

# Bienvenidos a la práctica 9

Es esta práctica vamos a realizar una presentación de las siguientes extensiones

- Estamos utilizando varias extensiones de VS Code, estas son:
  - Marp for VS Code
  - Markmap
  - PlantUML o PlantUML Previewer

La extencion Marp for VS Code nos permite realizar presentaciones desde un archivo de texto y la extencion lo interpretara para generar la presentacion

Para colocar las letras en negrita, italica, codigo, etc, es necesario colocar ciertos simbolos para que lo pueda interpretar

- Para colocar una palabra en negrita es necesario que la palabra deseada este entre cuatro asteriscos (\*\* negrita \*\*)
  - **Negrita**
- Para colocar una letra en italica es necesario que la palabra deseada este entre dos asteriscos (\* italica \*)
  - *Italica*
- Para colocar una palabra tachada es necesario colocar la palabra entre este cuatro de este simbolo ~~ ( ~~ tachado ~~ )
  - ~~Tachado~~

- Para colocar código es necesario que este entre el símbolo del acento (`código`)
  - `System.out.println("Hola Mundo");`
- Para hacer que se crea una nueva diapositiva es necesario colocar tres guiones seguidos (---)

Ahora para hacer títulos es necesario colocar el simbolo del gato (# Título 1) antes del titulo

- # Título 1

- Para colocar subtítulos es muy sencillo, solo se agrega simbolo del gato (## Título 2)

- **Título 2**

- Para agregar mas subtítulos solo se van agregando sucesivamente mas simbolos de gato (### Título 3...)

- **Título 3**

Para agregar imagenes a la presentacion se debe hacer:

- Tener la imagen en la misma carpeta donde se esta creando la presentación
- Colocar la siguiente linea "`![width:1000px].(nombredeimagen.png)`"
- El numero seguido de los dos puntos es para ajustar el tamaño de la imagen, se puede ir cambiando

# Universo





Tambien se pueden agregar tablas a la presentación

- Para ello se debe de utilizar el siguiente simbolo (|), este simbolo marca donde inicia y termina cada celda de la tabla (|Titulo 1|Titulo 2|)
- Para sombrear alguna celda, es necesario colocar (|---|) debajo de la celda deseada

# Tabla

Titulo 1	Titulo 2
celda 1	celda 2
celda 3	celda 4

La siguiente extensión a revisar es Markmap. Esta herramienta nos permite crear mapas de ideas dinámicos de una manera muy sencilla

- Su proceso de creación es muy similar a agregar títulos en esta presentación.
- Recordando que para colocar títulos y subtítulos solo es necesario ir colocando el símbolo de #

- Entre menor sea el numero de # mayor jerarquia tiene la rama
  - Un solo gato denota el tema principal (# Tema principal)
  - Dos gatos denota subtemas del principal (## Subtema)
  - Tres gatos denota subtema del subtema (### Subtema del subtema) y asi sucesivamente
- Para ir creando el mapa de ideas es esencial que se coloquen cada linea de forma ordenada o si no, colocaran las ramas o subramas en sitios donde no corresponden

- Un ejemplo:
  - •# Animal
  - •## Terrestre
  - •## Aereo
  - •### Perro

En este ejemplo se da a entender que el tema principal son los animales, que tiene dos subramas llamadas terrestre y aereo, y dentro de la subrama aereo se encuentra perro

Esto esta mal porque el perro no debe de estar en esta rama si no en otra, para ello es necesario que se coloque de la siguiente manera

- ·# Animal
- ·## Terrestre
- ·### Perro
- ·## Aereo

Aqui decimos que perro se encuentra dentro de la ramma terrestre

Por ultimo la extensión PlantUML Previewer. Esta herramienta nos permite realizar diagramas

- Para empezar con los diagramas primero se necesita colocar la siguiente linea *@startuml* y para terminar se coloca *@enduml*
- Para hacer los diagramas de clases es como si colocaras codigo de las clases, se agrega similar a la hora de codificar, agregando los atributos y metodos de una clase

- Por ejemplo:
  - `class TransporteAcuatico{`
    - `-velocidad: int`
    - `-capacidad: String`
    - `+aumentarVelocidad():void`
    - `}`
- Para saber si una clase hereda o si es una interfaz es necesario poner el nombre de la clase y con ayuda de los siguientes simbolos <|-  
- (simbolizando una flecha) podemos decir que clase hereda, si es interfaz o clase abstracta



- Ejemplo:

MediosTransporte<|-- TransporteAcuatico

TransporteAcuatico<|-- Barco: interfaz