MANEJO DE VISUAL CODE

EJERCICIO 1

Objetivo: Crear nuestra primera página web en html 5 con el editor Visual Code. Para familiarizarnos con las características básicas del editor. De tal forma, que lo hagamos lo más rápida posible. Para ello, Visual Code dispone de una extensión llamada **Emmet** que nos ayudará a tal fin. Generalmente está instalada pero si no está deberemos ir a la sección de extensiones para instalarla.

1. Crea la primera página web con el nombre de ejercicio1.html. Deberá contener los siguientes elementos:
   - Una cabecera de nivel 1. El texto es “Visual Code”.
   - Una linea horizontal.

- Busca una imagen en internet e insértala en la página.
- Un párrafo con el siguiente texto. “Vscode es un excepcional editor de textos que aporta muchas características útiles a la hora de programar o editar código. El editor está cargado de funcionalidades útiles y cómodas desde el punto de la usabilidad y eficiencia, utilizando el método geek y convirtiendo nuestro trabajo de edición de texto en una experiencia cada vez más sencilla y agradable, a medida que vamos aprendiendo a utilizar todas sus funcionalidades.”

2. Busca información sobre Emmet en este link y contesta a las siguientes preguntas.
    1. Qué hace las siguiente combinaciones:
        1. `ul>li*5>strong+em`
        2. `div>(header>ul>li*2>a)+footer>p`
        3. `(div>dl>(dt+dd)*3)+footer>p`
        4. `div#header+div.page+div#footer.class1.class2.class3`
    2. ¿Con que código se generará el siguiente fragmento html?

```
<div>
  <ul>
    <li></li>
  </ul>
</div>
```

    3. ¿Qué hace el operador ^?. Pon un ejemplo.

EJERCICIO2

Objetivo: Ordenar el siguiente código html. Para ello podemos utilizar el siguiente combinación de teclas `Alt + ↑ / ↓` , con el que subes y bajas lineas.

```
<ul>
  <li>
    <span>línea 4</span>
    <span>Nada importante 4</span>
  </li>
  <li>
    <span>línea 3</span>
    <span>Nada importante 3</span>
  </li>
  <li>
```

```
      <span>línea 2</span>
      <span>Nada importante 2</span>
    </li>
    <li>
      <span>línea 1</span>
      <span>Nada importante 1</span>
    </li>
  </ul>
  ls -la

  <span> hola</span>
  <span>amarillo</span>
  <span>rojo</span>
  <span>verde</span>
  <span>naranja</span>
  <span>morado</span>
  <span>negro</span>
  <span>blanco</span>
```

EJERCICIO 3

Para comentar todas las lineas seguidas podemos utilizar, una vez que se haya seleccionado , la siguiente combinación de teclas `Ctrl +shift+ /`. Comenta toda la lista anterior. A continuación descomentala de la misma forma que antes.
Ahora vamos a comentar solo el bloque que seleccionemos. Para ello selecciona, por ejemplo linea 1, y utiliza la combinación de teclas `ctrl+shift+a` .
Para saber que shortcut es el correspondiente, ya que según el sistema operativo cambia las combianciones, deberemos ir a File-→ preferences → keyboards shortcuts `[ctrl+K ctrl+S ]`

EJERCICIO 4

A partir del código del ejercicio 1 copiar . Ahora manteniendo pulsado el Ctrl o el Alt y haciendo click sobre el link del fichero nos da un error. Dando la posibilidad de crear ese fichero en esas carpetas de forma automática. Créalo.

EJERCICIO 5

Crea un fichero **importar.ts** con el siguiente código:

```
import { saludar } from './funciones';
 const saludo = saludar( 'Thanos' );
console.log(saludo);
```

Ahora crea el fichero **funciones.ts** con la siguiente función.

```
export function saludar( nombre: string ) {
 return `Hola ${ nombre }!!!`;
}
```

Con la tecla `F12` podemos ir a la definición de esa función y con el conjunto de teclas `Ctrl + shif+ F10` lo podemos ver en la misma ventana. (revisa los shortcuts).

EJERCICIO 6

Para borra lineas podemos utilizar la combinación `Ctrl + Shift + K` pero lo interesante es por ejemplo que seleccionemos todas las ocurrencias de una selección y borrar todas las lineas donde aparezcan .
Para seleccionar todas las ocurrencias utilizamos combinación Ctrl + Shift + L sobre la palabra a buscar. Ahora en borra todas las lineas que aparezca nada en el código html del **Ejercicio 2** . A continuación déjalo como estaba con la combinación `Ctrl +Z` y con la combinación `Ctrl + Shift + Z` rehaz el borrado.

EJERCICIO 8

Para evitar ninguna distracción vscode dispone del modo Zen. Para entrar en este modo tenemos la combinación `Ctrl + K Z` (primero el control + k, sueltas el control y aprietas la Z). Para quitarlo igual. También con Ctrl+B tenemos la opción de quitar la columna de la izquierda.

EJERCICIO 9

Ahora vamos a utilizar el emmet-wrap. Nuestro objetivo es seleccionar ls -la (**Ejercicio 2**) y envolverlo con la etiqueta `<code>` . Para ello deberemos ir a la paleta de comandos con `Ctrl + SHIFT + P` y buscar emmet-wrap y una vez que lo hemos encontrado apretamos enter. Seleccionamos el objetivo y escribimos en la paleta code.

Pero esto son demasiados pasos con lo que podemos abreviarlo insertando un **shortcuts. (Archivo --> preferencias-->Métodos abreviado de teclado**) para la función emmet-wrap. Claro, esta combinación de teclas no tiene que estar seleccionada previamente. Ahora crea el shortcut.

EJERCICIO 10

Vamos a copiar y pegar lineas directamente sin tantos pasos. Mediante la siguiente combinación de teclas `ctrl+shift+alt+down` (es útil que lo cambies) podemos clonar una linea o una selección. Ahora clona la linea `<span> Hola Mundo<span>` veinte veces.

EJERCICIO 11

Con la combinación de teclas `shift + Alt+ ↑ / ↓` podemos crear distintos cursores. Y con `shift + Alt+ →` expandimos la selección. El ejercicio consiste , con el código de span del **Ejercicio 2** , crear una clase con el contenido de la etiqueta del span con los mínimos pasos. Es decir :

```
<!-- Objetivo final -->
<span class="amarillo">amarillo</span>
<span class="rojo">rojo</span>
<span class="verde">verde</span>
<span class="naranja">naranja</span>
<span class="morado">morado</span>
<span class="negro">negro</span>
<span class="blanco">blanco</span>
```

EJERCICIO 12

En el siguiente ejercicio seguiremos utilizando los multicursores de tal forma que el siguiente código.No hay ningún shortcuts para la transformación de minusculas a mayúsculas. Crealo.

```
const hulk = 'brouce banner';
const Hawkeye = 'cinton francis';
const ironman = 'tony stark';
const spiderman = 'peter parker';
const viudaNegra = 'natalia romanova';
```

Se convierta en el siguiente con los menores pasos.

```
const hulk        = 'Brouce banner';
const Hawkeye     = 'Cinton francis';
const ironman     = 'Tony stark';
const spiderman   = 'Peter parker';
const viudaNegra  = 'Natalia romanova';
```

EJERCICIO 13

Este ejercicio es un poco más difícil porque el código no sigue un patrón determinado. Disponemos del siguiente código :

```
<span>amarillo</span>
<p>rojo</p>
<div-personalizado>verde</div-personalizado>
<bold>naranja</bold>
<otro-div-complejo>naranja-azul</otro-div-complejo>
```

Y nuestro objetivo es el siguiente:

```
<span class="amarillo">amarillo</span>
<p class="rojo">rojo</p>
<div-personalizado class="verde">verde</div-personalizado>
<bold class="naranja">naranja</bold>
<otro-div-complejo class="naranja-azul">naranja-azul</otro-div-complejo>
```

En este caso vamos a crear los multicursores mediante el ratón con el objetivo de seleccionar cada contenido que es de una palabra. Con `Ctrl + Doble click` podemos crear todos los multicursores y seleccionar cada palabra pero si te das cuenta el último dispone de dos palabras con lo cual deberás seleccionarlo con el ratón.

EJERCICIO 14

Pulsando `Ctrl + D`, seleccionamos la próxima ocurrencia de la palabra/string que tenemos seleccionada y se crea un cursor. Podemos utilizar está función para pasar de este código.

```
<span>Uno</span>
<span>Dos</span>
<span>Cuarenta y cinco</span>
<span>Un millón, ciento cincuenta mil</span>
```

```

```

Al siguiente.

```
ul>
    <li>Uno</li>
    <li>Dos</li>
    <li>Cuarenta y cinco</li>
    <li>Un millón, ciento cincuenta mil</li>
</ul>
```

EJERCICIO 15

Con lo que hemos aprendido de los multicursores y crea una array con los días de la semana.
Lunes
martes
miércoles
jueves
viernes
sábado
domingo.

El objetivo final será:

```
const dias= ['lunes', 'martes', 'miércoles', 'jueves', 'viernes',
'sábado',
'domingo'];
```

EJERCICIO 16

Para este ejercicio abre el fichero `ir-a-la-linea.css` y borrar la clase en la línea 7798. Esto se puede hacer de dos formas. Mediante Ctrl+P y después con los dos puntos o mediante Ctrl+G.

EJERCICIO 17

Un snippet es un conjunto de código que se genera de forma automática. Hay extensiones que crean este código. Pero si queremos crear nuestros snippet personalizados deberemos. Primero ir a **Archivo** → **Preferencias** → **Configurar fragmentos de usuario** a continuación nos saldrá la paleta de comandos y tendremos que seleccionar si lo queremos de forma global o para el entorno de trabajo actual.
Por ejemplo vamos a crear un snippet "ccc" tab para que nos muestre console.log ('hola'); .
En los ficheros de javascript y typescript. Para ello tendremos que modificar un fichero JSON. Siguiendo con el formato que nos va indicar cuando introducimos el nobre del snippet:

1. "la función del log"
2. prefix → la abreviatura de nuestro código.
3. Body → nuestro código (estudia lo que hace $1 $2)
4. description → breve descripción del funcionamiento.

EJERCICIO 18

Genera un snippet que te genere un array como el de días de la semana.

EJERCICIO 20

Visual code permite realizar búsquedas o reemplazos utilizando **Expresiones regulares**. Esto es una forma excelente de hacer cambios utilizando patrones. Para activar el soporte de expresiones regulares simplemente hay que pulsar el primer botón de la barra de

búsqueda (Ctrl+F → En la misma página , Ctrl+shift+F → para todo los ficheros simbolizado por los caracteres .* .Las expresiones regulares es muy amplio, pero es muy útil para automatizar tareas de búsqueda con patrones muy variables o desconocidos. Aquí algunos ejemplos:

```
^a  Línea que empiece por a
a$  Línea que acabe en a
.   Cualquier carácter
a\* Cero o más «a»
a+  Una o más «a»
a|b Carácter «a» o «b»
[aeiou] Una vocal minúscula
[^aeiou] Carácter no vocal minúscula
```

Podemos prácticas con las expresiones regulares aquí. En este ejercicio buscaremos las lineas que empiecen por <li.
Con el código del **Ejercicio 2**.

EJERCICIO 21

vscode permite extender sus capacidades mediante la instalación extensiones. Explica mediante un ejemplo las utilidades de las siguiente extensiones

1. Bookmarks
2. live Server.
3. Color Highlight