

P1.1. INTRODUCCIÓN AL ENTORNO DE TRABAJO

ÍNDICE

1. INTRODUCCIÓN.....	2
2. QUE TIENES QUE ENTREGAR.....	2
3. PRÁCTICA.....	3
3.1. Entorno de Trabajo	3
3.2. Trabajando con un Archivo de Markdown.....	4
3.3. Trabajando con HTML	5
3.4. Trabajando con JSON	7
3.5. Trabajando con XML	9

1. INTRODUCCIÓN

Los lenguajes de marcas son normalmente manipulados con una herramienta de trabajo sencilla como un editor de código. En esta práctica vamos a utilizar el Visual Studio Code que viene instalado por defecto en los ordenadores Lliurex de la Generalitat.



Visual Studio Code

Es una herramienta de trabajo multi propósito, sirve tanto para lenguajes de programaciones estructurados, modulares, enfocados a objetos y también para lenguajes de marcas.

No estás obligado a hacer uso del Visual Studio Code (VSC a partir de ahora), pero si que es recomendado.

Elige tu editor de código para empezar la práctica.

2. QUE TIENES QUE ENTREGAR

Vamos a trabajar todo el rato con una carpeta que tienes que crear en tu ordenador. La carpeta tendrá el nombre de "PRÁCTICA 1 – Nombre Apellido". Obviamente, debes sustituir el nombre y el apellido con el tuyo mismo.

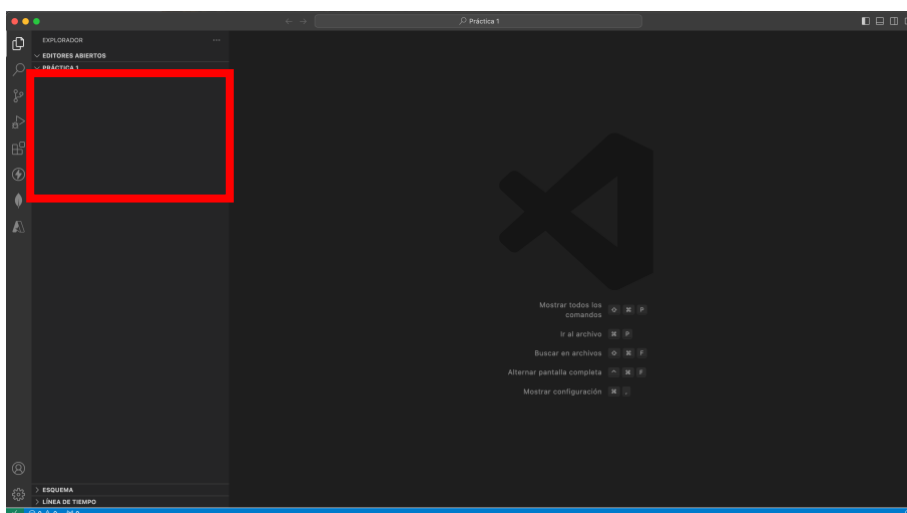
En esta carpeta vamos a hacer modificaciones durante la práctica y crear nuevos archivos. En el fi de la práctica tienes que subirla a AULES comprendida en .zip.

3. PRÁCTICA

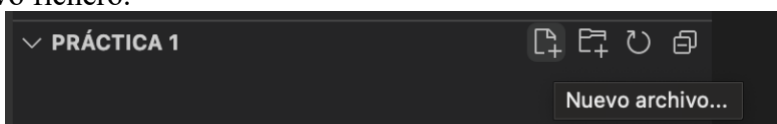
3.1. Entorno de Trabajo

El Visual Studio Code trabaja con un directorio para abrir su panel lateral. Ahí lo hacen la mayoría de editores de código. Ves a *Archivado > Abrir Carpeta* para elegir tu carpeta de trabajo.

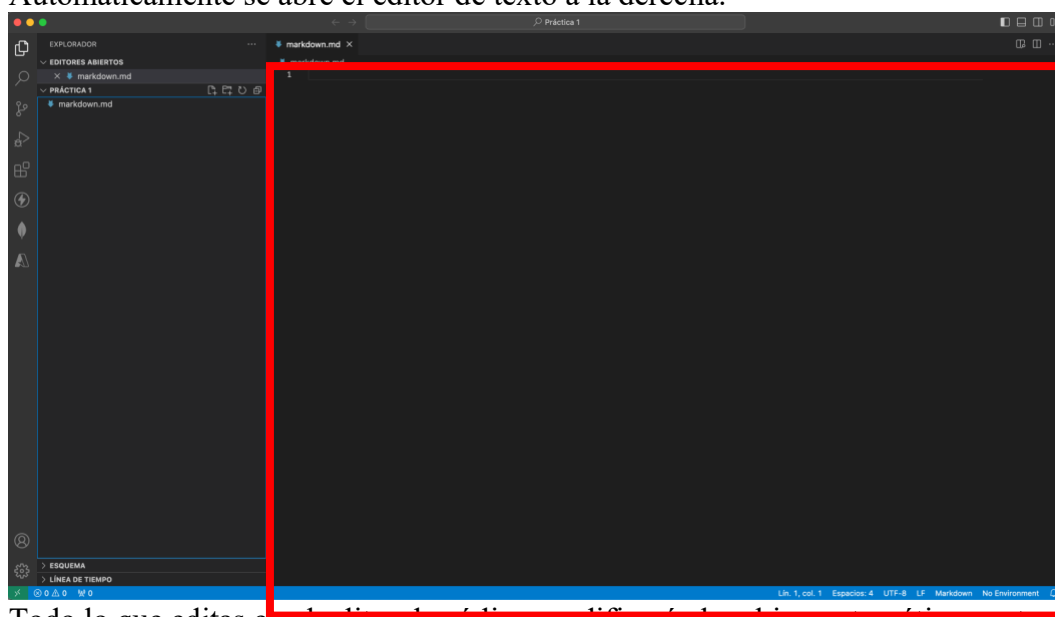
Una vez estás en la carpeta de trabajo, verás que no aparece ningún fichero en la barra lateral.



Crea un fichero con el nombre 'markdown.md', puedes hacer clic derecho o clicar en el icono de nuevo fichero.



Automáticamente se abre el editor de texto a la derecha.



Todo lo que editas en el editor de código modificará el archivo automáticamente. Si quieres desactivar el auto-guardado puedes ir a *Archivo > Auto-guardado* para desactivarlo o activarlo. Es más cómodo trabajar con el auto-guardado activado. La barreta de encima es una barreta multi-función, puede ser usada por:

- Buscar archivos.
- Ejecutar pedidos.
- Escoger extensiones.
- Ejecutar macros.
- Hacer búsqueda dentro de archivo.

Las funciones las iremos conociendo poco a poco.

También debes darte cuenta de que el archivo ha cambiado el icono cuando le has dado nombre. Esto es porque VSC reconoce automáticamente el lenguaje en función del contenido del fichero o en función de lo que hay dentro.

3.2. Trabajando con un Archivo de Markdown

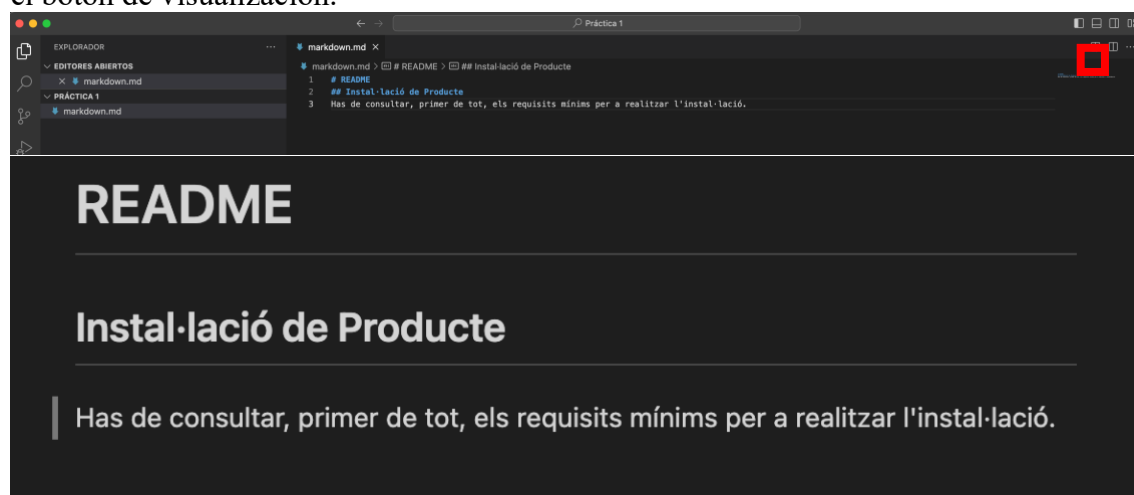
Markdown, como ya hemos visto, es un lenguaje de marcas muy sencillo y orientado a tener una traducción inmediata a HTML. Si agregas las siguientes líneas al archivo verás como VSC reconoce las marcas.

```
# README
```

```
## Instalación de Producto
```

Debes consultar, primero de todo, los requisitos mínimos para realizar la instalación.

Si quieres ver cómo luce el código a un visualizador de Markdown, puedes hacerlo con el botón de visualización.



Ahora, debes hacer un documento Markdown de una página de GitHub de tu propio producto de software libre. Como Markdown es muy sencillo, debes buscar en la web cómo hacer un documento que contenga la siguiente información:

- Icono del producto (puedes utilizar cualquier icono de Internet).
- Título
- Apartado de "Instalación del producto" con enlaces externos a otras webs (puede ser arbitrario, no hace falta que tenga sentido).

- Apartado de las características básicas del producto, es decir, que hace que lo hace tan bueno. Tienes que hacer una lista sin numerar.
- Apartado de contacto contigo, con la dirección de email con un Link del tipo 'mailto:'.
- Puedes ser creativo y meter ejemplos de código como el siguiente. No te vuelvas loco si no lo encuentras. Lo importante es aprender la filosofía básica.

Exemple:

```
>>> import tensorflow as tf
>>> tf.add(1, 2).numpy()
3
>>> hello = tf.constant('Hola Llenguatge de Marques!')
>>> hello.numpy()
b'Hola Llenguatge de Marques!'
```

Si no sabes algo, es normal. Acá tienes ejemplos de páginas de GitHub (de excelentes software libres que hacen mucho bueno en la comunidad abierta) donde puedes encontrar el archivo Readme.md. Puedes analizarlo para hacer que el tuyo funcione:

- [PrusaSlic3r](#)
- [Servidor Nextcloud](#)
- [Photopea](#)
- [Tensorflow](#)

3.3.Trabajando con HTML

Ahora puedes crear otro fichero dentro de la carpeta, llamarlo 'index.html'. Verás que también cambia el icono y lo reconoce.

Puedes empezar copiando y pegando el siguiente código en el archivo.

```
<!DOCTYPE html>
<html>
  <body>
    
    <h1>README</h1>
    <h2> Com instal·lar el meu Software</h2>
    <p>Has de tindre en conter que els requisits mínims son:</p>
    <ul>
```

```
<li><a href="www.ubuntu.com">Ubuntu</a></li>

<li><a href="www.virtualbox.com">Virtual Box</a></li>

</ul>

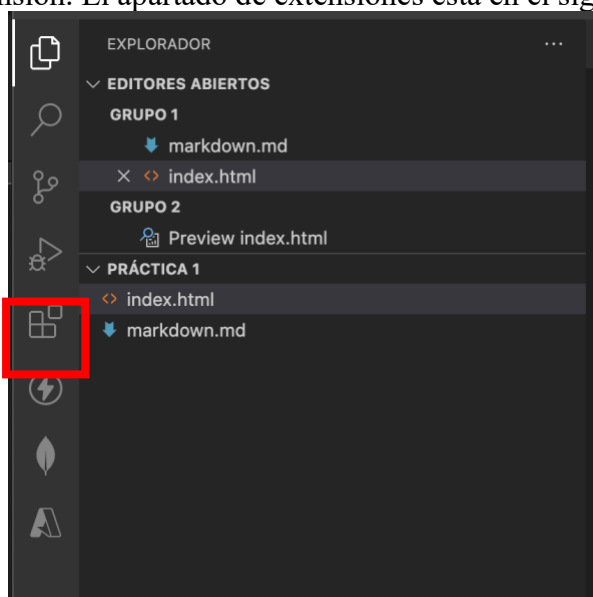
Ara pots executar el següent codi:

<pre>sudo apt-get install lliurex</pre>

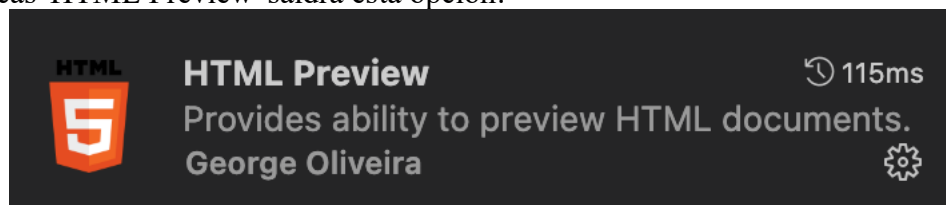
</body>

</html>
```

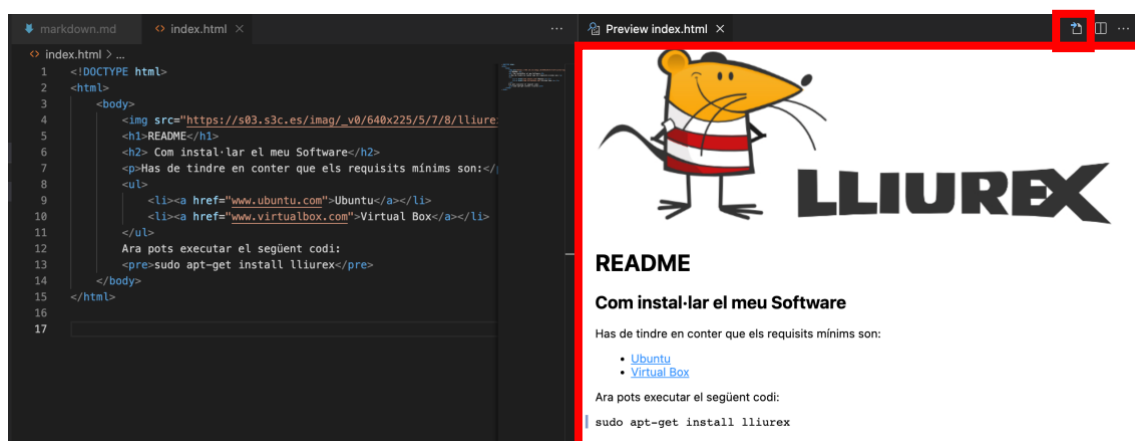
Como puedes ver, no hay opción de visualización en el VSC para HTML. Hay que instalarla como extensión. El apartado de extensiones está en el siguiente icono:



Si buscas 'HTML Preview' saldrá esta opción:



Si la instalas, verás que ya aparece el icono como pasaba con el Markdown.



Está claro que no dominas HTML y no hace falta para la práctica. Lo interesante es que le metes curiosidad por tratar de descubrir qué hace cada apartado.

Trata de imitar, con la ayuda de Internet y del ejemplo que vos he dado, la página de Markdown que has hecho antes de tu software. Incluye los mismos puntos.

3.4.Trabajando con JSON

JSON es un tipo de archivo muy utilizado en la Internet. Crea un nuevo archivo llamado 'temps.json' y abralo.

Puedes empezar copiando y pegando el siguiente fragmento de código:

```
{ "coord":  
  { "lon": -2.94,  
    "lat": 43.26 }, "weather":  
  [ { "id": 802, "main": "Clouds",  
    "description": "scattered clouds",  
    "icon": "03n" },  
    ],  
  "base": "stations", "main":  
  { "temp": 280.71, "feels_like": 277.32, "temp_min": 279.15, "temp_max": 283.71,  
    "pressure": 1021,  
    "humidity": 81  
    },  
  "wind": { "speed": 3.1, "deg": 240 }, "clouds": { "all": 40 },  
  "dt": 1576087793,  
  "sys":  
    { "type": 1,  
      "id": 6438, "country": "ES",  
      "sunrise": 1576049618,  
      "sunset": 1576082177 }, "timezone": 3600, "id": 3128026, "name": "Bilbao",  
  "cod": 200  
  }
```

Lo que ves es un ejemplo de contestación de la API OpenWeather para un lugar determinado. ¿A que no se ve nada? Esto es porque no tiene un orden bueno para que una persona lo pueda leer. Para arreglarlo debemos instalar una extensión: "Prettify JSON".

Una vez instalada, ves en el archivo JSON y haz clic en la barra multi-función. Selecciona la opción de ejecutar pedidos y después escribe "Prettify JSON". ¡Magia! Modifica el archivo para las coordenadas:

- Longitud: 39.47
- Latitud: -0.33

También debes hacer que el nombre de la ciudad sea 'Valencia', y además que el tiempo sea "Sunny".

Sigue jugando y fijándote en la jerarquía de árbol. Siempre puedes meterlo bonito otra vez con el "Prettify JSON".

3.5.Trabajando con XML

Por último, vamos a trabajar con un archivo XML. Ya sabes que hacer, crea el archivo 'llibres.xml'. Puedes empezar copiando y pegando la siguiente parte de código:

```
<?xml version="1.0"?>
<cataleg>
  <llibre id="131">
    <titol>Tirant Lo Blanc</titol>
  </llibre>
</cataleg>
```

Como ves, es un chico catálogo de libros. Es muy fácil entender la dinámica del XML. Por ahora, sólo tienes que hacer lo siguiente respetando la jerarquía y normas básicas:

- Crear otro libro con código "132" y el título que tú veas.
- Crear otro libro con código "133" y el título que tú veas.
- Añadir una descripción a los nuevos libros.
- Añadir un código de ubicación a los tres libros, coloca códigos de ubicaciones diferentes.
- Hacer una base de datos en el mismo fichero con todos los códigos de ubicación y el nombre del lugar real.