

Lenguajes de marcas y sistemas de gestión de la información



UT03 – CSS

1 – Introducción – Sintaxis

De la semántica a la presentación

Ya hemos estudiado HTML, y hemos visto que:

- Es un lenguaje de marcado, que trabaja en modo texto
- Usa etiquetas (tags) que se añaden al contenido para estructurarlo
- Se centra en la semántica, en el significado y organización del contenido, no en el aspecto

Aunque se centra en el contenido, hemos podido ver que por defecto muestra ciertos elementos con cierto aspecto, por ejemplo:

- Encabezados (h1 – h6) en negrita y con tamaño en función de su nivel.
- Elemento address en cursiva (o itálica).

Vamos a aprender a definir el aspecto de los elementos HTML que no tienen estilo, y a modificar el aspecto por defecto de aquellos que ya lo tengan.

CSS

CSS es un lenguaje, no de programación, pero sigue siendo un lenguaje.

Es un lenguaje declarativo (no imperativo), que define (declara) el aspecto de los elementos de una página web.

También se puede utilizar CSS con otros tipos de documentos, no solo con HTML. También se puede usar con XML, por ejemplo.

Creado el año 1996, su estándar, como el del HTML, lo define y lo mantiene el W3C (World Wide Web consortium).



CSS

Se han publicado, desde 1996, tres especificaciones de CSS:

- CSS 1 – La primera versión oficial. Diciembre de 1996.
- CSS 2 – Mayo de 1998.
- CSS 2.1 – Junio de 2011.
- CSS 3 – No se puede dar una fecha específica, porque cambió la forma de especificar el estándar.

CSS había crecido mucho, y era poco operativo mantener todas las características en una sola especificación.

Así que se partió la especificación en múltiples especificaciones más pequeñas, y se han ido añadiendo otras.

CSS

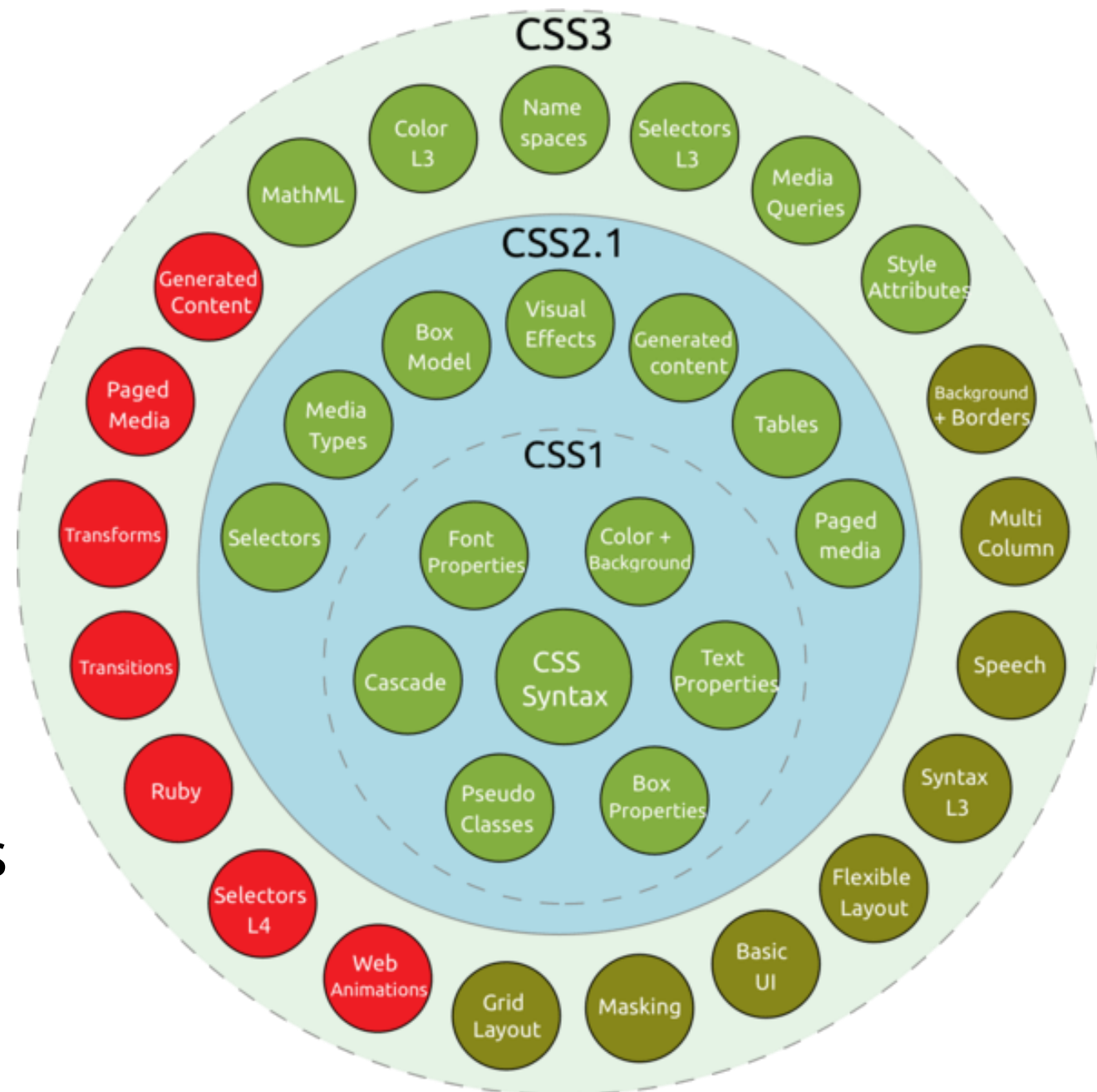
No habrá una especificación CSS 4.

Lo que ocurre es que cada una de las especificaciones "menores", denominadas "módulo", crece de forma independiente.

Por ejemplo, el módulo de color ya tiene propuesto un candidato a estándar para el nivel 4, y hay un borrador del nivel 5.

Para buscar distintos módulos y sus versiones (no solo CSS):

<https://www.w3.org/TR/>



CSS

CSS son las siglas de "Cascading Style Sheets", hojas de estilo en cascada.

Con CSS se crean hojas de estilo (style sheets) para definir la presentación de un documento HTML cuando se visualiza en un navegador.

Antes de la existencia y generalización de CSS, toda la presentación, el aspecto, se debía realizar usando ciertos atributos y elementos de HTML.

La mayoría de estos atributos y elementos están obsoletos o desaconsejados hoy día, y por eso no los hemos visto en la parte de HTML. Por ejemplo, el elemento "<center>", o los atributos "bgcolor", "align", "width" o "height".

CSS

La palabra "cascading" o "en cascada" se refiere a la forma en la que aplican los estilos sobre los elementos.

Los estilos para un elemento se aplican uno detrás de otro, de forma que sobre los cambios que introduce un estilo, se aplican los del siguiente, luego el siguiente, y así sucesivamente.

El orden en que se aplican los estilos se determina por cuatro factores:

- Posición y orden del estilo. El orden en que aparecen los estilos.
- Especificidad. Hay estilos que aplican a muy pocos elementos html, y otros a muchos o incluso a todos los elementos html de una página.
- Origen: si es un estilo en línea o viene de una hoja de estilos externa.
- Importancia: podemos elegir priorizar unos estilos frente a otros

Ventajas de usar CSS

Las principales ventajas de usar CSS frente a otros sistemas de formato de documentos son:

- Gestión centralizada de los estilos de un solo punto. Esto implica:
 - Ahorro de tiempo
 - Más fácil mantenimiento y detección de errores
- Carga más rápida de las páginas
- Más control sobre los estilos que con sólo HTML. Mucho más flexible.
- Mejor soporte por distintos dispositivos, incluyendo navegadores dentro de dispositivos móviles. Se puede comprobar el soporte de los navegadores a las distintas características en <https://caniuse.com>

Cosas que se pueden hacer con CSS

Con CSS se pueden hacer, entre otras, las siguientes cosas:

- Modificar el aspecto por defecto de los elementos HTML.
- Transformar elementos cambiando su tamaño (escalado), rotándolo, sesgándolo, ...
- Crear animaciones y efectos sin el uso de Java o JavaScript.
- Hacer que una web sea responsive, haciendo que se adapte a los móviles, o a otros dispositivos de mayor tamaño.
- Crear versiones para imprimir de una página.
- ...

Cómo usar CSS

Hay tres formas de utilizar CSS en un documento HTML:

Archivo CSS externo.

- La más habitual y recomendable.
- Se usa un elemento `<link rel="stylesheet">`, que enlaza a un fichero .css independiente. Debe ubicarse dentro de la cabecera (`<head>`).

Bloque de estilos en la página.

- Se usa un elemento `<style>`. Debe ubicarse en la cabecera.

Estilos en línea

- Se usa el atributo `style` en el elemento HTML.

Cómo usar CSS – Fichero externo

Se enlaza con un elemento link:

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

Es la forma recomendada de utilizar CSS. Se recomienda poner estos elementos lo antes posible dentro de <head>, para que se apliquen antes de que se carguen scripts contenido.

Se puede enlazar más de una hoja de estilos. En este caso, el orden es importante. Los estilos se aplican en el orden en que se declaran, y los últimos en declararse pueden sobrescribir los que se declararon primero.

La principal ventaja de las hojas de estilo externas es que centralizamos el aspecto de la web, y con modificar un solo fichero podemos modificar el aspecto de múltiples páginas.

Cómo usar CSS – Bloque de estilos

Se declaran los estilos con un elemento `<style>`

```
<style>  
    div { background: red; color: white;}  
</style>
```

Puede servir para cosas puntuales, pero no es lo más recomendado.

Puede provocar una repetición de reglas de estilo en distintas páginas

Aumenta el peso (tamaño) de los documentos HTML.

También es más lenta la carga de la página, porque no se cachean.

`<style>` tiene que estar dentro de `<head>`, pero funcionará da igual donde lo pongamos. El problema de colocarlo fuera de `<head>` es que puede producirse un "flash" de contenido sin estilos, hasta que se cargan.

Cómo usar CSS – Estilos en línea

Se usa el atributo global style en los elementos html

```
<p style="color:red">Este texto será rojo</p>
```

El atributo style, global, se puede aplicar a cualquier elemento html.

Se pueden poner varios estilos separados por punto y coma ";"

```
<p style="background:gray; color:red">Este texto será rojo</p>
```

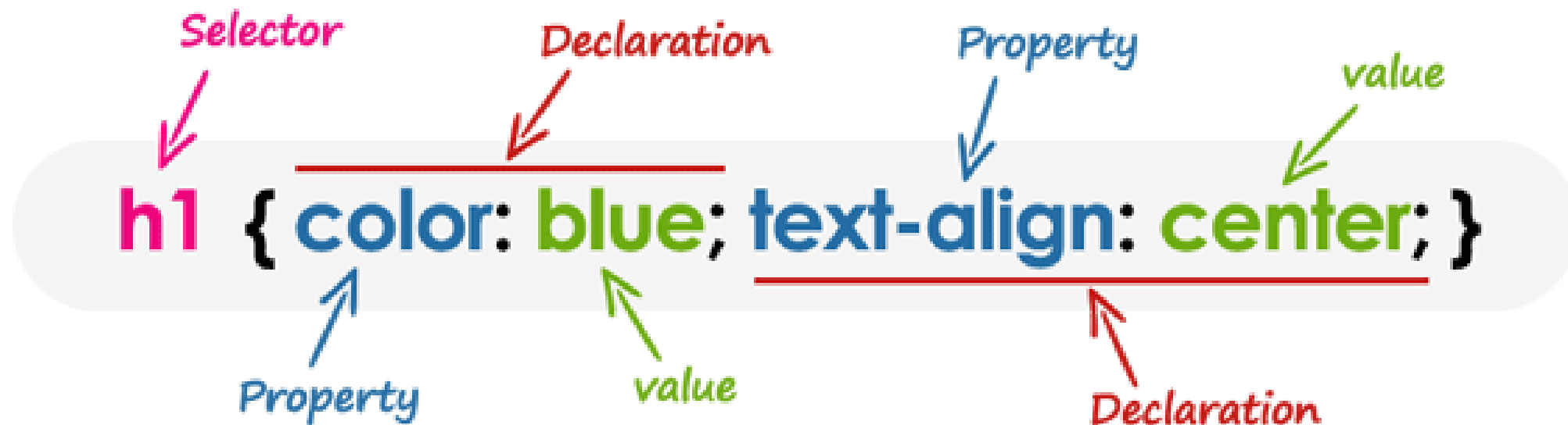
De nuevo, puede servir para cosas puntuales, pero no es lo más recomendado. Siempre se prefiere el uso de una hoja de estilos externa.

Los problemas potenciales son los mismos que con el elemento <style>, puede provocar una repetición de reglas de estilo en distintas páginas y aumenta el peso (tamaño) de los documentos HTML.

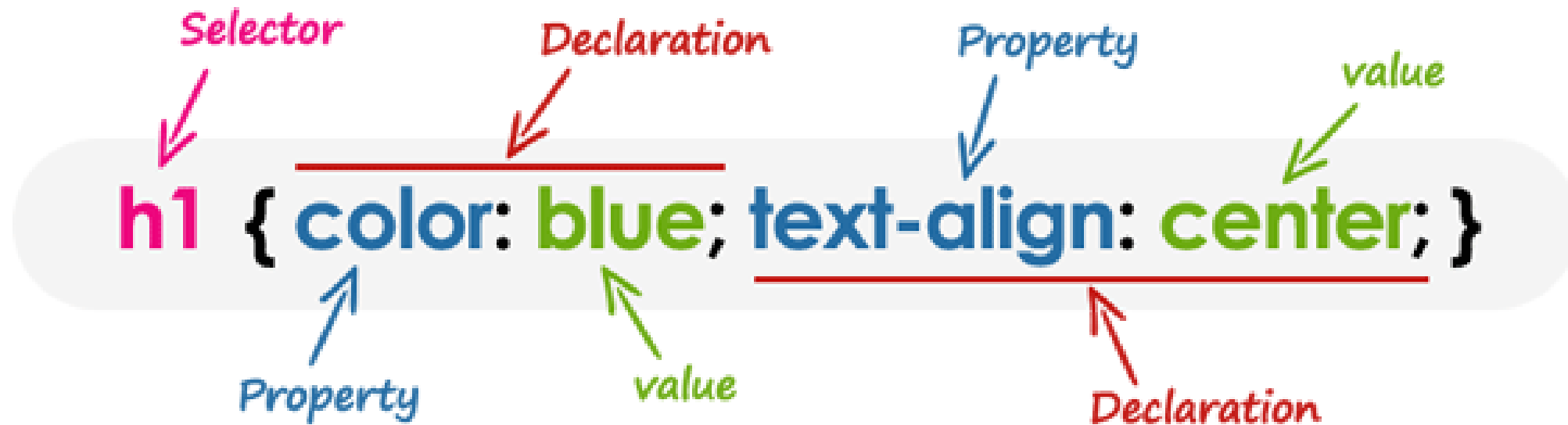
Sintaxis CSS

Una hoja de estilos está formada por reglas, que el navegador aplica a los diferentes elementos.

Una regla CSS tiene dos partes, un selector y una o más declaraciones. Si hay más de una declaración se separan por punto y coma.



Sintaxis CSS



El selector indica a que elemento o elementos del documento HTML se debe aplicar el estilo.

A continuación, se ponen llaves para agrupar todas las declaraciones asociadas al selector.

Cada una de las declaraciones tienen la forma “propiedad: valor”. Las declaraciones pueden estar en la misma línea o en varias.

Sintaxis CSS – Comentarios

Los comentarios en CSS comienzan con “/*” y se cierran con “*/”. Igual que los comentarios de varias líneas en Java.

```
/* Comentario CSS en una línea */  
p { font-weight: bold; }
```

```
/* Comentario CSS  
de varias líneas */  
p { font-weight: bold; }
```

No se pueden hacer comentarios de una línea al estilo Java (//). Si queremos añadir comentarios a una declaración, lo haremos con /**/

```
p {  
    /* Comentario a la declaración */  
    font-weight: bold;  
}
```


Sintaxis CSS - Ejemplos

```
/* Establece fuente para el cuerpo de la página */  
body { font-family: Arial, Helvetica, sans-serif; }
```

```
/* Todos los párrafos con fondo morado y texto azul claro */  
p {  
    color: aqua;  
    background-color: rebeccapurple;  
}
```

```
/* Elementos de lista (li) con borde, una separación del contenido  
respecto al borde y un margen que los separa del resto de elementos. */  
ul li {  
    border: 1px solid red;  
    padding: .1em;  
    margin: 2px 10px;
```

Sintaxis CSS – Reglas @ (at-rules)

Hay ciertas directivas CSS que empiezan con el símbolo @ (at-rules)

Estas reglas o directivas tienen múltiples y variadas utilidades, y veremos algunas más adelante. De momento, vamos a ver sólo @import.

@import permite importar reglas desde otra hoja de estilo.

```
@import url(reset.css);
```

```
@import url(colores.css);
```

```
@import url(https://cdn.jsdelivr.net/npm/bootstrap@5.3.2  
/dist/css/bootstrap.min.css)
```

Sintaxis CSS – Recomendaciones

Un fichero CSS puede estar formado por varios cientos de reglas.

A medida que un fichero CSS crece se hace más difícil de leer y mantener.

Al igual que en otros lenguajes, hay una serie de buenas prácticas que ayudan a leer y mantener el código CSS. Las dos más importantes:

- En cada regla CSS, escribe una declaración por línea.
- Mantén el documento indentado.

Esto, en VS Code es muy fácil de conseguir. Basta con que pidáis a VS Code que formatee el código con el menú contextual (botón derecho)

Se puede activar el formateo automático al guardar un documento. Se hace en File > Preferences > Settings y buscar "Format On Save". También puede ser útil activar "Format On Paste"

Sintaxis CSS – Convención para id y class

Los atributos id y class son muy importantes al definir reglas y selectores CSS (y JavaScript)

Aunque no es obligatorio, se recomienda usar la notación lower-kebab-case para el valor de los atributos id y class.

No recomendado:

```
<section class="newProducts">  
  <header id="newProductsHeader"><h2>Products</h2></header>  
</section>
```

Recomendado:

```
<section class="new-products">  
  <header id="new-products-header"><h2>Products</h2></header>  
</section>
```