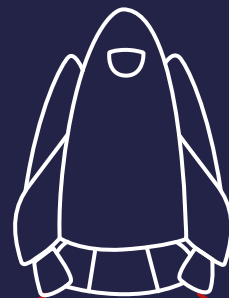


Programa académico CAMPUS



Ciclo 2:
CSS
INTRODUCCION



Agenda



EJEMPLOS

Ejercicios

2

3

1

INTRUDUCCION



Hojas de Estilo en Cascada (**Cascading Style Sheets**), es un lenguaje que nos permite controlar el aspecto de las páginas web escritas en HTML o en cualquier lenguaje de marcado basado en XML.



Ventajas de usar CSS

- Separa la estructura de un documento de su presentación estética.
- Limpia y hace más sencilla y fácil de entender y modificar el código HTML de las páginas.
- Facilita cambiando los valores de un solo archivo (en caso de que el estilo CSS se encuentra en un archivo externo) modificar el estilo de varias páginas a la misma vez.
- Una página web puede tener distintos estilos, para ser usada y vista de forma adecuada en equipos diferentes, ya sea una PC de escritorio, una TabletPC o un teléfono celular.



CSS

HTML



HTML the Skeleton



CSS



CSS the Skin



JS



Javascript the Brain



Funcionamiento

- CSS funciona a base de reglas, es decir, **declaraciones** sobre el estilo de uno o más elementos.
- Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML.



CSS



Uso de Reglas

La regla tiene dos partes: Un **selector** y la **declaración**.

- A su vez la declaración está compuesta por una **propiedad** y el **valor** que se le asigne.

Selector

h1

Declaration

{color:blue; font-size:12px;}

Declaration

Property

Value

Property

Value



Uso de Reglas

Concepto	Descripción
Selector	Elemento o elementos del documento que vamos a seleccionar para aplicarle un estilo concreto.
Propiedad	Característica principal que vamos a definir con el valor indicado a continuación.
Valor	Cada propiedad tiene una serie de valores concretos asignables, con los que tendrá uno u otro comportamiento.
Comentario	Fragmento de texto entre <code>/*</code> y <code>*/</code> con anotaciones o aclaraciones para el desarrollador (el navegador los ignorará).
Regla	Par de propiedad y su respectivo valor asociado.



CSS

Uso de Reglas

selector {

propiedad : valor ;

propiedad : valor ;

propiedad : valor ;

}

una regla por línea

terminada con punto y coma

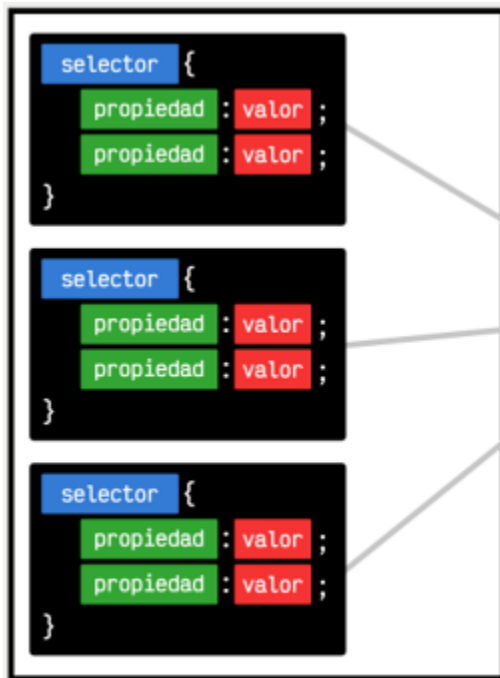
terminación de regla

opcional (al ser la última)



CSS

DOCUMENTO CSS



DOCUMENTO CSS

conjunto de reglas css



CSS



Uso de Reglas

```
h1 {  
  padding-left: 11em;  
  font-family: Georgia, "Times New Roman", Times,  
  serif;  
  color: red;  
  background-color: #d8da3d;  
}
```



Tipos de selectores

Selector universal

Sintaxis: `* { atributo:valor; }`

Ejemplo: `* { color:grey; }/* El estilo se aplicará a todos los elementos de la página*/`



Tipos de selectores

Selector etiqueta

Sintaxis: **etiqueta** { **atributo:valor** }

Ejemplo: **p** {color: green;} /* El estilo se aplicará a todos los elementos <p> */



Tipos de selectores

Selector clase

Sintaxis: **.clase** { atributo:valor }

Ejemplo: **.blend** { color:red; } /* El estilo se aplicará a cualquier elemento que tenga la clase .blend */



Tipos de selectores

Selector identificador

El selector identificador utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento con un id dado en un documento.

Sintaxis: **#id** { atributo:valor }

Ejemplo: **#cent** { color:blue; } /* El estilo se aplicará al elemento que tenga el id #cent */



Cómo funcionan los selectores en CSS

- `tipo`
- `.selector-de-clase`
- `#selector-de-id`
- `[atributo]`
- `*`



Tipos de selectores

Selector descendiente

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.

Sintaxis: **selector1 selector2 selectorN { atributo:valor }**

Ejemplo: **div p {color:black;}** /* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div */



Tipos de selectores

Combinación de selectores

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

Sintaxis: **selector1, selector2, selector3** { atributo:valor }

Ejemplo: div, p { color:orange; } /* El estilo se aplica a todos los divs y párrafos */



Combinaciones de selectores

- Adyacente `.primero + .segundo`
- Hijo `.prade > .hijo`
- Descendiente `.abuelo .bisnieto`
- Hermano `.hermano ~ .hermana`



1. Pintar a Paco y Luis con **Tomato**
2. Pintar al Nieto con **RebeccaPurple**
3. Darle de comer a los Nietos y Bisnietos **Bold**
4. Pintar al Sobrino **Teal**

```
<div class="abuelo">
  <div class="padre">
    <p>Hugo</p>
    <p>Paco</p>
    <p>Luis</p>
    <div>
      <span>Bisnieto</span>
    </div>
    <p>Primo</p>
    <div>Sobrino</div>
  </div>
  <div class="hermano">
    <span>
      Nieto
    </span>
  </div>
</div>
```

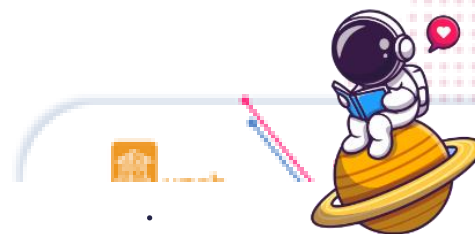


Nombrar correctamente los selectores

Para conseguir más claridad en el código y soporte en todos los navegadores conviene no comenzar el nombre de los selectores con mayúsculas, números ni caracteres especiales.

Evitar: #1div, .=div, DivContent

Mejor utilizar: #div1, .div, divContent



Separar las palabras mediante guiones o mediante mayúsculas

/* Opción 1: Palabras separadas por mayúsculas */
navMenu { padding: 2em; border: 2px solid green; }

/* Opción 2: Palabras separadas por guiones */
nav-menu { padding: 2em; border: 2px solid green; }



Legibilidad

- Adopta una única forma de escribir tu código para que sea más fácil de mantener y de encontrar cualquier elemento.

- */* Opción 1: Estilos en una línea */*

```
.nav-menu { padding: 2em; border: 2px solid green; }
```

- */* Opción 2: Cada estilo en una línea */*

```
.nav-menu {  
    padding: 2em;  
    border: 2px solid green;  
}
```



Combinar elementos

Cuando varios elementos disponen de las mismas propiedades es recomendable compartirlas en vez de volver a repetir el código. Para ello podemos utilizar selectores combinados, tal y como hemos visto anteriormente.

```
h1, h2, h3 { font-family: Arial; font-weight: 700; }
```



Utilizar selectores descendientes

Es conveniente utilizar selectores descendientes siempre que sea posible antes de crear un selector clase o un selector identificador. De esta forma, el código estará más limpio, dispondrá de menos selectores clase e identificador y se comprenderá mucho mejor.

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



Utilizar propiedades abreviadas

Siempre que sea posible es recomendable utilizar clases abreviadas para conseguir una reducción de código.

```
/* Propiedades margin-left, margin-right y margin-top */  
.nav-menu {margin-left: 5px; margin-right: 5px; margin-top: 5px;}
```

```
/* Propiedad abreviada margin */  
.nav-menu {margin: 5px 5px 0px 5px;}
```



Formatos cortos

```
div {  
  padding-top: 8px;  
  padding-right: 16px;  
  padding-bottom: 8px;  
  padding-left: 16px;  
}  
  
div {  
  padding: 8px; /* Se aplica a todos los lados */  
}  
  
div {  
  padding: 8px 16px; /* Vertical, Horizontal */  
}  
div {  
  padding: 8px 16px 0; /* Arriba, Horizontal, Abajo */  
}
```



Utilizar nombres descriptivos en los selectores

Mediante la utilización de nombres que permitan averiguar fácilmente a qué elemento le estamos dando estilo comprenderemos el código fácilmente.

```
.nav-button { background: blue; } /* Estilo del botón de la navbar*/
```



Evitar utilizar como nombre de un selector una característica visual

Al utilizar en el nombre de un selector una característica visual como el color, el tamaño o la posición, si posteriormente modificamos esa característica, también deberíamos cambiar el nombre del selector. Esto complica mucho el código ya que tendríamos que actualizar todas las referencias a ese selector en el código HTML.

```
/* Selector con nombre que define la característica visual del color */  
.menu-red { background: red; }
```

```
/* Utilizar mejor: */  
.nav-menu { background: red; }
```



Probar el diseño en los diferentes navegadores

Para descubrir si se producen errores de visualización en los navegadores lo recomendable es instalarlos todos en el equipo. Algunos de los más utilizados son los siguientes:



<https://www.browserling.com/>

También podemos hacer uso de la aplicación [browserling](#) que permite ver el desarrollo en varias versiones diferentes de cada navegador.



CSS

Validar el código CSS

Detectar errores en el código es esencial y se puede hacer fácilmente mediante un validador CSS. El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS.



<https://jigsaw.w3.org/css-validator/>



Aplicar estilos CSS en un documento HTML

CSS en línea: Los estilos en línea son declaraciones CSS que se integran en las etiquetas HTML mediante el atributo **style**. Este método tan solo afecta al elemento en el que se integra el código. El CSS en línea es complicado de entender y mantener ya que mezcla los estilos CSS con el código HTML.

Ejemplo:

```
<p style="color:green">Párrafo de color verde.</p>
```



Aplicar estilos CSS en un documento HTML

CSS incrustado en la cabecera: se utiliza la etiqueta `<style>` en la cabecera `<head>` del fichero HTML del sitio. La desventaja de este método es que a la hora de realizar cualquier cambio, se debe realizar en múltiples páginas diferentes y el código estará repetido. Su uso puede llegar a ser necesario en el caso de utilizar un gestor de contenido que no permita modificar el archivo CSS directamente.

Ejemplo:

```
<html>
<head>
  <title>CSS incrustado en la cabecera</title>
  <style> p { color: green; } </style>
</head>
<body>
  <p>Párrafo de color verde.</p>
</body>
</html>
```



Aplicar estilos CSS en un documento HTML

CSS en hojas de estilo externas: se utiliza la etiqueta `<style>` en la cabecera. Mediante hojas de estilo externas se consigue separar el archivo de estilos del fichero HTML. El archivo de estilos cuenta con la extensión **.css** y se referencia desde HTML mediante el elemento `<link>`. Este es el método más eficiente y más sencillo de mantener ya que el código CSS se encuentra separado del archivo HTML.

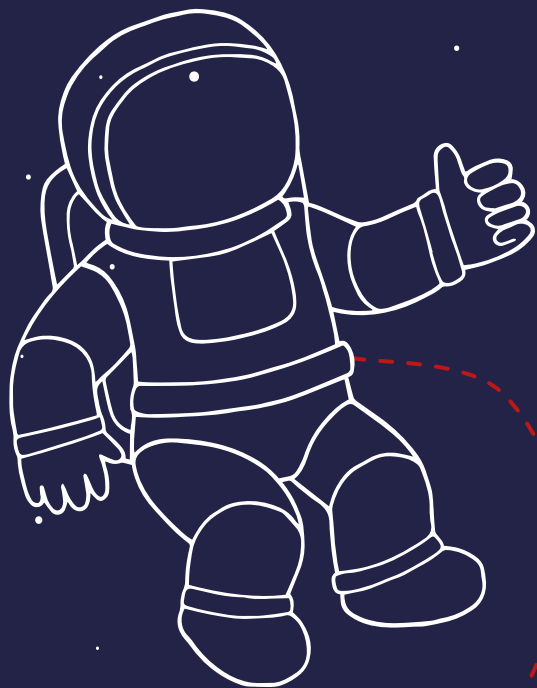
```
<html>
<head>
  <link rel="stylesheet" href="https:\\www.sitioweb\\styles.css">
</head>
<body>
  <p>Párrafo de color verde.</p>
</body>
</html>
```



FORMAS DE ENLAZAR

Nombre	Mediante...	Descripción
Archivo CSS externo	Etiqueta <code><link rel="stylesheet"></code>	El código se escribe en un archivo <code>.css</code> a parte. Recomendado.
Bloque de estilos	Etiqueta <code><style></code>	El código se escribe en una etiqueta <code><style></code> en el documento HTML.
Estilos en línea	Atributo HTML <code>style="..."</code>	El código se escribe en un atributo HTML <code>style</code> en una etiqueta.





Programa acad mico CAMPUS

