Introducción a HTML

Tabla de contenidos

6. El tag `<head>` tag
7. Classes y IDs
8. El tag `<span>` tag
9. El tag `<div>` tag
10. CSS
    - Background color
    - Floating
    - Positioning
    - Margins and Padding
11. El tag `<link>`.

Estructura de los documentos HTML

- Para esta clase usaremos HTML5 y CSS3
- Una página HTML se compone de las siguiente funciones:

```
<!doctype html>
<html>
        <head>
                <title>
                        My Website
                </title>
        </head>
        <body>
                Hello, World!
        </body>
</html>
```

- Puedes abrilo en la carpeta **1 - Structure** fichero `part1.html`.
- Los ficherlos .html se pueden abrir en el navegador

- HTML es un lenguaje de marcado (*HyperText Markup Language*), el cual consiste en etiquetas o tags.
- Estos tags se escriben entre `<` y `>` como `<sometag>`
- Todas las tags se tienen que cerrar con otra igual que contiene `/`, después del primer `<`. Ej: `</sometag>`.
- Por ejemplo: `<html>` es un tag que cierra con `</html>`, de la misma manera que `<head>` y `</head>` o `<body>` y `</body>`.
- El tag de apertura y cierre juntos son un elemento.
- El contenido está entre los tags.

- Los tag organizan la página. Existen muchos los cuales puedes ver en:
  - HTML Dog Tag List
  - W3Schools Tag List
  - Quackit HTML Tag List

```
<!doctype html>
<html>
        <head>
                <title>
                        My Website
                </title>
        </head>
        <body>
                Hello, World!
        </body>
</html>
```

- Los elementos están anidados
- Como por ejemplo en el caso del elemento `<head>`, contiene el elemento `<title>`.
- Típicamente indentamos los elementos como en python.

- Marcamos el tipo de documento con `<!doctype html>`. Todos los documento HTML tienen que tener este tag.
- Toda la página web tiene que estar dentro del tag `<html>` y la última línea del documento es `</html>`.
- Dentro del tag `<html>` siempre tenemos dos elementos: `<head>` y `<body>`.
- El contenido en `<head></head>` incluye la informacion que necesita el navegador pero el usuario no necesita conocer. Por ejemplo, el título que se muestra en el navegador, este se define como `<title>`
- El contenido en `<body></body>` es la parte visible de la página web.

Comentarios

Los comentarios se escriben de la siguiente manera:

```
<!-- This is an HTML comment! -->
```

- Abrimos con `<!--` y cerramos con `-->`.
- Puedes añadir comentarios donde se quiera.
- Pueden ser de una línea o varias.

Ejercicio

- Remplaza "Hello, World!" con un texto de tu gusto y ábrelo en el navegador.

Estructurando el texto

- Para ello tenemos los tags: `<h1>`, `<p>`, `<ul>`, y `<li>`.
- Puedes ver un ejemplo en `part2.html` en la carpeta **1 - Structure**.
- `<h1>` es para añadir un título.
- Para un título menor podemos usar `<h2>`, podemos bajar hasta `<h6>`.
- `<p>` es para generar párrafos, es decir un bloque de texto.
- `<ul>` para listas de elementos, donde cada elemento se define con el tag `<li>`.

Links

- Los enlaces se generan con el tag `<a>`.
- Ejemplo:

```
<p>Enlace a <a href="http://www.elpais.com/">un periodico.</a></p>
```

Ejercicio

- En la carpeta **2 - Tags** fichero `page1.html`, pega el enlace.
- Prueba a abrir `page1.html` y hacer click.

- `<a>` es el tag del link
- Con `href=` definimos el link. Es un atributo del tag.
- En html podemos definir atributos como:

```
<tag attribute="value1" attribute2="value2">Content of tag</tag>`
```

- Dentro del tag `<a>` escribimos lo que queremos que aparezca como link.

Links a páginas internas

- Cuando queremos generar un link a una página del mismo directorio no ponemos la url entera.
- Ejemplo:

```
<a href="page2.html">Click here to go back to Page 2.</a>
```

Ejercicio

- Pega:

```
<a href="page2.html">Click here to go back to Page 2.</a>
```

el fichero `page1.html` y comprueba qué pasa.

Otros tags

- Imágenes
- Line breaks
- Tablas

Imágenes

- Con el tag `<img>` podemos añadir imágenes.
- Añade a `page1.html`:

```
<img src="https://i.imgur.com/B9q0A.gif" alt="I could have danced all night"/>
```

- Si abres la página en el navegador verás la imagen.
- No necesitamos cerrar el tag si ponemos `/` al final
- El atributo `src` indica el path o url donde se encuetra la imagen.
- El atributo `alt` es el texto alternativo de la imagen.

Line breaks

- Si queremos añadir un salto de línea tenemos dos tags:
  - Salto de línea simple: `<br>`
  - Generando una línea `<hr>`

Tablas

- Exiten varios tags para generar tablas, los principales son: `<table>`, `<tr>`, `<th>`, and `<td>`.
- Ejemplo en la carpeta **2 - Tags** fichero `tables.html`.
- Con `<table>` creamos la tabla. Dentro tendremos todos los elementos de la tabla.
- Con el atributo `border` podemos definir el ancho de los bordes de la tabla.

```
<table border="1">
</table>
```

- Con `<tr>` añadimos filas a la tabla (*table row*).

```
<table border="1">
        <tr>
        </tr>
        <tr>
        </tr>
        <tr>
        </tr>
</table>
```

- Para los datos de la tabla podemos usar:
  - `<th>` (*table header*) para el encabezado de la tabla.
  - `<td>` (*table data*) para los datos de la tabla.

```
<table border="1">
        <tr>
                <th></th>
                <th></th>
                <th></th>
                <th></th>
        </tr>
        <tr>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
        </tr>
        <tr>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
        </tr>
</table>
```

- Añadimos los datos:

```
<table border="1">
        <tr>
                <th>Item</th>
                <th>Quantity</th>
                <th>Rate</th>
                <th>Cost</th>
        </tr>
        <tr>
                <td>Candy</td>
                <td>10</td>
                <td>$.50</td>
                <td>$5.00</td>
        </tr>
        <tr>
                <td>Toothpaste</td>
                <td>2</td>
                <td>$3.00</td>
                <td>$6.00</td>
        </tr>
</table>
```

Estilo

- Hasta ahora el diseño es muy simple.
- Vamos a ver como añadir color, bordes y estilo de texto.
- Veremos posteriormente una forma más adecuada de hacerlo.

Colores

- Trabajamos en la carpeta **3 - Styles**.
- Modificaremos el fichero `style1.html`.
- Añadir:

```
<p style="color: red">This text is in red</p>
```

- Abrir en el navegador.

- El atributo `style` nos permite cambiar los colores, tamaños, bordes, etc.
- Podemos definir colores directamente con el nombre del color como: `blue`, `yellow`, `cyan`, `magenta`, etc
- Podemos usar los colores en RGB o HEX.
- RGB de 0 a 255 por cada color.
- Lo definimos como: `style="color: rgb(255,0,0)"`.
- HEX consiste en 6 dígitos hexadecimal, empezando por el signo `#`. Ejemplo: `style="color: #FF0000"`, FF para rojo 00 para el azul y 00 para el verde.
- Puedes alguna de ests herramientas:
  - Color Picker
  - HTML color codes and names
  - HTML Color Codes
  - HTML Color Picker

Ejercicio:

- Añade colores a varios tags del fichero `style1.html`.
  - `<h1>` con color `#005DFC`
  - `<h3>` con color `rgb(242,127,56)`
  - `<p>` con color `lightblue`.

Width y Height

- Para cambiar el tamaño de algun elemento tenemos opciones usando el atributo `style` o los atributos `width` y `height`.

```
<img src="https://i.imgur.com/4ihC2Yb.gif" />
```

- Si queremos que la imagen sea de un tamaño de 600x800px.

```
<img src="https://i.imgur.com/4ihC2Yb.gif" width="600" height="800" />
```

- Probamos en el fichero `style1.html`
- Podemos definir solamente uno de estos atributos para mantener el ratio de aspecto:

```
<img src="https://i.imgur.com/4ihC2Yb.gif" width="600" />
```

- Podemos hacer lo mismo con el atributo style:

```
<img src="https://i.imgur.com/4ihC2Yb.gif" style="width: 600px" />
```

- El atributo `style` es para *inline styles*, normalmente usaremos un fichero para definir todos nuestros estilos.
- La sintaxis es `style="property: value"` donde *property* es una propiedad del estilo como puede ser `color`, `width` o `height`.
- Para poner varias propiedades:

```
<img src="https://i.imgur.com/4ihC2Yb.gif" style="width: 600px; height: 800px" />
```

Borders

- Podemos añadir bordes a cualquier elemento:
- Por ejemplo `border="5"` añade un borde de 5 píxeles.
- Ejemplo:

```
<img src="https://i.imgur.com/4ihC2Yb.gif" style="border:5px solid black" />
```

- El atributo `border` tiene 3 partes, la primera es el grosor del borde, el siguiente es el estilo del borde como por ejemplo `solid`, `dotted`, `dashed`, o `double`, y por último el color.

- Puedes ver opciones aquí.
- Diferentes ejemplos:

```
<img src="https://i.imgur.com/4ihC2Yb.gif" style="border:5px dotted #ffcc00" />
<img src="https://i.imgur.com/4ihC2Yb.gif" style="border:10px ridge rgb(77, 145, 99); width: 300px" />
<img src="https://i.imgur.com/4ihC2Yb.gif" style="border:8px outset red" />
<img src="https://i.imgur.com/4ihC2Yb.gif" style="border:3px double #333a21; height: 30px" />
```

Estilos de texto

- Podemos usar otros modificadores de estilo para el texto.
- Ejemplo:

```
<p style="text-align: center; font-weight: bold">This text is center and bold.</p>
```

- Con `text-align` podemos elegir si queremos el texto `center`, `left`, o `right`.
- Con `font-weight` podemos cambiar entre: `normal`, `bold`, `bolder`, `lighter` o números `100`, `200`, `300`, `400`, `500`, `600`, `700`, `800`, `900` (donde 400 es lo mismo que normal y 700 es bold).

- Otro ejemplo:

```
<p style="font-family: Arial; font-style: italic">This text is italic and arial.</p>
```

- `font-family` y `font-style` para cambiar las propiedades de la fuente.
- `font-style` puede ser `normal`, `oblique`, o `italic`.

El Tag `<head>`

- `<head>` lo usamos para información que el usuario no ve.
- Podemos incluir en el los tags:
  - `<title>`
  - `<meta>` para los *metadata*, por ejemplo:

```
<meta name="description" content="The best cooking website in the entire universe.  You're welcome.">
```

- Definir el author:

```
<meta name="author" content="Sexy McGoodlooking">
```

CSS

- Hasta ahora el estilo lo estabamos incluyendo en el atributo `<style>`.
- Podemos incluirlo en el `<head>` o en un fichero aparte.
- Trabajamos sobre el fichero `style2.html` de la carpeta **3 - Styles**.
- Podemos incluir el estilo en el tag `<head>` de la siguiente manera:

```
<style>
        body { }

        h1 { }

        p { }

        ol { }
</style>
```

- Esta sintaxis es la denominada CSS o *Cascading Style Sheets*.
- Dentro del tag `<style>` tenemos código CSS.
- En el ejemplo definimos `body`, `h1`, `p`, y `ol`. Estos son tags de html donde aplicaremos el estilo, es lo que definimos como selectores.
- El código se define de la siguiente manera: `selector { code }`

- Para cada selector tenemos un conjunto de *property*:*value*:

```
<style>
        body {
                font-family: Arial;
        }

        h1 {
                color: red;
                text-align: center;
        }

        p {
                font-weight: bolder;
        }

        img {
                width: 400px;
                border: 5px solid #333333;
        }

        ol {
                color: #333333;
        }
</style>
```

Classes y IDs

- Podemos editar los estilos del mismo tag en varios sitios usando clases.
- Usamos el fichero `style3.html` de la carpeta **3 - Styles**
- Si queremos editar el estilo de cada tag de forma individual usamos el atributo class para identificar cada elemento.
- Para definir el estilo de una clase lo hacemos de la siguiente forma:

```
.nombreclase {

}
```

Ejemplo:

```
p {
        font-family: Arial;
}
.poemtitle {
        font-weight: bolder;
}
.author {
        color: #555555;
}
.poem {
        font-style: italic;
}
```

IDs

- Los id son iguales que las clases pero solo podemos tener un único elemento.
- Ejemplo:

```
<p id="special">This is so special that I want it uniquely styled forever.</p>
```

- Para definir el estilo usamos `#`:

```
#special {

}
```

El tag `<span>`

- El tag `<span>` es invisible a menos que se le aplique un estilo.
- Se usa para cambiar el estilo de *inline-elements*, como por ejemplo una palabra.
- Ejemplo:

```
<p>"My grandmother started walking <span>five miles a day</span> when she was sixty. She's ninety-seven now, and <span>we don't know where the heck she is.</span>" </p>
<p>~ Ellen DeGeneres </p>
```

- Unicamente veremos el contenido diferente si aplicamos un estilo:

```
p span {
        font-style: italic;
}
```

- `p span` es lo que se denomina *nesting* CSS
- Cuando tenemos un espacio entre tags el estilo solo afecta a los elementos que están incluídos en la estructura.
- Puedes tener span definido y cambiar el estilo cuando aparece en una determinada clase:

```
span {
        font-weight: bold;
}
.author span {
        color: #999999;
}
```

El tag `<div>`

- El tag `<div>` es similar a `<span>` pero puede ser aplicable en forma de bloque.
- Es una de las que más se usan. Nos permite cambiar totalmente el estilo y formato de las páginas web.
- Vemos como funciona con un ejemplo (carpeta **5 - Layout** fichero `homepage.html`):

```
<!doctype html>
<html>
        <head>
                <title> My Website </title>
                <style>

                </style>
        </head>
        <body>
                <div class="header"></div>
                <div class="menu"></div>
                <div class="content"></div>
                <div class="footer"></div>
        </body>
</html>
```

- Cada `<divs>` tiene una `class` sobre la que podemos aplicar un estilo.
- Podemos generar un estilo para cada una de las clases:

```
html {

}
body {

}
.header {

}
.menu {

}
.content {

}
.footer {

}
```

- Podemos generar un layout cambiado las dimensiones de cada `<div>`:

```
html {
        height: 100%;
}
body {
        height: 100%;
}
.header {
        width: 100%;
        height: 60px;
}
.menu {
        height: 100%;
        width: 15%;
}
.content {
        height: 200px;
}
.footer {
        height: 60px;
        width: 100%;
}
```

- Con `%` definimos porcentajes del elemento que contiene al elemento sobre el que aplicamos el estilo.
- Cuando en `.menu` definimos `width: 15%;`, el elemento menu ocupara un 15% del elemento que lo contiene en este caso `<body>`.

Background color

- Con el atributo `background-color` podemos cambiar el color del fondo de cada elemento.

- Ejemplo:

```
html {
        height: 100%;
}
body {
        height: 100%;
}
.header {
        background-color: #99B5DD;
        width: 100%;
        height: 60px;
}
.menu {
        background-color: #DE90B1;
        height: 100%;
        width: 15%;
}
.content {
        height: 200px;
}
.footer {
        background-color: #0F215D;
        height: 60px;
        width: 100%;
}
```

Floating

- Con la etiqueta `float`. podemos mover los elementos a la derecha o izquierda permitiendo ordenar nuestro layout.
- Ejemplo: `float: left;` or `float: right;` moverá los elementos lo máximo posible a la derecha o izquierda.
- Algunas veces no funciona dependiendo de donde se aplique.
- Podemos arreglarlo con la propiedad `clear` en los elementos incluídos dentro del modificado; en particular `clear: both;` para desactivar float en ambos sentidos.

```css
html {
        height: 100%;
}
body {
        height: 100%;
}
.header {
        background-color: #99B5DD;
        width: 100%;
        height: 60px;
}
.menu {
        background-color: #DE90B1;
        height: 100%;
        width: 15%;
        float: left;
}
.content {
        height: 200px;
        float: left;
}
.footer {
        background-color: #0F215D;
        height: 60px;
        width: 100%;
        clear: both;
}
```

Positioning

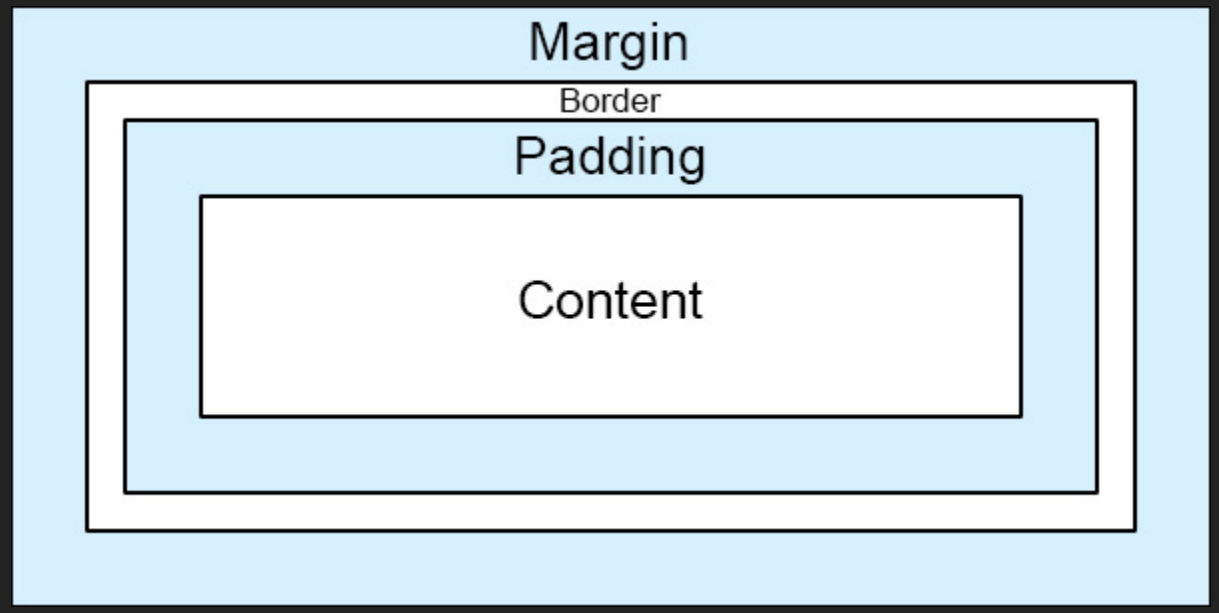
- Con la propiedad `position` podemos definir donde queremos los elementos.
- Por ejemplo la cabezera arriba y el footer abajo.
- Tenemos tres posibles estados: `absolute`, `relative`, y `fixed`
  - `absolute` para poner el elemento en una posición exacta pero se mueve con ella.
  - `relative` el elemento está en una posición relativa a su posición por defecto
  - `fixed` es como `absolute` pero siempre se mantiene en el sitio.

- Podemos definir la posición con los atributos `top`, `bottom`, `left`, y `right`.
- Si queremos que el header arriba siempre pondremos `position: absolute;` y `top: 0px;`
- Si tenemos una imagen y queremos moverla un poco podemos poner: `position: relative;` y `left: 5px`, para moverla 5px a la derecha.

```css
html {
        height: 100%;
}
body {
        height: 100%;
}
.header {
        background-color: #99B5DD;
        position: fixed;
        top: 0px;
        width: 100%;
        height: 60px;
}
.menu {
        background-color: #DE90B1;
        height: 100%;
        width: 15%;
        position: fixed;
        left: 0px;
        top: 60px;
}
.content {
        height: 200px;
        position: absolute;
        top: 60px;
        left: 15%;
}
.footer {
        background-color: #0F215D;
        position: fixed;
        bottom: 0px;
        height: 60px;
        width: 100%;
}
```

Margins and Padding

- Podemos cambiar los márgenes y padding de cada elemento:



- `margin` es el espacio hacia afuera del borde del elemento.
- `padding` es el espacio hacia dentro del borde del elemento.

- El tag `<body>` tiene un margen por defecto. Podemos cambiarlo:

```
body {
        height: 100%;
        margin: 0px;
}
```

- Por ejemplo el `content` está muy cerca del `.menu`.
- Podemos hacerlo de la siguiente manera:

```css
html {
        height: 100%;
}
body {
        height: 100%;
        margin: 0px;
}
.header {
        background-color: #99B5DD;
        position: fixed;
        top: 0px;
        width: 100%;
        height: 60px;
        padding: 10px;
}
.menu {
        background-color: #DE90B1;
        height: 100%;
        width: 15%;
        position: fixed;
        left: 0px;
        top: 60px;
        padding: 10px;
}
.content {
        height: 200px;
        position: absolute;
        top: 60px;
        left: 15%;
}
.footer {
        background-color: #0F215D;
        position: fixed;
        bottom: 0px;
        height: 60px;
        width: 100%;
}
```

Podemos elegir el padding y el margen de cada lado de la siguiente manera:

- `margin: 5px 10px 15px 0px;`
  - top margin es de 5px
  - right margin es de 10px
  - bottom margin es de 15px
  - left margin es de 0px
- `margin: 15px 0px 5px;`
  - top margin es de 15px
  - right y left margins son de 0px
  - bottom margin es de 5px
- `margin: 5px 10px;`
  - top y bottom margins son de 5px
  - right y left margins son de 10px
- `margin: 15px;`
  - todos los margenes son de 15px

```css
html {
        height: 100%;
}
body {
        height: 100%;
        margin: 0px;
}
.header {
        background-color: #99B5DD;
        position: fixed;
        top: 0px;
        width: 100%;
        height: 60px;
        padding: 0px 10px;
}
.menu {
        background-color: #DE90B1;
        height: 100%;
        width: 15%;
        position: fixed;
        left: 0px;
        top: 60px;
        padding: 10px 0px 0px;
}
.content {
        height: 200px;
        position: absolute;
        top: 60px;
        left: 15%;
        padding: 10px;
}
.footer {
        background-color: #0F215D;
        position: fixed;
        bottom: 0px;
        height: 60px;
        width: 100%;
}
```

El tag `<link>`

- Para poder reusar el mismo estilo en diferentes ficheros usamos el tag `<link>`
- Pasamos nuestro estilo a un fichero `.css`

```
<link rel="stylesheet" type="text/css" href="main.css">
```

- Puedes ver un ejemplo en la carpeta **6 - Linking**.

Comentarios CSS

- Podemos comentar los ficheros CSS de la siguiente manera:

```
/* This is a comment in CSS! */
```

- Empiezan con `/*` y terminan con `*/`.

Formularios y Botones

- Formularios y se pueden incluir directamente en html,
- Veremos con generar acciones con dash:

```
<button type="button">Click Me!</button>
```

Ejercicio Final

- Genera un html: `index.html` con los siguientes elementos:
  - Una foto.
  - Texto
  - Tabla
  - Lista
- Usa tags `<div>` con un fichero css para el estilo.
- Crea un repo en github que se llame de la siguiente manera: username.github.io donde `username` es tu nombre de usuario en github. Si lo genermos de esa manera github nos servira para alojar nuestra web. Puedes encontrar más información en: https://pages.github.com/

- Haz un commit con los ficheros del ejercicio.
- Sube tus cambios.
- Entra desde un navegador en: username.github.io donde `username` es tu nombre de usuario en github.