



Hola Hello, Welcome!

Registra tu asistencia

<https://forms.gle/6hfdT5U5v5fF34RQ8>



Clase 8



5 Minutos de Código

Repaso de fundamentos de programación - Challenge 3

Conversión de grados fahrenheit a Celcius

Escribe una función que reciba un número y convierta esa cantidad a grados F

Requisitos



1. Entrada: temperatura en Fahrenheit.
2. Proceso: aplicar fórmula $(F - 32) \times 5 / 9$.
3. Salida: temperatura equivalente en Celsius



2 Parte Github

Practica 2




Github - Flujo de Trabajo & Comandos Básicos

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

<https://github.com/joshnh/Git-Commands>



Practica 2 - Git Hub



Crea tu entorno de trabajo local

Genera una carpeta, aquí ira tu primer proyecto

Abre la terminal en esta ruta e inicializa git con:

1. git init



Primeros pasos

–Git Config (Configurar git en tu computadora).

2. git config --global user.name "Tu Nombre"

3. git config --global user.email "tuemail@ejemplo.com"




Clonar repositorio

<https://github.com/Codevs-Coderfy/logica-challenges>



Crear una Nueva Rama (Branch)

`git branch mi-nueva-rama`
(Asegurate de estar en la ruta que coincida con del repositorio clonado)



**Antes de hacer cualquier
cambio: Visualiza en qué rama
te encuentras.**

git branch



Moverse a la Nueva Rama (New Branch)

`git checkout mi-nueva-rama`

Para crear y al mismo tiempo moverse a la nueva rama, se usa git checkout -b <nombre-de-la-rama>





Ahora, Crea tu archivo js y subelo

Crea tu archivo: challenge3.js
Agrega la logica y envialo al repo



Verifica los cambios realizados

`git status`



Agregar y Confirmar los Cambios al entorno local

```
git add .  
git commit -m "Add Challenge 3"
```




Actualizar la rama principal con los últimos cambios del repositorio remoto

```
git fetch origin  
git checkout main  
git pull origin main
```



Fusionar la rama principal actualizada en tu rama de trabajo

git checkout nombre-de-la-rama
git merge main



Subir los Cambios a GitHub

```
git push origin mi-nueva-rama
```

Comandos utilizados



git clone: Crea una copia local del repositorio remoto en tu máquina.

git checkout -b: Crea y cambia a una nueva rama para trabajar en una característica o corrección específica.

git add: Añade los cambios realizados al área de preparación para su posterior confirmación.

git commit: Confirma los cambios añadidos al área de preparación con un mensaje descriptivo.

git fetch: Descarga los últimos cambios del repositorio remoto sin fusionarlos.

git pull: Descarga y fusiona los últimos cambios del repositorio remoto en tu rama actual.

git merge: Fusiona los cambios de una rama en otra.

git push: Sube tus cambios confirmados al repositorio remoto.




Tips

Utiliza mensajes de commit claros y descriptivos para facilitar la comprensión de los cambios realizados.

Mantén tu rama actualizada con la rama principal para evitar conflictos al fusionar.

Revisa los cambios antes de crear un Pull Request para asegurarte de que todo esté correcto.



Desarrollo web

Pasos iniciales

HTML



Qué es HTML

HyperText Markup Language.

Es el lenguaje que estructura el contenido de una página web.

No es un lenguaje de programación: es un lenguaje de marcado.



Más contexto de nuestro Doctype Html...



`<!DOCTYPE html>` → Indica al navegador que este archivo es **HTML5**.

`<html lang="es">` → Es el contenedor principal de todo el documento. El atributo `lang="es"` indica que el idioma es español.

`<head>` → Contiene información **meta** del sitio:

- `<meta charset="UTF-8">` → Codificación de caracteres, para que se muestren acentos y símbolos correctamente.
- `<title>` → Título de la página que aparece en la pestaña del navegador.
- Aquí también se agregan estilos (`<link>`), scripts (`<script>`), y otros metadatos.


`<body>` → Contiene **todo lo visible en la página**: textos, imágenes, enlaces, botones, etc.



**El Body construye nuestro
sitio...**

Etiquetas principales dentro del `<body>`

- `<header>` → Encabezado de la página o sección (normalmente con título, logo o menú).
- `<nav>` → Menú de navegación o enlaces a otras páginas.
- `<main>` → Contenido principal del sitio. Solo debería haber **un `<main>` por página**.
- `<section>` → Sección del contenido, como un bloque temático.
- `<article>` → Artículo independiente, por ejemplo un post de blog o noticia.
- `<aside>` → Contenido lateral o complementario, como un sidebar o anuncios.
- `<footer>` → Pie de página, con información de contacto, derechos de autor, links adicionales.



!+Tab Para crear nuestro primer doc html en visual studio code

...


En Body ira todo lo necesario para construir el esqueleto de nuestra página web

```
<h1>Bienvenidos a mi página web</h1>
```

```
<p>Esta es mi primera página HTML.</p>
```

```
<a href="https://www.google.com">Ir a Google</a>
```

Qué puede incluirse en nuestro html?



- Encabezados (títulos) `<h1>` a `<h6>`
- Párrafos `<p>`
- Listas ``, ``, ``
- Enlaces `<a>`
- Imágenes ``
- Tablas `<table>`

Otros elementos comunes



- **Divisores** `<div>` → Contenedor genérico para agrupar contenido.
- **Secciones semánticas** `<header>`, `<footer>`, `<main>`, `<section>`, `<article>`, `<aside>`.
- **Botones** `<button>` → Para acciones del usuario.
- **Formularios** `<form>` con `<input>`, `<textarea>`, `<select>` → Para recopilar información.
- **Tablas** `<table>` con `<tr>`, `<th>`, `<td>` → Para organizar datos tabulares.



A grandes rasgos

Permite que los navegadores muestren contenido correctamente.

Se combina con CSS y JavaScript:

- CSS → le da **estilo**
- JavaScript → le da **interactividad**