

A yellow square containing the letters "JS" in a bold, black, sans-serif font.

SEMINARIO DE  
LENGUAJES

EXPLICACIÓN PRÁCTICA 1

CONFIGURACIÓN BÁSICA Y  
PRIMEROS PASOS EN JAVASCRIPT

# AGENDA

1. Introducción A Git
2. Introducción A Javascript

# INTRODUCCIÓN A GIT

En esta materia, vamos a utilizar Git como nuestro sistema de control de versiones para manejar el código fuente de nuestros proyectos.

# ¿QUÉ ES GIT?

Git es un sistema de control de versiones distribuido, diseñado para manejar todo tipo de proyectos con velocidad y eficiencia.

## CONTROL DE VERSIONES

Git no solo almacena los cambios en los archivos de nuestro proyecto, sino que también permite navegar por todo el historial de cambios. Esto nos da la capacidad de comparar versiones, entender quién hizo qué y cuándo, y revertir a cualquier estado anterior del proyecto si es necesario.

# OPERACIONES BÁSICAS DE GIT<sup>1</sup>

- **Clone:** Copiar un repositorio existente en una nueva ubicación local.
- **Add:** Agregar cambios en archivos a la área de preparación.
- **Commit:** Registrar los cambios preparados en el historial del repositorio.
- **Push:** Enviar cambios locales a un repositorio remoto.
- **Pull:** Actualizar el repositorio local con cambios desde un repositorio remoto.

## OPERACIONES BÁSICAS DE GIT <sup>2</sup>

No es necesario memorizar estas operaciones por ahora; lo importante es comprender los conceptos subyacentes. Profundizaremos en estos comandos en la próxima clase.

## **IMPORTANCIA DE GIT EN DESARROLLO**

Git nos permite colaborar eficientemente, mantener un historial completo de cambios, y revertir a versiones anteriores si es necesario. Será una herramienta esencial en nuestra materia.

## PRÁCTICA CON GIT

A lo largo de la materia, realizaremos ejercicios prácticos para familiarizarnos con Git, desde configurar nuestro entorno hasta colaborar en proyectos grupales.

# INSTALACIÓN DE GIT

- Windows: Instalar git-bash desde su [sitio oficial](#).
- Linux: Utilizar el comando `sudo apt-get install git` en la terminal.

# CONFIGURACIÓN DE GIT

Antes de comenzar a utilizar Git, es importante configurar nuestro nombre y dirección de correo electrónico. Esto nos permitirá identificar quién realizó cada cambio en el proyecto.

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu@email.com"
```

# VERIFICACIÓN

Para asegurarnos de que todo está correctamente instalado, podemos verificar las versiones de Git y de configuración

- `git --version`
- `git config --list`

# CLONAR EL REPOSITORIO DEL GRUPO

Para clonarse el repositorio de cada grupo de trabajo hay que hacer el siguiente comando.

```
git clone git@gitlab.catedras.linti.unlp.edu.ar:js/grupos-2024/grupo
```

A partir de ése momento, el repositorio se clonará en la carpeta que se encuentre en la terminal. Vamos a poder realizar modificaciones y subirlas al repositorio.

# INTRODUCCIÓN A JAVASCRIPT

JavaScript es un lenguaje de programación que se ejecuta (originalmente) en el navegador del cliente y permite crear páginas web interactivas.

A diferencia de HTML y CSS, que proporcionan la estructura y el estilo de una página, JavaScript proporciona funcionalidad interactiva.

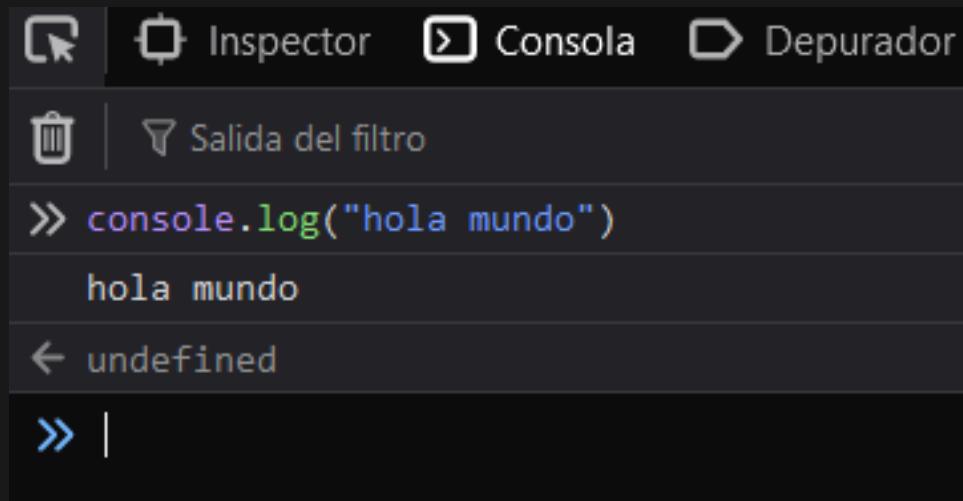
Es un lenguaje de programación interpretado, lo que significa que el código se ejecuta directamente sin necesidad de compilación previa.

# EJEMPLO DE UN SCRIPT EN JAVASCRIPT

```
// Esto es un comentario
let contador = 1
while (contador <= 5) {
    console.log(contador)
    contador++
}
```

# EJECUTAR JAVASCRIPT DESDE LA TERMINAL DEL NAVEGADOR

La consola del navegador es una herramienta poderosa que te permite interactuar con la página web mediante código JavaScript.



A screenshot of a browser's developer tools interface, specifically the 'Consola' (Console) tab. The tab bar at the top includes 'Inspector', 'Consola' (which is selected), and 'Depurador'. Below the tabs is a toolbar with a trash can icon and a filter icon. The main area shows a command line input followed by its output. The command entered is 'console.log("hola mundo")'. The output is 'hola mundo', indicating the code was executed successfully. At the bottom of the console, there is a prompt '» |' where the user can enter more commands.

```
» console.log("hola mundo")
hola mundo
» |
```

# EJECUTAR JAVASCRIPT EN UN ARCHIVO HTML<sup>1</sup>

JavaScript se puede incorporar en documentos HTML de dos maneras:

- Internamente, mediante el uso de la etiqueta `<script>` dentro del documento HTML
- Externamente, vinculando un archivo JavaScript externo

# EJECUTAR JAVASCRIPT EN UN ARCHIVO HTML<sup>2</sup>

## MEDIANTE EL USO DE LA ETIQUETA <script>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi Primera Página con JavaScript</title>
  </head>
  <body>
    <h1>¡Hola Mundo!</h1>
    <script>
      console.log('Hola Mundo desde el script interno')
    </script>
  </body>
</html>
```

# EJECUTAR JAVASCRIPT EN UN ARCHIVO HTML<sup>3</sup>

## VINCULANDO UN ARCHIVO JAVASCRIPT EXTERNO

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi Primera Página con JavaScript</title>
  </head>
  <body>
    <h1>¡Hola Mundo!</h1>
    <script src="script.js"></script>
  </body>
</html>
```

# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador

```
alert('¡Bienvenido a JavaScript!')
```



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador

```
alert('¡Bienvenido a JavaScript!')
```

Muestra una ventana de alerta con un mensaje.



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador

```
alert('¡Bienvenido a JavaScript!')
```

Muestra una ventana de alerta con un mensaje.

## Variables y Operaciones Básicas



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador

```
alert('¡Bienvenido a JavaScript!')
```

Muestra una ventana de alerta con un mensaje.

## Variables y Operaciones Básicas

```
let numero = 10  
console.log(numero * 2)
```



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>1</sup>

## Manipulación del DOM

```
document.body.style.backgroundColor = "lightblue"
```

Este código cambia el color de fondo de la página a celeste.

## Alertas en el Navegador

```
alert('¡Bienvenido a JavaScript!')
```

Muestra una ventana de alerta con un mensaje.

## Variables y Operaciones Básicas

```
let numero = 10  
console.log(numero * 2)
```

Imprime 20 en la consola.



# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>2</sup>

# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>2</sup>

Interacción con el Usuario

# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>2</sup>

## Interacción con el Usuario

```
let nombre = prompt("¿Cuál es tu nombre?")
console.log('Hola, ' + nombre)
```

# EJEMPLOS PRÁCTICOS PARA EJECUTAR<sup>2</sup>

## Interacción con el Usuario

```
let nombre = prompt("¿Cuál es tu nombre?")
console.log('Hola, ' + nombre)
```

Solicita al usuario su nombre y luego lo saluda.

**FIN DE LA CLASE**