

Clase01: 30 Abril del 2024

Comandos visual code

- **Windows:** ==CTRL+SHIFT+P== : Abre el panel de control que tiene VSC.
- **Windows:** ==CTRL+P== : Abre panel para cambiar de diferente archivo que tengas en tu carpeta.
- **Windows:** ==CTRL+B== : La pestaña que te muestra tus archivos.
- **Windows:** ==CTRL+D== : Te permite cambiar el nombre de varias palabras/variables del mismo nombre.
- **Windows:** ==CTRL+F== : Te ayuda a buscar palabra/variable en el código.
- **Windows:** ==CTRL+S== : Es la opción para guardar.
- **Windows:** ==ALT + or -== : Si está abierta pestañas te permite cambiar a otra.
- **Windows:** ==ALT + up or down== : Mueve toda la línea de acuerdo a la flecha.
- **Windows:** ==SHIFT+ALT+ up or down== : Copia toda la línea a la siguiente.
- **Windows:** ==SHIFT+ALT+A== : Pone para comentar una línea.
- **Windows:** ==CTRL+K+C== : Pone toda la línea como comentario.
- **Windows:** ==CTRL+T== : Permite mostrar todos los símbolos.
- **Windows:** ==CTRL+SPACE== : Te ayuda a autocompletar una palabra.
- **Windows:** ==CTRL+x== : Borra una línea.
- **Windows:** ==CLICK en el archivo + F2== : Cambia el nombre del archivo.

Comandos de la terminal

- pwd : Me permite saber en qué directorio estoy trabajando.
- touch : Para crear un archivo. Ej: `touch readme.md`
- code : Para crear archivos. Ej: `code readme.md`
- touch "" >> texto2.txt : Dentro de las comillas escribo lo que quiero que se escriba al crear el archivo --
> Ej: `touch "hola mundo" >> texto2.txt`

Git y Github



Git es un sistema de control de versiones rastrea los cambios que se han hecho en un conjunto de archivos es decir, un proyecto.

GitHub es un servicio que almacena repositorios en sus servidores y control de versiones usando git.

Recomendaciones de otros comandos que tiene markdown

- Si queremos enumerar hacemos lo siguiente:
 1. Linea 1
 2. Linea 2
 1. Linea 1
 2. linea 2
 3. Linea 3
- Si queremos poner otro signo para ideas hacemos lo siguiente:
 - Linea 1
 - Linea 2
 - Linea 1
 - linea 2
 - Linea 3

Clase03: 31 Abril del 2024

Markdown



Es un formato que nos permite generar archivos xml y html utilizando una sintáxis bastante clara y sencilla y sobre todo fácil de escribir.

- markdown (para convertir a pdf mi markdown):
 1. Windows: CTRL+SHIFT+P
 2. Despues busco la opcion de "markdown export pdf"
 3. Se crea automaticamente un archivo pdf de tu markdown
- `t` * palabra o texto *
- ****** palabra o texto ******
- ******* palabra o texto *******
- windows: ALT+9+6 (') or windows: ALT+1+2+6 (~)
 - Nos permite poner una parte del codigo para analizar despues de (``) tenemos que poner el lenguaje de progrmacion
 - Ejemplos: 1. `java public int sumar(){ int i=10; }` 2. `java public clas Hola{ /* clase principal */ public class void main{ System.out.println(); } }`
- Para hacer cuadros:

Columna 1	columna 2
-----------	-----------

Columna 1	columna 2
A	B
C	D

- Comandos de git
 1. git init : Crea un carpeta oculta en tu carpeta de proyecto y va aparecer tus archivo con "U".
 2. git status : Te muestra que archivos has guardado --> (rojo: no esta guardado and verde: esta guardado).
 3. git commit : Para guradar el proyecto.
 4. git checkout -b "Nombre de rama" : Crea y cambia de rama.
 5. git branch : Crea una rama.
 6. git push : Subo todo al github.
 7. git pull : Nos ayuda a traer los cambios hechos en la nube.
 8. git clone url : Baja el archivo de algun proyecto.
 9. git ignore : Dentro del archivo puedes poner archivos q va ignorar o no va a guardar.
 10. git config --global user.name "nombre" : Colocar nombre de ususario.
 11. git config user.name : Verificar si esta su nombre de ususario. (**Es lo mismo para el email solo cambia user.email**)
 12. git config --global init.default branch "main" : Cambiar de master a main.
 13. git commit -m Descripcion :Agregar cambios al repositorio local y una descripcion del cambio.
 14. git log :Muestra historial de commits. Si agregas "--oneline", aparece mas compacto.
 15. git add . : Agregar cambios al commit.
 16. git config --global core.editor : Asociar al vscode.
 17. git branch -m nombre actual nombre nuevo : Cmabiar nombre de rama sin estar en la rama a la que se va a cambiar.
 18. git branch -d : Eliminar rama (locales no publicadas).
 19. git merge nombre-de-rama : Fusionar rama con main.
 20. git remote : Ver nombre del repositorio remoto.
 21. git fetch origin : Crear una rama local con los cambios de la rama remota.
 22. git push origin -d : Eliminar rama remota.