

TEMA3. HERENCIA, CASCADA Y ESPECIFICIDAD.

| | |
|---------------------------------------|---|
| 1. HERENCIA | 2 |
| 2. CASCADA | 3 |
| 3. ESPECIFICIDAD | 4 |
| 4. COMO EVITAR CONFLICTOS DE ESTILOS. | 5 |

1. HERENCIA

Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

```
<!DOCTYPE htm>
<html lang="es"
<head>
<meta charset="UTF-8">
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
  body { color: blue; }
</style>
</head>

<body>
  <h1>Titular de la página</h1>
  <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

En el ejemplo anterior, el selector `body` solamente establece el color de la letra para el elemento `<body>`. No obstante, la propiedad `color` es una de las que se heredan de forma automática, por lo que todos los elementos descendientes de `<body>` muestran ese mismo color de letra. Por tanto, establecer el color de la letra en el elemento `<body>` de la página implica cambiar el color de letra de todos los elementos de la página.

```
<!DOCTYPE htm>
<html lang="es"
<head>
<meta charset="UTF-8">
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
  body { font-family: Arial; color: black; }
  h1 { font-family: Verdana; }
  p { color: red; }
</style>
</head>

<body>
  <h1>Titular de la página</h1>
  <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

En el ejemplo anterior, se establece en primer lugar el color y tipo de letra del elemento `<body>`, por lo que todos los elementos de la página se mostrarían con ese mismo color y tipo de letra. No obstante, las otras reglas CSS modifican alguno de los estilos heredados.

De esta forma, los elementos `<h1>` de la página se muestran con el tipo de letra Verdana establecido por el selector `h1` y se muestran de color negro que es el valor heredado del elemento `<body>`. Igualmente, los elementos `<p>` de la página se muestran del color rojo establecido por el selector `p` y con un tipo de letra Arial heredado del elemento `<body>`.

La mayoría de las propiedades CSS aplican la herencia de estilos de forma automática. Además, para aquellas propiedades que no se heredan automáticamente o para aquellos elementos que no heredan automáticamente, podemos utilizar el valor **inherit**, que nos permite heredar una propiedad que no se hereda automáticamente.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de herencia inherit</title>
  <style type="text/css">
    p{ color: green;}
    a{ color: inherit;}
  </style>
</head>
<body>
  <p>Estamos viendo cómo funciona la <a href="#">herencia</a> en CSS</p>
  <p>Estamos viendo cómo funciona la <a href="#">herencia</a> en CSS</p>
</body>
</html>
```

En este ejemplo podemos ver cómo funciona el valor inherit para la propiedad color cuando utilizamos los enlaces con la etiqueta a. Si quitamos los estilos de la etiqueta a, podemos ver cómo funciona.

También tenemos la posibilidad de evitar la herencia por defecto que realiza css, podemos hacer que un elemento no herede las propiedades del elemento padre, para ello vamos a dar el valor **initial** a la propiedad que no queremos que sea heredada.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de herencia inherit</title>
  <style type="text/css">
    p{ color: green;}
    span{ color: initial;}
  </style>
</head>
<body>
  <p>Estamos viendo cómo funciona la <span>herencia</span> en CSS</p>
</body>
</html>
```

En el ejemplo podemos ver como la etiqueta span no va a heredar el color verde del contenedor y va a tomar el estilo del color por defecto que es negro.

2. CASCADA

La cascada hace referencia al orden de importancia que tienen las reglas en CSS cuando dos reglas tienen la misma especificidad,

La cascada establece un criterio cuando dos reglas dan estilo al mismo elemento y tienen la misma especificidad.

El orden de menor a mayor es el siguiente:

- ✓ **Fichero externo**, es el que tiene menor peso.
- ✓ **CSS en el head** tiene un siguiente nivel de peso, es decir, prevalece el estilo del CSS en el head sobre el CSS del fichero externo, cuando son sobre el mismo elemento y tienen la misma especificidad.
- ✓ **CSS en línea** es el que tiene mayor peso de todos y prevalece sobre los anteriores. **Hay que recordar que no es conveniente aplicar estilos en línea.**

Si tenemos dos dentro del alguno de los anteriores, prevalecerá la última regla que haya escrito.

```
p { color: red;}
p { color: blue;}
```

En este caso el párrafo será azul.

3. ESPECIFICIDAD

En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen a un mismo elemento HTML. El problema de estas reglas múltiples es que se generan conflictos de estilos.

Para evitar estos conflictos, la W3C estable un peso para cada uno de los estilos, de tal forma que soluciona los conflictos que se generan.

Los pesos son los siguientes:

- ✓ **Etiqueta** -> 1
- ✓ **Clases y pseudoclases** -> 10
- ✓ **Id** -> 100 (Recuerda que no es aconsejable utilizar id para dar estilos)
- ✓ **!important** -> Tiene mayor peso que todos los anteriores. (Recuerda que no es conveniente utilizarlo)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Especificidad, herencia y cascada</title>
  <style>
    #parrafo{ color: green; }
    .parrafo{ color: red; }
    a{ color: inherit;}
    .parrafo-2{ color: black;}
    p{ color: green;}
    /* p{ color: green !important;}*/
```

```
</style>

</head>
<body>
  <p>Estamos viendo la <a href="#">herencia</a> en CSS</p>
  <p class="parrafo">Estamos viendo la <a href="#">cascada</a> en
  CSS</p>
  <p id="parrafo">Estamos viendo la <a href="#">cascada</a> en
  CSS</p>
  <p class="parrafo-2">El párrafo que queremos cambiar y no
  podremos</p>
</body>
</html>
```

de la página". Por último, el selector `p#especial` sólo hace referencia a "el párrafo de la página cuyo atributo `id` sea igual a `especial`". Como el selector `p#especial` es el más específico, su declaración es la que se tiene en cuenta y por tanto el párrafo se muestra de color verde.

4. COMO EVITAR CONFLICTOS DE ESTILOS.

El orden por defecto en el que `css` resuelve los conflictos de estilos es:

1. Important
2. Cascada
3. Especificidad.

Para evitar los problemas de estilo es conveniente dar los estilos siempre de en fichero externo, para no tener problemas de cascada y por otro lado dar estilos utilizando siempre clases para evitar problemas de especificidad.

Podemos comprobar la especificidad de nuestro `css` en: <https://jonassebastianohlsson.com/specificity-graph/>