

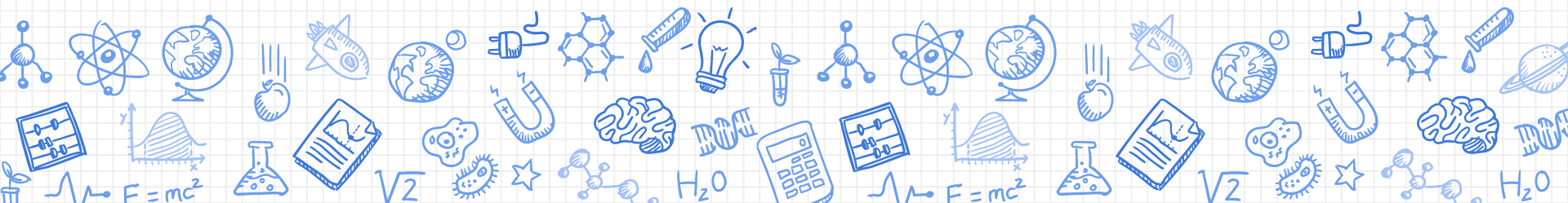
Clase 06

En la clase de hoy:

- Git/GitHub.
- Flexbox



Git / GitHub



- Git es un **sistema de control de versiones**.
- Registra los cambios que hacemos en los diferentes archivos de un proyecto a lo largo del tiempo.
- Si bien existen varios sistemas de control, Git es el más utilizado y aceptado por la comunidad.

GitHub

- GitHub es un servicio de **alojamiento de repositorios**.
- Es un sitio web donde podemos compartir el código de nuestro proyecto con todos los cambios que guardó Git.
- Otra plataforma muy conocida es BitBucket.



- Lo primero que vamos a hacer es instalar Git. Para eso, vamos a la página oficial: <https://git-scm.com/> y lo instalamos como cualquier otro programa.
- Mientras se descarga Git, vamos a entrar a GitHub: <https://github.com/> y crear una cuenta.

- Una vez instalado, vamos a buscar:
 - En Windows: Git Bash.
 - En Mac/Linux: Terminal.
- Se nos va a abrir una pantalla negra donde vamos a escribir lo siguiente: **git --version**
- Si luego de dar enter nos dice qué versión de Git tenemos, sin importar cuál sea, somos felices!

Configuración

- Cuando esté instalado, vamos a hacer una configuración básica.
- Esto es muy importante ya que cada vez que confirmamos un cambio, se va a “firmar” con estos datos.
- Para configurar nuestros datos vamos a ejecutar los siguientes comandos:




```
git config --global user.name "nombreUsuario"
```

```
git config --global user.email "emailUsuario"
```

- La idea es que estos datos se completen igual a su cuenta de GitHub.

- Para manejarnos en la consola, vamos a necesitar conocer tres comandos:
 - **cd**
 - **cd ..**
 - **ls** (Mac/Linux) / **dir** (Windows).

- El comando **cd** nos permite movernos a un directorio específico. Podemos pasarle una ruta absoluta de nuestra computadora o movernos con rutas relativas.
- Por ejemplo:
 - **cd Coderhouse/Clase08**

-

- Los comandos **ls** o **dir**, nos van a servir para ver el listado de cosas que hay en el lugar donde estamos parados.
- Entonces rápidamente vamos a saber dónde estamos, y tener una referencia de como usar el comando **cd** para llegar a donde queramos.

-

- Para crear un repositorio, dentro de la consola debemos pararnos en la carpeta de nuestro proyecto.
- Una vez dentro, ejecutamos el comando **git init**.

- Una vez que ejecutamos el **git init**, que es único para toda la historia de la humanidad, por proyecto, trabajamos normalmente en nuestro proyecto.
- Cuando hacemos algún cambio significativo que queremos guardar, vamos a la consola y ejecutamos el **git add**.
- El punto va a hacer que se agreguen todos los archivos por más de que no se hayan modificado.

Git – git commit

- Luego de agregar los archivos debemos hacer el commit.
- El commit es el momento en el que decimos... “Che, lo que hice hasta acá lo voy a guardar, definitivamente”.
- Para eso, escribimos **git commit -m** “**mensaje**”, el mensaje es el que cada uno quiera, pero la idea es explicar qué se hizo en ese commit.

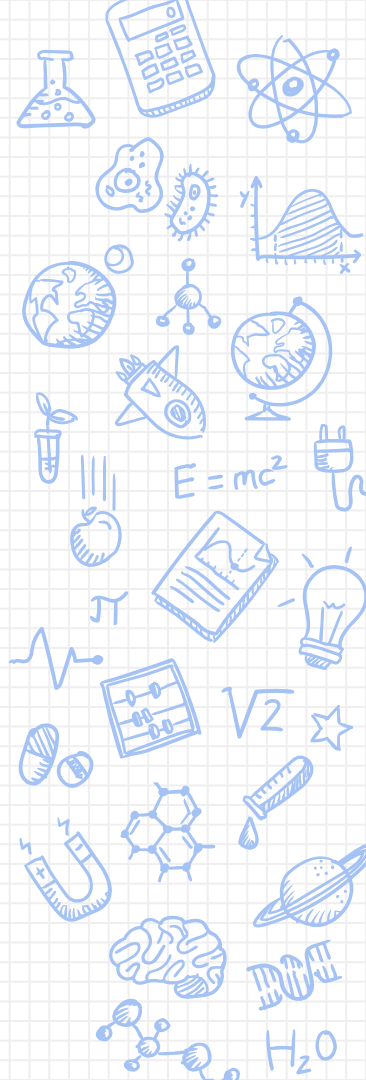


- **git status** es un comando que vamos a usar todo el tiempo. Nos va a servir para saber qué está pasando con nuestro repositorio.
- Simplemente nos va a decir qué archivos se modificaron, si falta hacer el add de algo, etc.

- Nuestro proyecto local está listo, es momento de conectarlo con GitHub.
- Para esto, debemos entrar a nuestro perfil y crear un nuevo repositorio, le ponemos un nombre y alguna descripción.
- Y luego nos va a dar un pequeño “tutorial”.
- Para conectar el repositorio local con el remoto, vamos a usar un comando llamado **git remote add origin**.

GitHub

- Una vez generado el repositorio en GitHub, vamos a notar que nos muestra un link. Ese link debemos copiarlo y agregarlo al final del comando:
 - `git remote add origin link`
- Listo. Eso es todo para vincular los repositorios, y sólo se hace una vez.



GitHub

- Si bien están vinculados, falta un último paso que es subir los archivos.
- Para esto, vamos a usar un último comando llamado **git push origin master**.
- Este comando lo ejecutamos en la consola y nos va a indicar cuando se complete la subida.
- Una vez completa, vamos a GitHub, recargamos la página y vemos nuestros archivos.

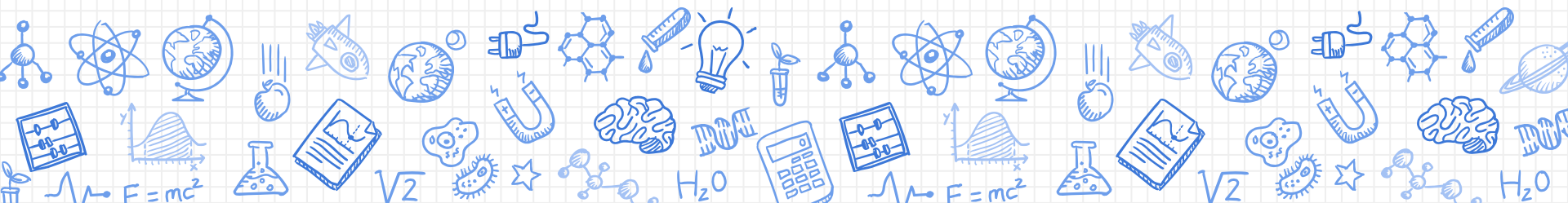


Simplificando

- Los pasos a seguir en la consola siempre que iniciemos un proyecto son:
 - `git init` → una única vez.
 - `git add .` → cada vez que haya cambios.
 - `git commit -m "..."` → luego de cada add.
 - `git remote add origin "link"` → una única vez.
 - `git push -u origin master` → una sola vez, después solo `git push`.

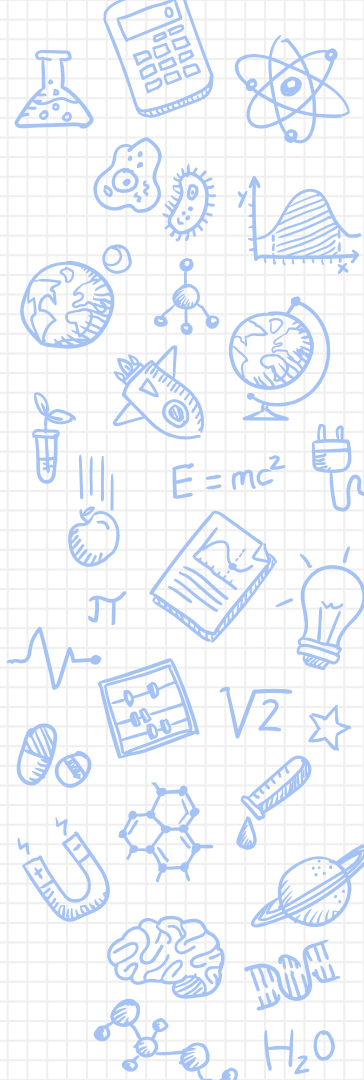
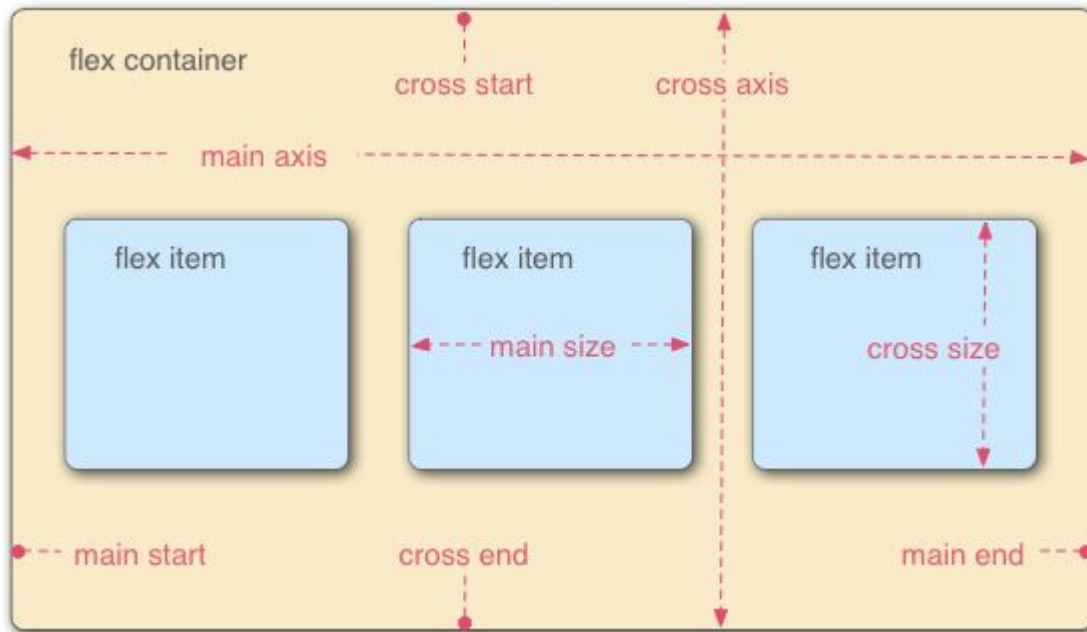


Flexbox



- Flexbox o simplemente flex, es un nuevo módulo que nos presenta CSS3.
- Se basa en un conjunto de propiedades que intentan facilitarnos la tarea de posicionar los elementos.
- De alguna forma, reemplaza a la propiedad float.

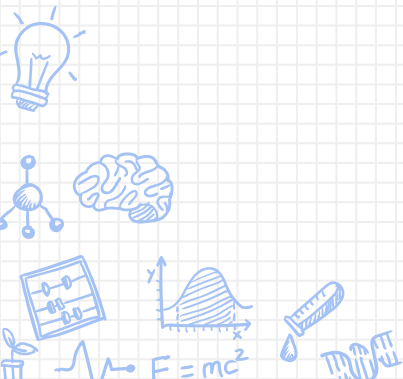
Flexbox

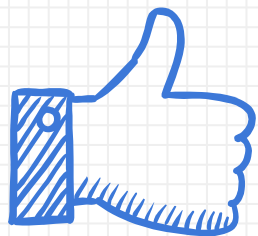


A collection of blue line-art icons in the top-left corner, including a molecular structure, an atom, a beaker with a chemical reaction, the chemical formula H_2O , a globe, a hand, a rocket, and a network diagram.

Desafío:

Completar los 24 niveles del Flexbox Froggy

A collection of blue line-art icons in the top-right corner, including a calculator, a hexagonal molecule, a globe, a plug, a book, a cell, a microorganism, a magnet, a star, and a test tube with a plant.



SE ACABÓ!

Dudas?

Escriban a:

✕ federico.mirandaa@gmail.com