



# Tablas en HTML y despliegue de datos JSON

# Etiqueta table

- Las tablas nos permiten organizar y mostrar datos en nuestro documento html, en forma de filas y columnas.
- Estructura básica:

```
<table>
  <thead>
    <!-- Cabezal de la tabla -->
  </thead>
  <tbody>
    <!-- Cuerpo de la tabla -->
  </tbody>
</table>
```

- Con <thead> agrupamos el contenido del encabezado
- Con <tbody> agrupamos el contenido del cuerpo



# Etiqueta tr, td y th

- Con `<tr></tr>` definimos una fila en nuestra tabla.
- La fila estará compuesta por celdas
- Con `<td></td>` definimos una celda del cuerpo
- Con `<th>` definimos una celda del encabezado

# Ejemplo

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Edad</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Juan</td>
      <td>Perez</td>
      <td>27</td>
    </tr>
    <tr>
      <td>Ana</td>
      <td>Rosa</td>
      <td>29</td>
    </tr>
  </tbody>
</table>
```

Nombre	Apellido	Edad
Juan	Perez	27
Ana	Rosa	29

# Aplicando estilos a la tabla con CSS

- Podemos aplicarle un borde a la tabla y sus elementos:

```
table, th, td {  
  border: 1px solid black;  
}
```

- Con esto le estamos diciendo que tendrá un borde de 1 px de tamaño, sólido, y de color negro.

Nombre	Apellido	Edad
Juan	Perez	27
Ana	Rosa	29

# Bordes con css

- Podemos contraer o “colapsar” los bordes de la tabla en uno solo, con la siguiente propiedad:
  - `border-collapse: collapse;`
- El código css queda de la siguiente forma:

```
table, th, td {  
  border: 1px solid black;  
  border-collapse: collapse;  
}
```

# Ajustar el ancho de la tabla:

- Podemos hacer que la tabla se ajuste al tamaño de la pantalla:
  - `width: 100%;`
- Con la siguiente propiedad, podemos centrar el texto del contenido de la tabla:
  - `text-align: center;`
- El código quedara así:

```
table {  
  width: 100%;  
  text-align: center;  
}
```

# Ancho, alto y márgenes

- Podríamos aplicar la propiedad `height` a los `th` (celdas del encabezado), para que tengan un espaciado mayor.

```
th {  
    /* Puede ponerse otro valor != 50 */  
    height: 50px;  
}
```

- Con `margin` le podemos aplicar un margen

```
table {  
    width: 100%;  
    margin-top: 30px;  
}
```

- Con `padding` le aplicamos un espacio entre el borde y el contenido de la tabla

```
th, td {  
    padding: 10px;  
}
```



# Divisores horizontales:

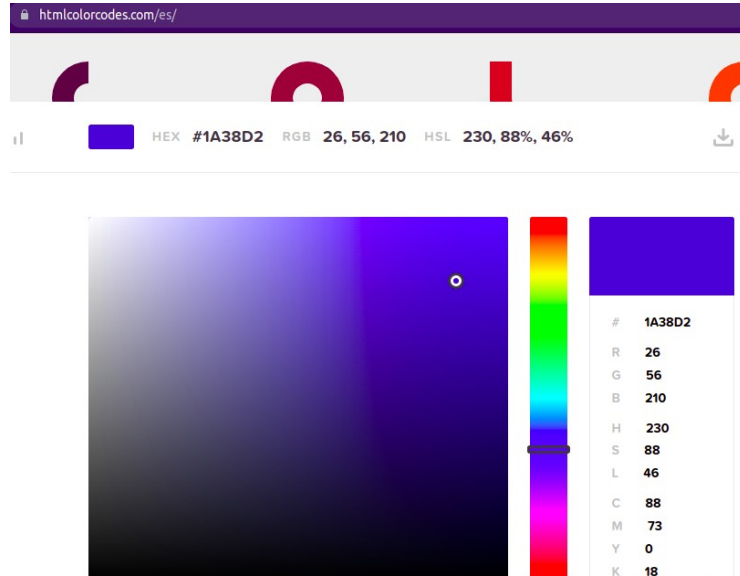
- Podemos agregar un divisor horizontal con la propiedad **border-bottom** (aplicada a th y td)

```
th, td {  
  padding: 10px;  
  border-bottom: 1px solid #rgb(197, 187, 187);  
}
```

- Le indicamos el tamaño, el tipo (sólido) y el color
- El color puede ser un rgb() o un hexadecimal #

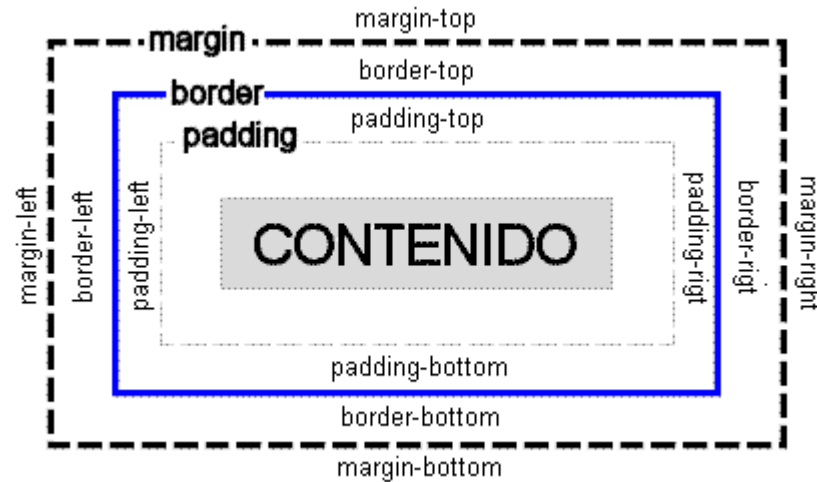
# Colores

- Visual Studio Code nos permite seleccionar un color y generar un rgb
- En caso de usar otro editor, podemos generar un hexadecimal o un rgb en alguna página de internet:



# Margin al div

- Pondremos la tabla dentro de un contenedor (div) y al div le aplicaremos un margen
- 



# Margin

```
<body>
<div class="miTabla">
  <h2>Mis datos:</h2>
  <table>
    <thead>
      <tr>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Edad</th>
```

```
.miTabla {
  margin: 35px;
}
```

# Color de fondo al thead

- Con estas propiedades aplicamos color de fondo y color de texto:

```
th {  
  background-color: #04AA6D;  
  color: white;  
}
```

- La tabla va quedando así:

**Mis datos:**

Nombre	Apellido	Edad
Juan	Perez	27
Ana	Rosa	29

# Fuente

- Con `font-family` indicamos la tipografía

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

- Con `*` indicamos que esta propiedad vale para todos los elementos
- El intérprete buscará las fuentes y aplicará la primera que encuentre en los recursos del sistema.
- La página va quedando así:

**Mis datos:**

Nombre	Apellido	Edad
Juan	Perez	27
Ana	Rosa	29

# Pseudo clases en css

- Una pseudo-clase es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.
- Por ejemplo, `:hover` aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.

```
tr:hover { background-color: #bfe9b5; }
```

- Con esto indicamos que cuando se pase el cursor por el tr, el color de fondo cambiará

# Desplegar un JSON en una tabla

- Hay varias formas de hacer esto
- Una de ellas es ir creando la tabla desde JavaScript, y agregarla como child al div
- Otra forma, es usar la propiedad innerHTML
- Lo primero que debemos hacer es tener el JSON
- En este caso, vamos a obtenerlo con una petición, usando fetch.
- La información se deberá mostrar al cargar la página, entonces vamos a usar el evento Load



# Hagamos un ejemplo...

- Usemos la API publica JSON-Placeholder

```
addEventListener("load", async () => {
  const data = await getData();
  showData(data);
})

const getData = async () => {
  const data = await fetch("https://jsonplaceholder.typicode.com/users");
  const json = await data.json();
  return json;
}

const showData = (data) => {
  // Show data in table
  const tbody = document.getElementById("tbody");

  data.forEach(user => {
    const tr = document.createElement("tr");
    const tdId = document.createElement("td")
    const tdName = document.createElement("td")
    const tdUser = document.createElement("td")
    const tdCity = document.createElement("td")
    const tdWebsite = document.createElement("td")
    tdId.appendChild(document.createTextNode(user.id));
    tdName.appendChild(document.createTextNode(user.name))
    tdUser.appendChild(document.createTextNode(user.username))
    tdCity.appendChild(document.createTextNode(user.address.city))
    tdWebsite.appendChild(document.createTextNode(user.website))

    tr.appendChild(tdId)
    tr.appendChild(tdName)
```

# Fuente

- Documentación de Tablas:

<https://devdocs.io/html/element/table>

- Estilos de tablas con css:

<https://www.mclibre.org/consultar/amaya/css/css-tablas-modos-bordes.html>

-