

# Paper encuentro 4

<b>Repaso clase anterior</b>	1
<b>Continuar con ejercicio</b>	1
Estructurar assets y enlazar al documento	1
Formatos de imágenes	1
<b>Git</b>	2
Crear repositorio en github.com	2
Iniciar repositorio local	4
Subiendo mis cambios al repositorio remoto	4
Obteniendo los últimos cambios	4
<b>Posicionamientos, alto y ancho</b>	5
<b>Selectores</b>	5
IDs	5
Clases	5
Herencia	6
<b>Utilidades</b>	6

## Repaso clase anterior

1. HTML
2. Elementos HTML
3. Elementos anidados
4. Clasificación

## Continuar con ejercicio

Armar estructura básica y contenidos

<https://xd.adobe.com/view/51795fcc-7085-48a6-bb10-26070be36c9d-042a/>

1. Preguntar si hay algún voluntario que quiera compartírnos pantalla
2. Revisar e ir ajustando al momento entre todos
3. Primero repasar elementos del header (title, charset y viewport, luego mostrar meta description).
4. Pasar a controlar y ajustar elementos del body, repaso por las etiquetas que vamos encontrando y sus propiedades principales.

## Estructurar assets y enlazar al documento

Todo elemento externo al documento que vamos a insertar, ya sean imágenes, css, scripts, etc. Se conoce como Asset (bienes en español).

- Crear estructura de carpetas
- Bajar imágenes del diseño enviado y estructurarlas
- Mostrar cómo incorporar las imágenes al documento

## Formatos de imágenes

<https://www.ionos.es/digitalguide/paginas-web/disenio-web/cuales-son-los-formatos-de-imagen-mas-importantes/>

	JPG	PNG	GIF	SVG
Esquemas de color	RGB, escala de grises, CMYK	RGB, escala de grises, colores indexados	Colores indexados	RGB, nombres de color de SVG
Número de colores	Hasta 16,7 mill.	Hasta 18 trillones	Hasta 256	Hasta 16,7 mill.

Canales de color	Tres	Tres (más un canal alfa)	Uno	Tres (más un canal alfa)
Profundidad de bits	8 bits por canal	1–16 bits por canal	1–8 bits	8 bits por canal
Compresión	Alta, con pérdidas	Alta, sin pérdidas	Escasa	Ninguna
Tamaño de los archivos	Muy pequeño	Pequeño	Grande	Individual
Animaciones	No	No	Sí	Sí
Adecuado para	Fotos	Imágenes y gráficos de pequeña envergadura (p. ej., logotipos), fotos sin pérdidas	Animaciones	Todo tipo de gráficos (logotipos, iconos, diagramas, etc.)

## Git

<https://medium.com/@sthefany/primeros-pasos-con-github-7d5e0769158c>

Git es un software de control de versiones. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

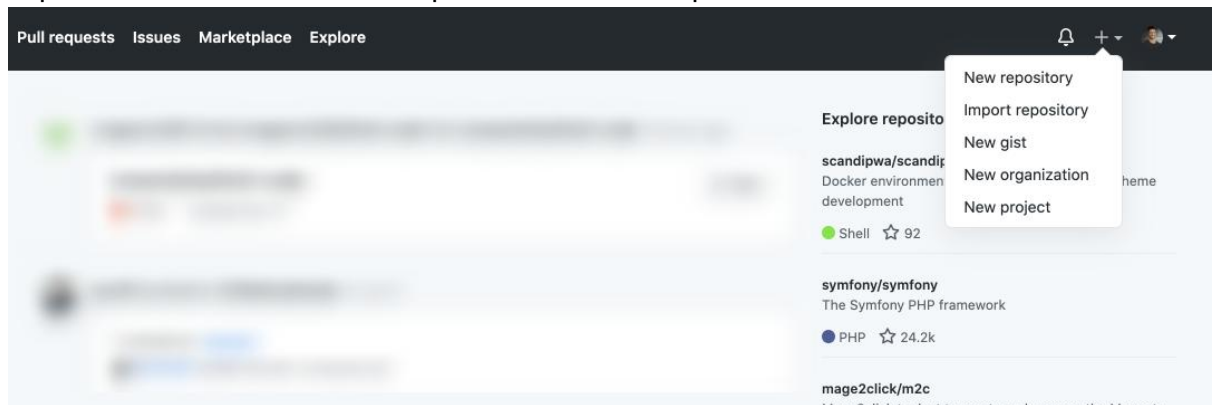
Github es una plataforma de desarrollo colaborativo donde se alojan los repositorios.

1. Crear repositorio público en github.com
2. Iniciar repo y configurar el origen al repo recién creado en github.com
3. revisar el estado de los archivos, stage y commit
4. Pull para traer cambios, push para enviarlos a nuestro origen

## Crear repositorio en github.com

1. Ingresar a [github.com](https://github.com)
2. Iniciar sesión con su cuenta (en caso de no tener cuenta deben crearse una)

3. Una vez dentro de su cuenta de github desde el botón + (más) ubicado en la parte superior derecha encontrará la opción de crear un repositorio




4. Luego en la vista de crear nuevo repositorio es necesario ingresar un nombre (sin espacios ni caracteres especiales) y dependiendo del alcance puede ser público o privado. El resto de opciones dejar en blanco.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

 joseamietta ▾

Repository name \*

/ prueba ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-octo-meme?](#)

Description (optional)



 **Public**

Anyone on the internet can see this repository. You choose who can commit.



 **Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

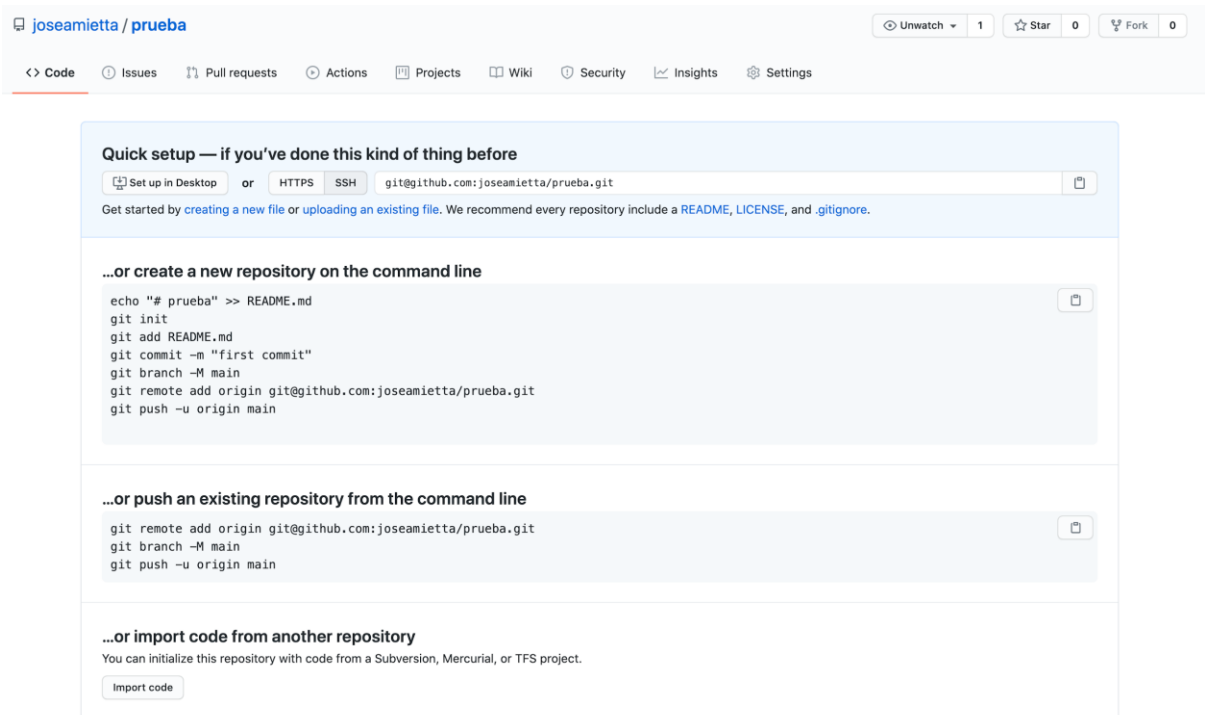
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

5. Luego de creado los dirige a la pantalla del repositorio con algunas indicaciones para subir contenido



## Iniciar repositorio local

Tenemos dos manera para iniciar un repositorio en local:

- `git init`
  - inicia un repositorio vacío, es la opción correcta para enlazar un repositorio de cero.
- `git clone <url-repositorio> <directorio-de-instalación>`
  - Recupera el contenido de un repositorio ya existente en la carpeta local indicada.

## Subiendo mis cambios al repositorio remoto

Antes de comenzar es importante saber que existen distintos estados en git, archivos modificados, archivos preparados (stage) y archivos listos para subir (commit).

- `git add <archivo>`: prepara los archivos
- `git commit`: agregar una entrada en el historial git
- `git push`: sube los cambios realizado en el commit

## Obteniendo los últimos cambios

- `git pull`: obtiene los cambios y actualiza nuestro local
- `git fetch`: obtiene los cambios pero nos los aplica

git log --oneline --graph --decorate --name-only

## Posicionamientos, alto y ancho

El posicionamiento, alto y ancho hacen referencia al espacio y lugar que un elemento HTML ocupa en el documento. Como vimos anteriormente, dependiendo de su clasificación esto varía.

Los elementos de bloque sabemos que ocupan el 100% del ancho del documento, pero el alto es variable de acuerdo al contenido, y los elementos en línea solo ocupan la porción del documento que necesitan dependiendo de su contenido, y los elementos vacíos que simplemente insertan un objeto al documento.

En un documento HTML tomar parte de la estructura de trabajo y mostrar el espacio que ocupan. Luego mostrar que esos estilos se pueden modificar mediante CSS en línea.

position: relative | absolute | fixed  
display: block | none | inline | table | flex | grid  
top:  
bottom:  
left:  
right:  
float: left | right | none  
width: px | rm | %  
height: px | rm | %  
margin: auto  
padding: auto

## Selectores

### IDs

- Son únicos
- Se identifica anteponiendo #
- Es el selector de más peso

<p id="texto" Elemento HTML</p>

### Clases

- Puede estar presente en más de un elemento
- Se identifica con el . (punto)
- Le sigue al ID dentro del peso

<p class="texto" Elemento HTML</p>

# Herencia

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos.

No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran. Normalmente, el sentido común dicta qué propiedades se heredan y cuáles no. En este enlace encontrarán información detallada al respecto

<https://www.w3.org/TR/css-2010/#properties>.

# Utilidades

- <https://gitexplorer.com/>