

Índice

Indice	1
Visual Code	2
Extensiones	2
Rem y Seteo del html	4
Creación de Elementos	5
CSS Grid	6
Media Querys	8

Visual Code

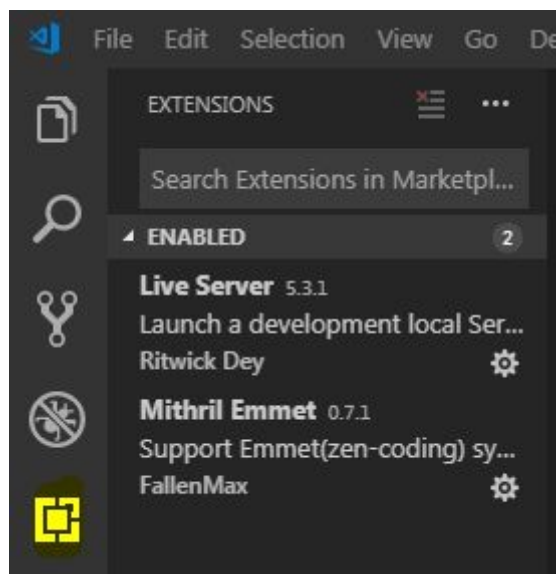
Extensiones

Todo el Proyecto será realizado en Visual Code, ya que contiene plugins que nos permitirán manejar de una manera fácil la creación de tag y otras utilidades.

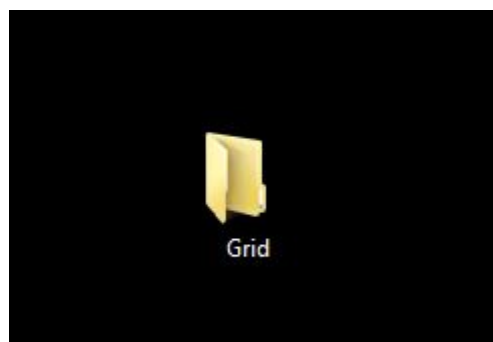
2 plugin que usaremos serán :

Live Server 5.3.1 : permite reflejar los cambios inmediatamente en el Proyecto.

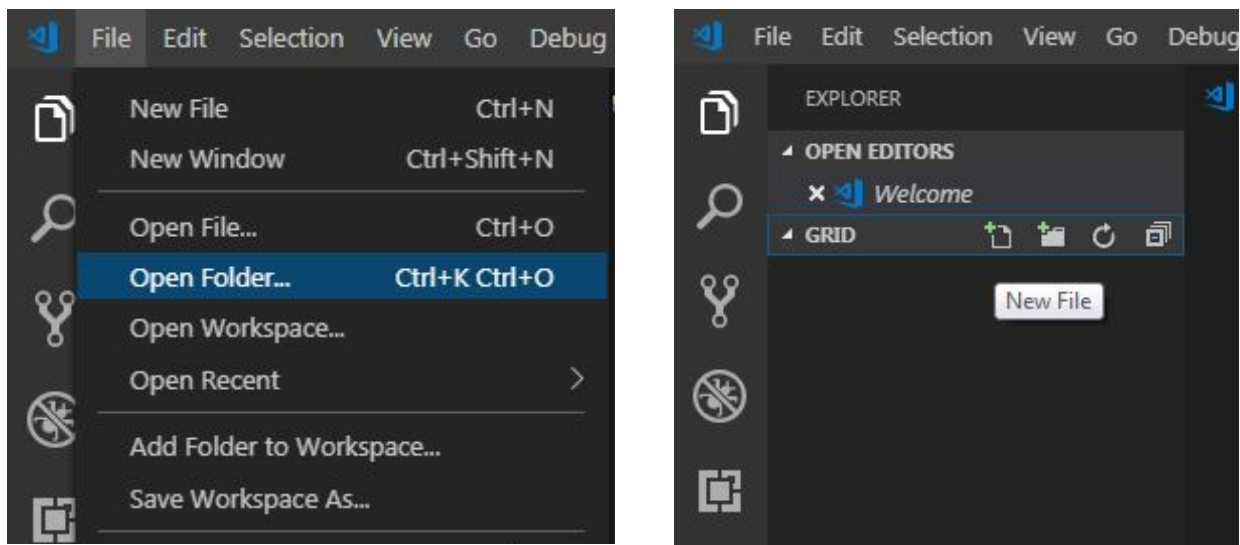
Mithril Emmet 0.7.1 : Brindara Ayudas al momento de la sintaxis, en la creación de Tags.



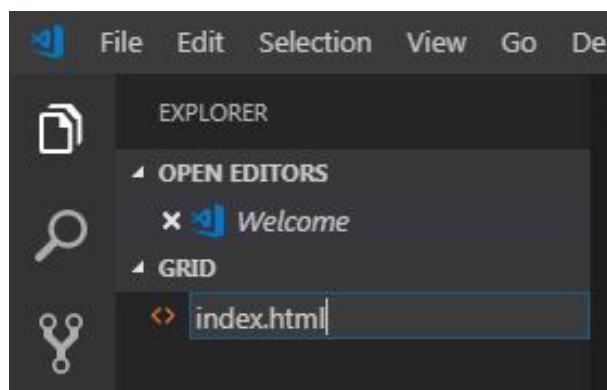
Creamos una carpeta nueva en este Caso la Llamaremos Grid.



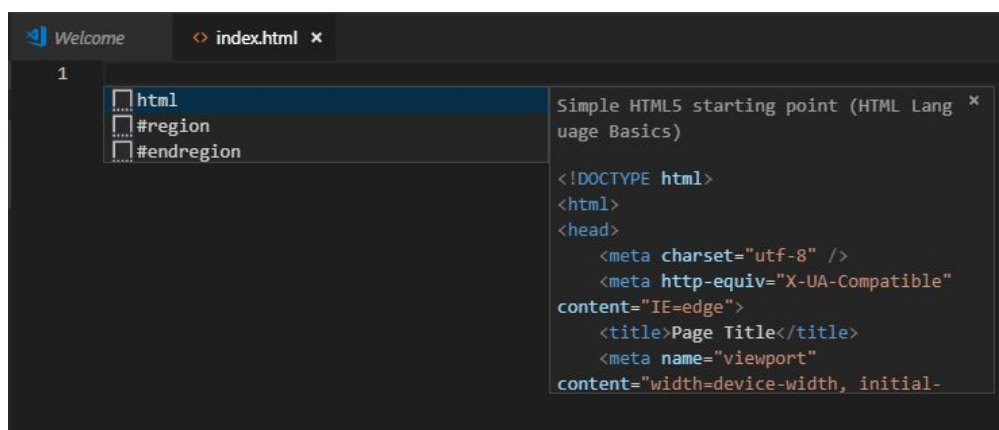
Seleccionamos desde Visual Code Open folder y elegimos la carpeta “Grid” que creamos



Con esto ya podremos crear distintos archivos y carpetas, a los archivos creados se les debe especificar la extensión manualmente.



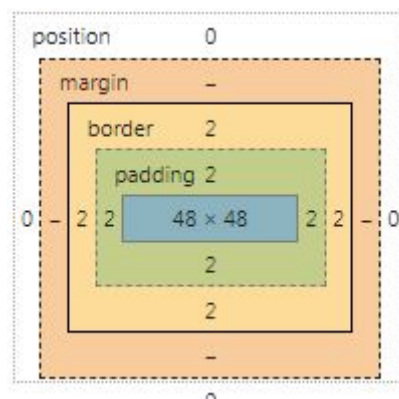
con las teclas Ctrl + Espacio se desplegará la ayuda, en este caso seleccionamos HTML y nos completará con una plantilla para crear el documento html.



Rem y Seteo del html

```
Welcome index.html # main.css x
1  html{
2      margin: 0;
3      padding: 0;
4      font-size: 10px;
5  }
```

siempre que comencemos un proyecto debemos setear los márgenes en 0 y el padding 0, la estructura de cada elemento es así para que sea más comprensible:



Al momento de indicar la medida rem estamos diciendo que se multiplique lo indicado en el html por la cantidad de rem, es decir **20rem serán $20 \times 10 = 200\text{px}$** .

cuando tenemos múltiples archivos css es recomendable crear un main, el cual se importará al proyecto, importamos solo uno, este main contendrá los import de TODOS los demás css, esto evitará llenar de importaciones el head del html.

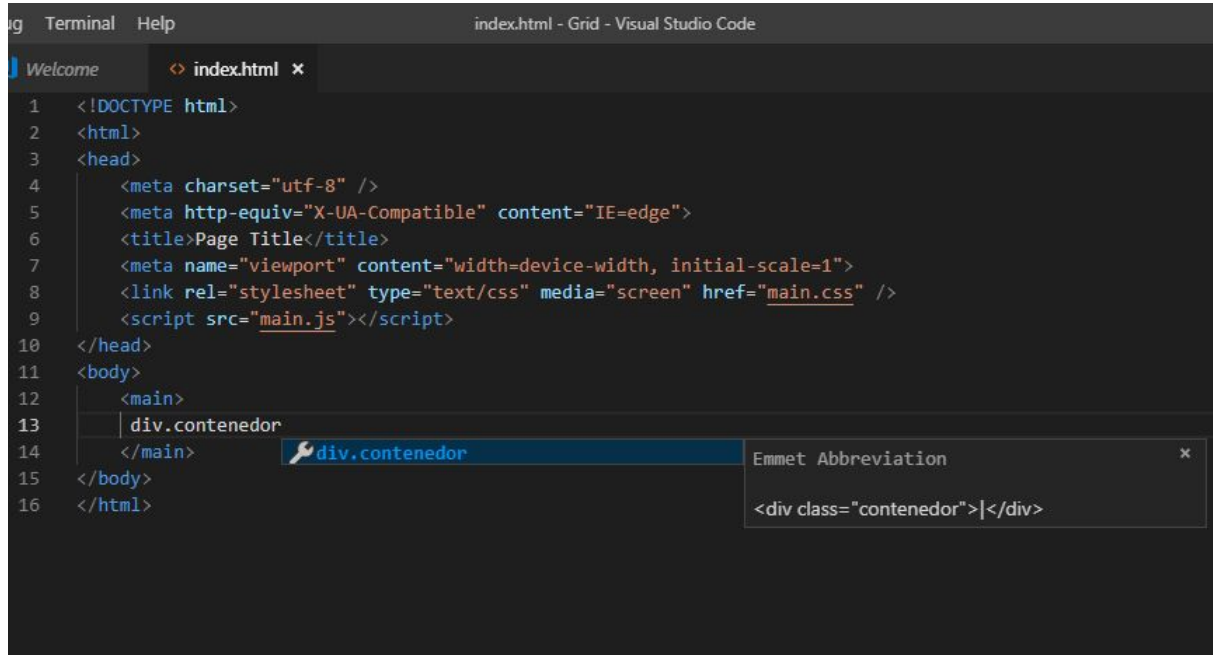
```
EXPLORER
└─ OPEN EDITORS
  └─ # main.css
  └─ CSS
    └─ # animacion.css
    └─ # main.css
    └─ # menuCircular.css
    └─ # style.css

# main.css x
1  @import url(/proyectos/LoremIpsum/css/menuCircular.css);
2  @import url(/proyectos/LoremIpsum/css/animacion.css);
3  @import url(/proyectos/LoremIpsum/css/style);

3  <head>
4      <meta charset="utf-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <title>Page Title</title>
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
9      <script src="main.js"></script>
10 </head>
```

Creación de Elementos

Podemos escribir Div. mas el nombre de la clase y el plugin emmet nor creará de inmediato el div con la clase asignada



The screenshot shows the Visual Studio Code editor with a file named 'index.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <title>Page Title</title>
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
9   <script src="main.js"></script>
10 </head>
11 <body>
12   <main>
13     | div.contenedor
14   </main>
15 </body>
16 </html>
```

An Emmet abbreviation suggestion box is open, showing the text: `<div class="contenedor">|</div>`.

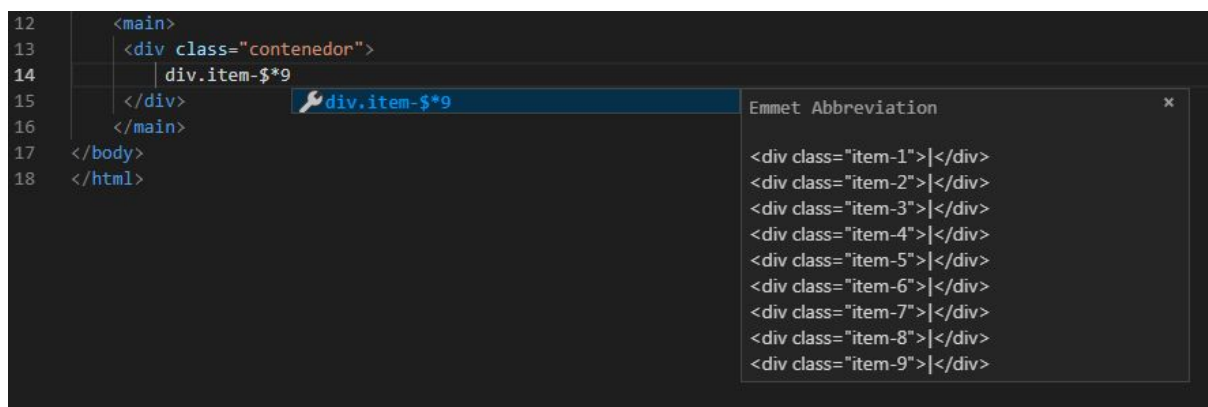


The close-up shows the following HTML structure:

```
<body>
| <main>
| | <div class="contenedor"></div>
| </main>
</body>
```

asi como tambien crearemos 9 elementos escribiendo:

div. + clase (item-) + \$ (será como un autoincrementable) + * por la cantidad de elementos



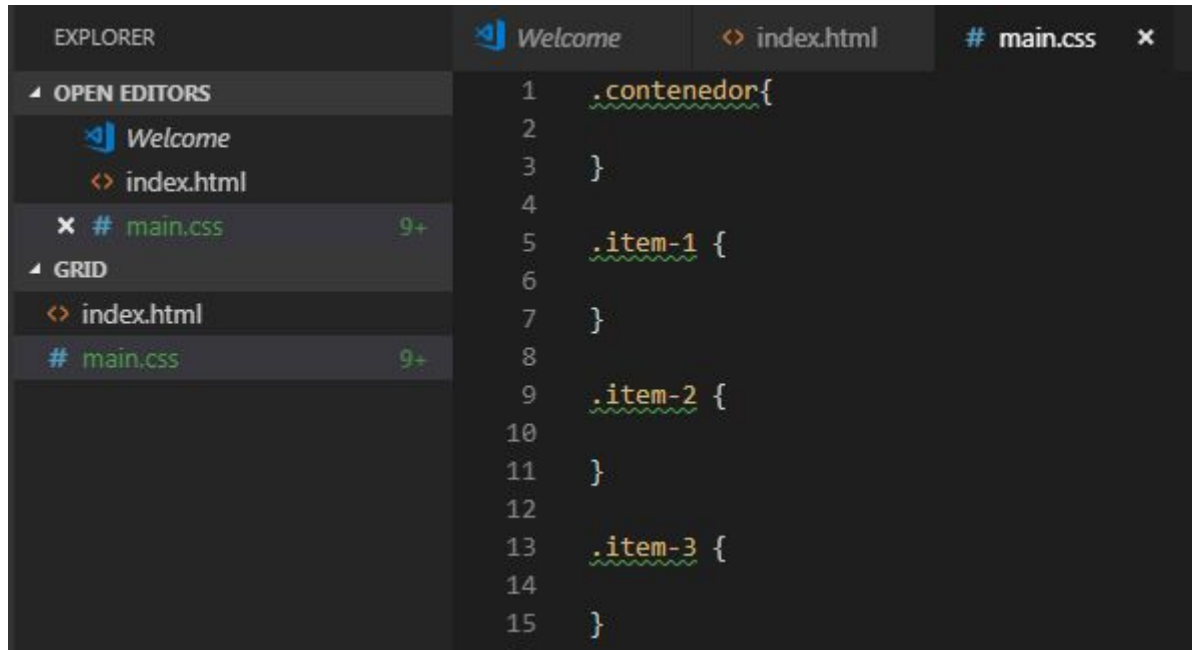
The screenshot shows the Visual Studio Code editor with the following HTML code:

```
12   <main>
13     <div class="contenedor">
14       | div.item-$*9
15     </div>
16   </main>
17 </body>
18 </html>
```

An Emmet abbreviation suggestion box is open, showing the text: `<div class="item-1">|</div>` followed by 8 more lines of similar code for items 2 through 9.

CSS Grid

Ahora Crearemos el documento css.

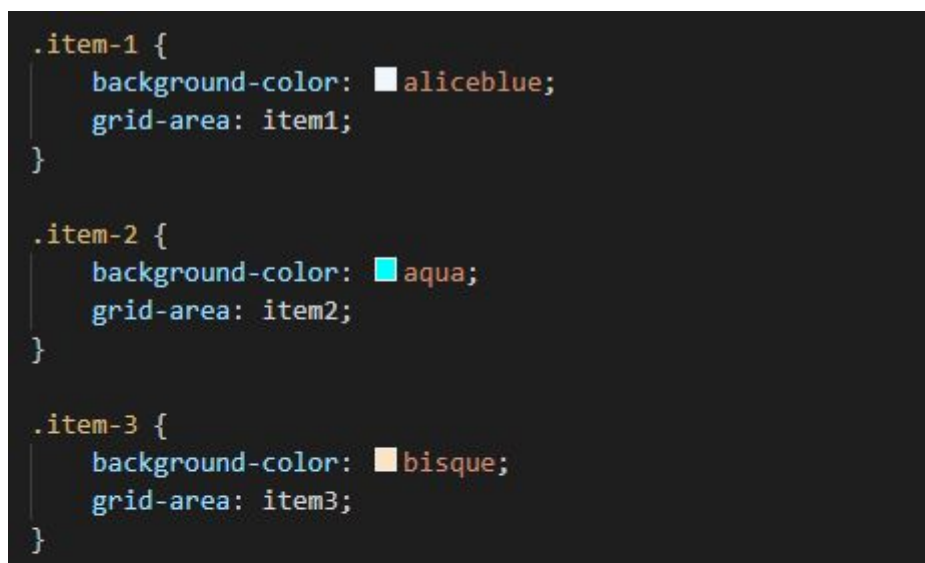


creando las clases que anteriormente indicamos en el html.

partiremos por los ítem.

les daremos un color de fondo para diferenciarlos. con **Background-Color:**

y lo más importante definir el nombre del área que ocupará el elemento con **grid-area:**



Luego en la clase contenedor:

indicaremos:

max-width : 100% para que el contenedor ocupe todo el ancho de la página.

display: grid : le da el formato de grilla para que las propiedades siguientes funcionen.

grid-template-rows :cantidad de filas y el alto, en este caso son 3 y ocupan 20 rem.

grid-template-columns :cantidad de cols y el ancho, 3 y ocupan 20 rem.

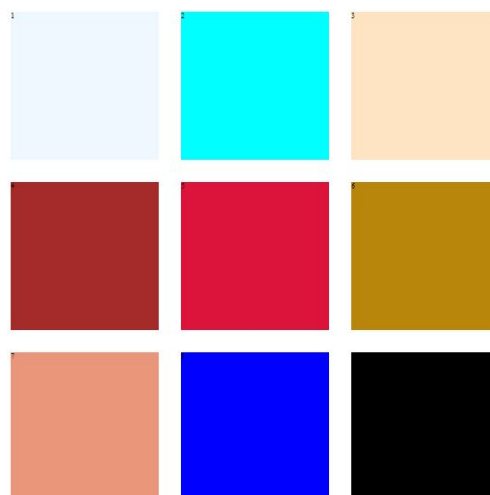
grid-row-gap : es el espacio entre filas:

grid-column-gap : es el espacio entre columnas:

grid-template-areas : definimos el espacio según el nombre de las áreas que definimos en cada elemento (grid-area) en forma filas, cada fila va en entre "" y separadas por espacio, en este caso se ocupó salto de línea (enter) para hacerla más gráfica en comparación al comportamiento que debiese tener en el grid.

```
.contenedor {
  max-width: 100%;
  display: grid;
  grid-template-rows: 20rem 20rem 20rem;
  grid-template-columns: 20rem 20rem 20rem;
  grid-column-gap: 3rem;
  grid-row-gap: 3rem;
  grid-template-areas:
    "item1 item2 item3"
    "item4 item5 item6"
    "item7 item8 item9"
  ;
}
```

Quedando de esta manera:

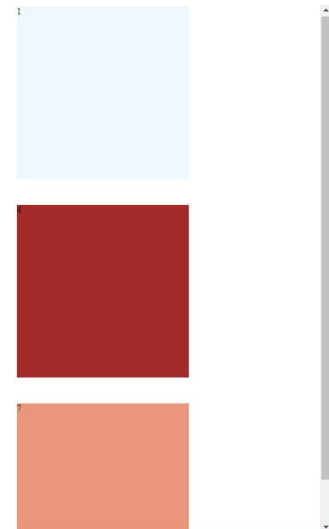


Media Querys

se definen @media y las podemos utilizar para hacer responsiva la página, en este caso se le indica que cuando tenga menos de 600px el comportamiento de las clases o id será el siguiente:

```
@media(max-width:600px){
  .contenedor{
    grid-template-rows: 20rem 20rem 20rem;
    grid-template-columns: 20rem ;
    grid-template-areas:
      "item1 "
      "item4 "
      "item7 "
      ;
  }
  .item-2 , .item-3, .item-5, .item-6, .item-8, .item-9{
    display: none;
  }
}
```

solo se mostrará el ítem 1,4,7 en una sola fila y ocultaremos los demás ítem con display none.



se pueden agregar múltiples media querys, pero siempre debe ir el más bajo al final del css, ya que de lo contrario se sobrescribieran.

```
@media(max-width:600px){
  .contenedor{
    grid-template-rows: 20rem 20rem 20rem;
    grid-template-columns: 20rem ;
    grid-template-areas:
      "item1 "
      "item4 "
      "item7 "
      ;
  }
  .item-2 , .item-3, .item-5, .item-6, .item-8, .item-9{
    display: none;
  }
}
@media(max-width:400px){
  .contenedor{
    grid-template-rows: 20rem 20rem 20rem;
    grid-template-columns: 20rem ;
    grid-template-areas:
      "item1 "
      "item4 "
      "item7 "
      ;
  }
  .item-2 , .item-3, .item-5, .item-6, .item-8, .item-9{
    display: none;
  }
}
```