

00 HTML Demo

1. Crear un nuevo fichero HTML

- Desde dentro de **VS Code** creamos un nuevo fichero HTML *./index.html*
- Una de las ventajas que ofrece **VS Code** es que incorpora plantillas para generar directamente estructuras de fichero complejas, dentro del fichero *./index.html* empezamos a escribir *html*, vemos como aparece una ventana emergente donde podemos seleccionar distintos tipos de fichero, en nuestro caso seleccionamos **html:5** y presionamos intro.

./index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <h2>Hello html!</h2>
</body>
</html>
```

2. Abrir el fichero HTML

- Desde el explorador de ficheros buscamos nuestro recién creado *index.html*.
- Botón derecho del ratón. Abrir con. Aquí aparecerán una serie de programas seleccionamos *Chrome*
- Nuestra página html se abrirá en el navegador.

02 Reglas CSS Demo

Previo

- Partimos desde la carpeta en la que hemos creado nuestro fichero *index.html* anterior.

1. Crear nuevos elementos en el fichero HTML

- Vamos a modificar nuestro HTML para incluir nuevos elementos

./index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <h2>Hello html!</h2>
+   <div>Soy un contenedor genérico</div>
+   <p>Simplemente algo de texto</p>
+   <div>
+     <p>Soy un parrafo dentro de un contenedor genérico</p>
+   </div>
</body>
</html>
```

2. Crear una hoja de estilos

- Vamos a añadir una hoja de estilos a nuestra página.
- Desde la raíz de la carpeta creamos el fichero *./site.css*

3. Referenciar la hoja de estilos dentro de index.html

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
+  <link rel="stylesheet" href="site.css">
</head>
<body>
  <h2>Hello html!</h2>
  <div>Soy un contenedor genérico</div>
  <p>Simplemente algo de texto</p>
  <div>
    <p>Soy un parrafo dentro de un contenedor genérico</p>
  </div>
</body>
</html>

```

4. Editar site.css

./site.css

```

body {
  font-family: Arial, Helvetica, sans-serif;
}

h2 {
  color: blue;
}

.container {
  background-color: red;
  color: white;
}

#simpletext {
  font-size: 32px;
  background-color: pink;
}

div {
  width: 300px;
  background-color: yellow;
}

div p {
  background-color: blue;
  color: white;
}

```

- Tenemos selectores simples *body, h2, div*
- Selectores de clase *.container*
- Selectores por id *#simpletext*
- Combinators descendents *div p*

5. Editar html

- Existen ciertos selectores, los de clase e id, dentro de **site.css** que para que se apliquen deben encontrar ciertos atributos dentro de los elemento HTML.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <link rel="stylesheet" href="site.css">
</head>
<body>

```

```
- <h2>Hello html!</h2>
+ <div>Soy un contenedor genérico</div>
- <div class="container">Soy un contenedor genérico</div>
+ <p>Simplemente algo de texto</p>
- <p id="simpletext">Simplemente algo de texto</p>
+ <div>
  <p>Soy un parrafo dentro de un contenedor genérico</p>
</div>
</body>
</html>
```

- Comentar resultados.

02 Block vs Inline

1. Creamos el site.css

```
./site.css
```

```
body {
    font-family: Arial, Helvetica, sans-serif;
}
```

```
.box {
  height: 100px;
  border: solid 1px black;
  background-color: pink;
  color: white;
  text-align: center;
}
```

```
.box .box-item {
  background-color: green;
  border: solid 1px white;
  color: white;
}
```

2. Creamos el siguiente fichero index.html

./inde.html

[illegible]

</html>

03 Shopping cart HTML

1. Creamos el fichero site.css, de momento este fichero lo dejamos vacío, lo utilizaremos en nuestro siguiente ejemplo.

2. Creamos un nuevo fichero index.html

./index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="site.css">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

3. Vamos a crear el HTML para generar nuestras distintas entradas dentro de nuestro carrito de la compra

- Para arrancar vamos a generar una serie de contenedores genéricos utilizando **div**. Esto nos permitirá crear distintas secciones dentro de nuestra página.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="site.css">
  <title>Document</title>
</head>
<body>
  <!-- diff -->
  <div class="shopping-container">
    <div class="shopping-header"></div>
    <div class="shopping-body"></div>
    <div class="shopping-footer"></div>
  </div>
  <!-- diff -->
</body>
</html>
```

- Ahora mismo las clases añadidas no tienen ningún efecto. No son reglas declaradas dentro de nuestro fichero css.
- Su único cometido de momento es hacer más legible la estructura del documento que queremos crear.

4. Dentro del contenedor shopping-body incluiremos los distintos items de nuestro carrito de la compra. Lo que queremos es que cada uno de esos items ocupe el ancho completo de la página. Además los elementos que componen el item tales como, si el item esta seleccionado o no, cantidad, precio etc, queremos que aparezcan en una sola línea.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="site.css">
  <title>Document</title>
</head>
```

```

<body>
  <div class="shopping-container">
    <div class="shopping-header">
      <!-- diff -->
      <h2>SHOPPING CART</h2>
      <!-- diff -->
    </div>
    <!-- diff -->
    <div class="shopping-body">
      <div class="shopping-item">
        <input type="checkbox">
        <img src="" alt="" srcset="">
        <span>Name</span>
        <span>Description</span>
        <button>+</button>
        <span>2</span>
        <button>-</button>
        <span>&#36;</span>
        <span>300</span>
      </div>
      <div class="shopping-item"></div>
      <div class="shopping-item"></div>
      <div class="shopping-item"></div>
    </div>
    <!-- diff -->
    <div class="shopping-footer">
      <!-- diff -->
      <h2>TOTAL</h2>
      <span>&#36;</span>
      <span>1200</span>
      <!-- diff -->
    </div>
  </div>
</body>
</html>

```

- Notar que hemos utilizado **div** para cada uno de las distintas entradas dentro del cuerpo del carrito de la compra.
- Los elementos html dentro de la propia entrada son todos en línea (span, button, img) por lo que nuestro objetivo estaría cubierto.

04 Shopping cart Style

1. Partiendo del ejemplo anterior, vamos a empezar cambiando el tipo de fuente de nuestro documento. Los formatos de fuente son muy importnates. Un buen formato de fuente puede mejorar mucho la apariencia de nuestra página web.

./site.css

```

body {
  font-family: Arial, Helvetica, sans-serif;
}

```

2. Para obtener la apariencia que buscamos vamos a modificar nuestro markup. Esto no es estrictamente ncesario, pero va a hacer que nuestra tarea sea más sencilla.

./index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="site.css">
  <title>Document</title>
</head>
<body>
  <div class="shopping-container">
    <div class="shopping-header"></div>

```

```

<div class="shopping-body">
  <div class="shopping-item">
    <!-- diff -->
    <div class="selection">
      <input id="item1" type="checkbox">
      <label for="item1"></label>
    </div>
    <img src="" alt="" srcset="">
    <div class="item-entry">
      <span class="item-name">Name</span>
      <span class="item-description">Description</span>
    </div>
    <div>
      <button class="quantity">+</button>
      <span class="quantity-amount">2</span>
      <button class="quantity">-</button>
    </div>
    <div class="value">
      <span>&#36;</span>
      <span>300</span>
    </div>
    <!-- diff -->
  </div>
  <div class="shopping-item"></div>
  <div class="shopping-item"></div>
  <div class="shopping-item"></div>
</div>
<div class="shopping-footer"></div>
</div>
</body>
</html>

```

- Notar que estamos utilizando de nuevo **div** para generar estructuras más complejas dentro de nuestro *HTML*. Estas agrupaciones nos otorgan sentido dentro de la estructura y por tanto un mejor entendimiento del markup.

3. Sin duda una de las tareas más complejas que existe dentro del estilado de los elementos HTML es su posicionamiento dentro de una página. Una característica relativamente nueva dentro de CSS, nos va a ayudar en tan ardua tarea.

./site.css

```

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* diff */
.shopping-header > h2 {
  margin: auto;
  padding: 20px;
  text-align: center;
}

.shopping-item {
  padding: 32px;
  border-radius: 3px;
  border: 1px solid #dcdcdc;
  display: flex;
  justify-content: space-between;
  margin-bottom: 8px;
}
/* diff */

```

- **.shopping-header > h2**, en los elementos en bloque cuando utilizamos *text-align:center* para centrar horizontalmente el contenido. Usando *padding* generamos espacio en todas las direcciones.
- **.shopping-item**, representa cada una de las entradas de nuestro carrito. Mediante *display:flex* y *justify-content: space-between*, conseguimos que todos los elementos dentro del contenedor se alineen dejando el mismo espacio entre ellos. Con *border* y *border-radius*, conseguimos estilar los lados del contenedor.

4. Manejar imagenes dentro de nuestros documentos puede ser difícil. Pensar que una imagen vendrá con una ancho y alto dados. Para solventar este problema, creamos la clase responsive.

```
./site.css

body {
  font-family: Arial, Helvetica, sans-serif;
}

.shopping-header > h2 {
  margin: auto;
  padding: 20px;
  text-align: center;
}

.shopping-item {
  padding: 32px;
  border-radius: 3px;
  border: 1px solid #dcdcdc;
  display: flex;
  justify-content: space-between;
  margin-bottom: 8px;
}

/* diff */
.responsive {
  max-width: 50%;
  width: 100px;
  height: auto;
}
/* diff */
```

- **.responsive**, le decimos que como máximo tenga un *50%* de su ancho. Al fijar el ancho con *100px*, estamos estableciendo el punto a partir del cual la imagen comenzará a reescalar.

5. Vamos a modificar el texto relativo al nombre y la descripción.

```
./site.css

.item-entry {
  width: 300px;
}

.item-name {
  text-transform: uppercase;
  color: #666666;
  font-size: 20px;
  font-weight: bold;
}

.item-description {
  text-transform: lowercase;
  font-size: 16px;
  display: block;
}
```

- **.item-entry**, les damos a todas las entradas el mismo ancho, con esto conseguimos que todas las entradas queden alineadas de manera vertical.
- **.item-name** utilizamos *text-transform: uppercase* para que el texto aparezca en mayuscula, jugamos con el color de la fuente con *color*, su tamaño *font-size*, su grosor *font-weight*.
- **.item-description**, como estamos utilizando un elemento en linea *span*, y queremos que pase a una nueva línea, lo transformamos en un elemento en bloque.

6. Ahora es el momento de crear el estilado de los botones de incremento y decremento del item dentro de nuestro carrito.

```
./site.css

body {
```

```

    font-family: Arial, Helvetica, sans-serif;
}

.shopping-header > h2 {
    margin: auto;
    padding: 20px;
    text-align: center;
}

.shopping-item {
    padding: 32px;
    border-radius: 3px;
    border: 1px solid #dcdcdc;
    display: flex;
    justify-content: space-between;
    margin-bottom: 8px;
}

.responsive {
    max-width: 50%;
    width: 100px;
    height: auto;
}

.item-entry {
    width: 300px;
}

.item-name {
    text-transform: uppercase;
    color: #666666;
    font-size: 20px;
    font-weight: bold;
}

.item-description {
    text-transform: lowercase;
    font-size: 16px;
    display: block;
}

/* diff */
.quantity {
    box-shadow: inset 0px 1px 0px 0px #ffffff;
    background-color: transparent;
    border-radius: 3px;
    border: 1px solid #dcdcdc;
    cursor: pointer;
    color: #666666;
    font-size: 16px;
    font-weight: bold;
    padding: 6px 12px;
    text-shadow: 0px 1px 0px #ffffff;
}

.quantity:active {
    position: relative;
    top: 1px;
}

/* diff */

```

- **.quantity**, el estilado del boton con *box-shadow* conseguimos dar cierta profundidad a nuestro boton, y con *text-shadow* damos sensación de que el texto del botón se resalta.
- **.quantity:active**, esta pseudo clase se activa cuando el botón está activo, esto es cuando el usuario hace click sobre el mismo, al modificar su posicionamiento en un pixel, generamos una animación que simula que se presiona.

7. Lo último que nos queda es modificar la apariencia del texto del resto de elemntos de nuestra página.

./site.css


```

body {
    font-family: Arial, Helvetica, sans-serif;
}

.shopping-header > h2 {
    margin: auto;
    padding: 20px;
    text-align: center;
}

.shopping-item {
    padding: 32px;
    border-radius: 3px;
    border: 1px solid #dcdcdc;
    display: flex;
    justify-content: space-between;
    margin-bottom: 8px;
}

.responsive {
    max-width: 50%;
    width: 100px;
    height: auto;
}

.item-entry {
    width: 300px;
}

.item-name {
    text-transform: uppercase;
    color: #666666;
    font-size: 20px;
    font-weight: bold;
}

.item-description {
    text-transform: lowercase;
    font-size: 16px;
    display: block;
}

/* diff */
.quantity {
    box-shadow: inset 0px 1px 0px 0px #ffffff;
    background-color: transparent;
    border-radius: 3px;
    border: 1px solid #dcdcdc;
    cursor: pointer;
    color: #666666;
    font-size: 16px;
    font-weight: bold;
    padding: 6px 12px;
    text-shadow: 0px 1px 0px #ffffff;
}

.quantity:active {
    position: relative;
    top: 1px;
}

/* diff */
.quantity-amount {
    color: #666666;
    font-size: 16px;
    font-weight: bold;
    margin-left: 16px;
    margin-right: 16px;
}

.value {

```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
font-weight: 500;
font-size: 24px;
min-width: 100px;
}

.shopping-footer {
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.shopping-footer > span {
font-weight: 700;
font-size: 30px;
}
/* diff */
```

- Aquí simplemente estamos retocando el estilado de la fuente, y ciertos anchos para conseguir que neustros elementos queden alineados.