**HTML**

<article> 🡪 usar si se puede leer solo sin necesidad del resto de la página.

<section> 🡪 se puede usar sin encabezado, siempre y cuando conceptualmente sea una sección de la página web.

**GIT**

Git 🡪 permite guardar en la nube y administrar cambios de líneas de código guardando en puntos de control (versiones). También permite convertir tu máquina en un servidor.

GitHub 🡪 repositorio remoto, trabajo de programación en conjunto, red social de Git.

Terminal 🡪 ejecutar programas y consultar cosas en el sistema operativo a través de comandos (Windows: cmd).

Trabajar con Git Bash (terminal). Todos los comandos de Git comienzan con git.

Cd 🡪 change directory, cambiar directorio/ruta de carpeta (el comando solo me lleva a mi carpeta de usuario).

Tecla *Tab* 🡪 trae directorio sugerido.

Pwd 🡪 print working directory, ver donde estoy parado en el directorio.

Q 🡪 salir.

Ls 🡪 lista contenido donde estoy parado.

Ls -l 🡪 listar uno por línea.

Ls -h 🡪 listar más amigable al usuario (KB, MB, …).

Ls -a 🡪 . carpeta actual, .. volver a una carpeta superior.

*Comando* --help 🡪 ayuda del comando.

Mkdir 🡪 crear carpeta de directorio.

Clear 🡪 limpiar pantalla terminal.

Alias 🡪 crear comando (alias nombre = ‘comando’).

Rm 🡪 eliminar archivo.

Rm - r 🡪 eliminar directorio (**MUCHO CUIDADO**).

Rm – i 🡪 eliminar interactivo, solicita confirmación antes de eliminar (y/n).

Git help 🡪 ayuda de git.

Git help *comando* 🡪 ayuda de un comando específico.

[…] 🡪 opcional, si no se indica hace la acción default.

. 🡪 todos los archivos de la ubicación donde estás parado.

Git init 🡪 crear un repositorio local git.

Git add 🡪 añadir archivo a Git para que sea controlado por Git.

Git commit 🡪 crear punto de restauración.

Git commit -m 🡪 incluir un mensaje al commit para seguimiento.

Git commit -a 🡪 commit todo lo que haya disponible para hacer commit.

Git status 🡪 verificar status de cambio de los archivos de mi repositorio.

Git config 🡪 configuración del repositorio.

Git config user.name “*nombre*” 🡪 definir nombre de usuario.

Git config user.email *mail@mail.com* 🡪 definir mail de usuario.

Git necesita que cada commit tenga un usuario autor (nombre y mail).

Archivos que empiezan con punto (.) son ocultos

GitHub 🡪 Fork: copiar un repositorio de otra persona para modificarlo.

Pull request: proponerle al dueño original del repositorio tomar las modificaciones hechas.

Git clone 🡪 copiar un repositorio remoto en tu repositorio local.

Branch 🡪 distintas ramas del código principal.

Master 🡪 branch principal.

Git branch -v 🡪 listar todos los branch del repositorio.

Git checkout *nombre-branch* 🡪 cambiar posicionamiento a cierto branch específico.

Git checkout -b *nombre-branch* 🡪 crear nuevo branch.

Git fetch 🡪bajar de repositorio remoto a repositorio local.

Git merge 🡪 bajar de repositorio local a directorio local.

Git pull 🡪 git fetch + git pull.

Git reset --soft HEAD~# 🡪 volver atrás cambios de un # de commits determinado. Luego hay que hacer commit para guardar los cambios.

Package Control 🡪 bajar para administrar plugins de Sublime Text, copiar código en consola de Sublime.

**CSS**

Style 🡪 atributo de las etiquetas para poner estilo a las mismas en el HTML definiendo ciertas propiedades (color de fondo, color de letra, tipo de letra, tamaño de letra, etc.). Se conoce como definir estilo in line.

<style type=”text/css”> … </style> 🡪 colocar estilos dentro del archivo HTML para no tener que crear una hoja de estilos (archivo CSS) aparte.

<link rel="stylesheet" type="text/css" href="./main.css"> 🡪 da estilo a un HTML relacionándolo con un archivo CSS separado.

Component CSS - Suite CSS 🡪 metodológica para aplicar estilos sistemáticamente a un HTML. Se utiliza una convención para clasificar etiquetas y que facilita su identificación y hace que la aplicación de estilos tenga una mejor performance. Se trabaja con paquetes de CSS que funcionan como módulos que pueden escalarse.

Clase componente 🡪 encapsula las declaraciones requeridas para dar estilo a un elemento en el contexto de un componente.

* *NombreClase* (elemento)
* *NombreClase-Componente* (componente del elemento)

Clase utilitaria 🡪 aplica una regla de estilo a un patrón sencillo y universal.

* u-*NombreClase* (regla genérica de estilo, ej: u-TextoRojo)

Usar class en vez de id 🡪 performance más rápida para los navegadores.

Inline no permite controlar ancho ni alto del elemento.

Definir al inicio de todo un font-size específico y controlar los tamaños de letra con la unidad de medida em como porcentajes de dicho font-size inicial.

Si no se define ancho y tiene propiedad display = block o inline-block, no se puede centrar un elemento horizontalmente.