

# Examen

## Lenguajes de marcas

Un **lenguaje de marcado o lenguaje de marcas** es una forma de codificar un documento para que, junto al texto, incorpore marcas con información adicional acerca de la estructura del texto o su presentación.

El lenguaje de marcas es el que especifica cuáles serán estas marcas, donde deben colocarse y el significado detrás de ellas.

## Las marcas

Las marcas son códigos que se incorporan en el documento para definir su formato, estilo de presentación o estructura de los datos. Estas marcas se adjuntan a los datos y suelen representarse entre los símbolos de < y >.

## Clasificación de los lenguajes de marcas

- Los **Lenguajes descriptivos o semánticos** están orientados a describir la estructura lógica de los datos que contiene. **XML** es un ejemplo.
- Los **Lenguajes procedimentales y de presentación** están orientados a especificar cómo debe representarse la información. Lo que se hace es indicar de qué forma se debe realizar la presentación de los datos. **HTML** es un ejemplo de este tipo de lenguajes.

## Organizaciones desarrolladoras.

- **Organización Internacional para la Estandarización (ISO)** el cual es el organismo que tiene como función buscar y definir estandarización de normas a nivel internacional. En 1986 hizo público el SGML (Standard Generalized Markup Language)

- **Consortio World Wide Web (W3C)** tiene la función de tutelar el crecimiento y organización de la web. Fue el encargado de normalizar el lenguaje HTML y de la creación de reglas para que cualquier persona pueda utilizar el lenguaje de marcas según sus necesidades pero manteniendo unos estándares comunes, (XML).

## Etiquetas, elementos y atributos.

- Los **Elementos** representan la estructura mediante la cual se organizará el contenido del documento, o acciones que el navegador interprete al trabajar con el documento. Consta de la *etiqueta de inicio, etiqueta de cabeza* y lo que se encuentre entre ellas dos.

```
<html>
  <head>TEXT0</head>
  <body>
    <p>TEXT0</p>
  </body>
</html>
```

El código de arriba está formado por 4 elementos. El elemento *html* que contiene 2 elementos hijos, *head* y *body*, y el elemento *body* a su vez contiene un elemento hijo *p*.

Toda la línea de código, completa, junto a sus etiquetas, atributos y contenido **es un elemento**. Todo en su conjunto.

- Las **etiquetas o tags** son textos que se encuentran entre los símbolos de menor y mayor que '<' '>' y indica al visualizador del documento de qué tipo es el fragmento de texto que está entre dichas etiquetas.

```
<html>
  <head></head>
  <body></body>
</html>
```

El código de arriba contiene 3 etiquetas, *html*, *head* y *body*.

- Los **atributos** es un par con sintaxis **nombre="valor"** que se encuentra dentro de las etiquetas de inicio de un elemento e **indica sus propiedades asociadas**.  
**Proporcionan información adicional sobre los elementos HTML.**

```
<link rel="stylesheet" href="style.css" />
```

*link* es la etiqueta principal y esta formada por 2 atributos. El atributo *rel* que especifica la relacion entre el actual documento y el documento enlazado. Y el atributo *href* que especifica la localizacion del documento enlazado.

```
<a href="google.es">Ir a google</a>
```

La etiqueta *a* es utilizada para enlazar una pagina con otra. El atributo *href* es para especificar la dirección.

```

```

*img* es la etiqueta para insertar una imagen en un documento html, el atributo *src* es para especificar la ruta de la imagen, el atributo *alt* es para especificar un texto alternativo, y los atributos *width* y *height* son para especificar las medidas de la imagen.

## Elemento de linea y elementos de bloque

- Los **elementos de linea** son aquellos que ocupan un espacio minimo necesario en horizontal.
- Los **elementos de bloque** siempre se inician en una nueva linea y se ocupan todo el espacio de izquierda a derecha hasta la medida de lo posible.

## Estructura de un documento HTML

```
<!DOCTYPE>
<html>
  <head></head>
  <body></body>
</html>
```

- La **etiqueta <!DOCTYPE html>** representa el tipo de documento y ayuda a los navegadores a mostrar correctamente las paginas web.
- El **elemento <html>** es obligatoria y sirve para identificar que el documento está codificado en lenguaje HTML, y para limitar el comienzo y el final.
- El **elemento <head>** es la cabecera. Además sirve como contenedor de elementos que proveen metadatos para el documento. El contenido dentro de este elemento no aparece de forma visual, pero sirve a los navegadores y otros sistemas información útil para procesar el documento.
  - Los metadatos que pueden estar dentro del elemento head pueden brindar información como el título, autor, juego de caracteres utilizado, las hojas de estilo utilizadas.
- El **elemento <body>** será donde se colocará todo aquello que será visual dentro del documento como texto, imágenes, tablas o videos, delimitadas por a su vez por otras etiquetas.

## Comentarios en HTML y CSS

Los comentarios son útiles para los desarrolladores, pues son trozos de texto que no tienen ningún efecto encima de la codificación. En html los comentarios se hacen mediante la siguiente sintaxis:

```
<!-- comentario -->
```

# Hojas de estilo CSS

Una hoja de estilo es un conjunto de reglas que definen la estética final de los documentos html. Las reglas están formadas por un *selector y conjunto de declaraciones*.

## Comentarios en CSS

Los comentarios son útiles para los desarrolladores, pues son trozos de texto que no tienen ningún efecto encima de la codificación. En html los comentarios se hacen mediante la siguiente sintaxis:

```
/* comentario */
```

## Sintaxis de un CSS

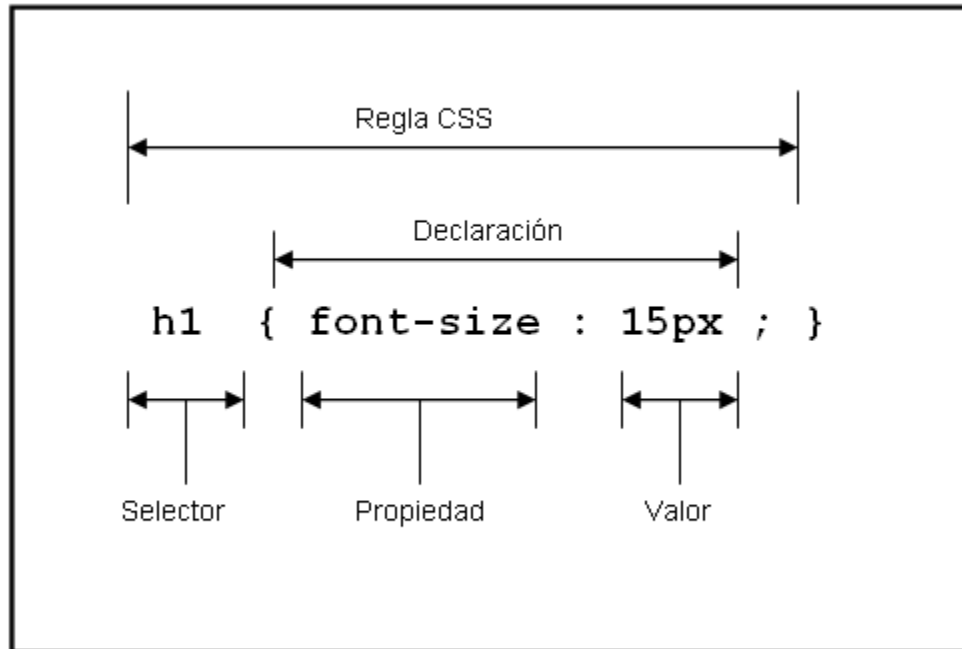
La sintaxis de un css está formado por **selectores y declaraciones**.

- Los **selectores** especifican a qué elemento de la página nos estamos refiriendo.
- Las **declaraciones** consisten en la asignación de un valor a una propiedad.

Haremos uso de los selectores para especificar a qué elemento del documento queremos modificar. Las declaraciones, que es la asignación de un valor a una propiedad, se hacen entre '{' y '}'.

```
selector {  
  propiedad: valor;  
  propiedad: valor;  
}
```

## Conceptos importantes



## Tipos de selectores

En las hojas de estilos los selectores son los encargados de especificar a que elemento se realizara la modificacion declarada en el CSS.

- **Selectores basicos**

- **Selector universal** se utiliza para seleccionar a todos los elementos de la pagina. [ \* ]

```
* {  
  color: red;  
}
```

- **Selectores de etiqueta** hace referencia a todas las etiquetas de la pagina que coincidan con el valor del selector. Si el selector es *h1* la declaracion afectara a todas las etiquetas *h1*.

```
h1 {  
  color: blue;  
  font-size: 1em;  
}
```

- **Selectores de clase** se utiliza cuando queremos darle a ciertos elementos que ya tienen un estilo definido una modificación excepcional. El selector viene inicializado por un '.' Dentro del elemento en el documento HTML se le debe de añadir el atributo `class="css_nombre_clase"` para que así el selector sepa a que elemento dirigirse.

En caso de que queramos a un elemento ya estilizado añadirle una modificación excepcional, podemos añadir la etiqueta delante de la clase.

```
h1, h2 {  
  color: red;  
}  
  
h2.excepcion {  
  color: green;  
}  
  
<h2 class="excepcion"></h2>
```

Si queremos generalizar el estilo para cualquier etiqueta, se trata de no vincular ninguna etiqueta con la clase.

```
<p class="hola">Hola</p>  
  
.hola {  
  color: green;  
}
```


- **Selectores id** son utilizados para referirnos específicamente a un elemento del html, lo hacemos mediante el atributo de etiqueta *id*. Se podría hacer lo

mismo con un selector de clase, pero de esta manera es mas eficiente pues un elemento de HTML unicamente puede tener un id irrepetible.

```
<p id="parrafo"></p>

#parrafo {
  color: blue;
}
```

- **Selectores combinadores**

- **Combinador de hermanos adyacentes** se aplica a todos los elementos que siguen directamente al primer elemento hermano y los dos estan dentro del mismo elemento padre. Su sintaxis es con '+'.  


```
h2 + p {
  propiedad: valor;
}

<body>
  <h2></h2>
  <p></p>
</body>
```

Todos los elementos seguidos de un *h2* que sean un *p* se le aplicara la declaracion de la regla css.

```
h2 + p {
  propiedad: valor;
}

<body>
  <h2></h2>
  <br>
  <p></p>
</body>
```

En este caso la regla css no se aplicara a los *p*, pues no son hermanos continuos, un nuevo elemento interfiere entre ellos dos.



- **Combinador general de hermanos** selecciona hermanos (no es necesario que esten de forma continua), y los dos tienen el mismo padre. Su sintaxis es con "~".

```
p ~ span {  
  propiedad: valor;  
}  
  
<body>  
  <p></p>  
  <br>  
  <span></span>  
</body>
```

Todos los elementos *span* que esten despues de *p* se le aplicara la regla css. Siempre y cuando tengan el mismo elemento padre.

- **Combinador de hijos** aplicara a todos los elementos que son directamente hijos del elemento padre. Su sintaxis es con ">".

```
ul > li {  
  propiedad: valor;  
}
```

Aplicara a todos los elementos *li* que son hijos directos del elemento padre *ul*.

- **Pseudoclases** permiten seleccionar elementos segun un estado especifico, en función de su posición en el documento, ayudan a reducir el exceso de clases y proporcionan un marcado mas flexible y fácil de mantener. Su sintaxis es con ":".
- **Pseudoelementos** se comportan de forma similar a las pseudoclases, mientras que los pseudoelementos suelen hacer referencia a partes determinadas de un elemento. Su sintaxis es con "::".

# Cajas

Los elementos en HTML se organizan en elementos de bloque y elementos en línea. La propiedad **display** permite modificar su comportamiento.

- **"block"** el comportamiento del elemento es un bloque. No acepta ningún elemento a su lado. Expande a todo lo ancho su contenedor.
- **"inline"** el elemento se coloca en línea junto a los demás.
- **"inline-block"** es una mezcla entre elemento de bloque y elemento de línea.
- **"none"** se utiliza para ocultar el elemento.
- **"list-item"** el elemento se comporta como un elemento de lista.

## Posicionamiento de cajas

- **position: static;** los elementos se posicionan en el flujo normal de la página, únicamente tiene en cuenta si es un elemento en línea o de bloque. No son afectados por las propiedades *top*, *bottom*, *left* y *right*.
- **position: relative;** el desplazamiento de las cajas se hace a partir de su posición original, no modifica la posición de los elementos contiguos pero puede solaparse encima de ellos, la caja se mueve con las propiedades *top*, *bottom*, *left* y *right*.
- **position: absolute;** se utiliza para establecer de forma precisa la posición en la que se muestra la caja del elemento. Las propiedades *top*, *bottom*, *right* y *left* empujan la caja hacia el interior de la ventana siempre y cuando el valor sea positivo. Siempre se posiciona a x distancia de los bordes de la ventana.
- **position: fixed;** es una alternativa particular del *absolute*. Los elementos con esta propiedad son inamovibles.
- **position: sticky;** los elementos son posicionados de forma relativa hasta que su bloque contenedor llegue al límite establecido de la página.
- 

## Propiedades para el posicionamiento

- **margin, margin-top, margin-bottom, margin-right, margin-left:** margen exterior que separa la caja de los demás elementos.
- **padding, padding-top, padding-bottom, padding-left, padding-right:** margen interno que se aplica desde el contenido del elemento hacia la caja.
- **top, bottom, left y right** se utiliza para establecer la ubicación de un elemento.
- **propiedad float** el elemento sale del flujo normal de página y aparece posicionado a la izquierda o derecha del interior de su contenedor.
- **propiedad clear**
- **propiedad z-index**