de Canales
Digitales B2B - B2C en
Productos Torre , Ayudante
en Desafío Latam, voluntario
Mozilla Chile ayudante en
taller Joomla dictado en la
Usach el 2011 - 2015 2015.

<https://www.linkedin.com/in/scarlett-melgarejo-venegas-38805626/>



Curso FrontEnd: Relatores



Carolina Pirela

Ingeniera en Sistemas -
Front End Developer React
| Javascript | Sass |
Bootstrap
Dev UI Mobile @ Seiza

<https://www.linkedin.com/in/caropi30/>



Cristian Pavéz

Head of AI and Data Innovation
en Soluciones - an AVOS Tech
Company
Ingeniero en Informática

<https://www.linkedin.com/in/cristian-pavez>

<gracias!>