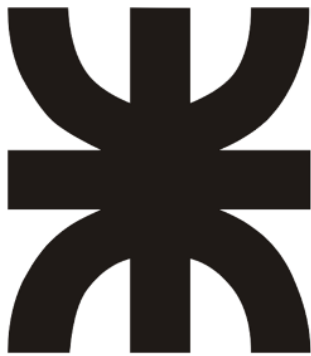


Entornos Gráficos

Práctica 2

DOCENTES

- DÍAZ DANIELA,
- JULIÁN BUTTI.



- Legajo: 44961. Cano Lautaro,
- Legajo: 44762. Gasparini Sofia,
- Legajo: 44800. Lorenzotti Guido

Año de cursado: 2021

Comisión 4E03

Grupo 01

EJERCICIO 1:

1. ¿ Qué es CSS y para qué se usa?

CSS son las siglas de Cascade Style Sheet (hojas de estilo en cascada, en español). Estas hojas de estilo nos permiten controlar la apariencia de una página web.

Con CSS podemos especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros, así como el lugar donde disponer texto e imágenes en la página.

2. CSS utiliza reglas para las declaraciones de estilo, ¿cómo funcionan?

Las declaraciones se dividen en “sentencias”.

Dichas sentencias se dividen en **reglas-arroba** y **reglas**.

Algunas **reglas-arroba** deben colocarse al principio de la hoja de estilo (como @charset, @import, o @namespace), pero el resto pueden colocarse en cualquier lugar de la hoja de estilos. Estas reglas están precedidas, como lo indica su nombre, por el símbolo @.

Las **reglas** están formadas por un **selector** y un **bloque de declaración**. El bloque de declaración empieza y acaba con llaves { }. Cada declaración está formada por una o varias **propiedades** y sus **valores** correspondientes. Las propiedades van separadas entre sí por **puntos y comas**. En general, si una propiedad necesita varios valores, los valores van separados por espacios en blanco. Además, si una propiedad admite varios valores alternativos, los valores van separados por comas.

3. ¿ Cuáles son las tres formas de dar estilo a un documento?

Se puede dar estilo a un documento mediante:

- Utilizando una hoja de estilos externa que estará vinculada a un documento a través de un elemento <link>. Este elemento se deberá colocar en el <head>.
- Aplicando los estilos directamente en los elementos que lo permiten, mediante el atributo <style>, dentro del <body>. Este método es el más desorganizado.
- Utilizando el elemento <style> dentro del <head>, aquí se colocaran todos los estilos dentro del mismo html, por lo que se conocerán antes de que la página cargue por completo.

4. ¿ Cuáles son los distintos tipos de selectores más utilizados?

Se pueden utilizar diversos tipos de selectores, entre ellos:

- **Selectores individuales:** se especifica un solo selector, por lo que el estilo solo se aplicará a él.
- **Selectores agrupados:** es posible escribir dos o más selectores, separados por “,” para que de este modo los estilos se apliquen a todos los selectores definidos.

- **Selector universal:** selecciona todos los elementos de HTML. Se indica con un `*`.
- **Selector de clase:** es necesario agregar el atributo `class` a los elementos que deseamos que se aplique el estilo, y definir un nombre de clase. Luego en el CSS se definirá dicho nombre de clase precedido por un `."`.
- **Selector ID:** es similar al selector de clase, pero el atributo que debe definirse es `"id"`, y en el CSS se llamará a dicho id precedido por un `"#"`. El id sólo puede ser asignado a un único elemento.
- **Selectores descendientes:** permite hacer referencia a un elemento que se encuentre dentro de otro. No solo está limitado a dos selectores. Se define poniendo el elemento padre primero y luego el hijo, separado por un espacio en blanco. También se puede definir que el estilo se aplique solamente a un hijo directo de otro elemento, colocando un `>` entre ambos selectores.

5. Ejemplifique cada uno.

- **Selectores individuales:** `p {color: red}`
- **Selectores agrupados:** `h1,p {color: red}`
- **Selector universal:** `* {background-color: yellow}`
- **Selector de clase:** `.destacado {color: blue}`
- **Selector ID:** `#intro {background-color: green}`
- **Selectores descendientes:**
 - `ul li span {color: blue}`
 - `h1 > span {color: red}`

6. ¿Qué es una pseudo-clase? ¿Cuáles son las más utilizadas aplicadas a vínculos?

Es una palabra clave aplicada a los selectores, y permite especificar un estado de ese elemento, por ejemplo si está vacío, si se pasa el cursor por encima, etc. Entre los más utilizados se encuentran:

- **:empty** : Permite seleccionar elementos que no tengan hijos, ni texto o espacios. Deben estar completamente vacíos.
- **:first-child** : Se utiliza para seleccionar elementos que son los primeros hijos de otros elementos (no los primeros hijos de un elemento, lo cual suele ser una confusión bastante común).
- **:last-child** : Se utiliza para seleccionar elementos que son los últimos hijos de un elemento.
- **:first-of-type** : Selecciona el primer elemento de su tipo entre un grupo de elementos hermanos. Si hay muchos `<p>` en una sección, seleccionará el primero.
- **:last-of-type** : Funciona del mismo modo que el anterior, pero selecciona el último.
- **:nth-child** : Selecciona un elemento en función de su posición con respecto a sus hermanos. por ejemplo `p:nth-child(3)` seleccionará el 3er `<p>`. También se

puede seleccionar una serie de elementos aplicando una **función** de n entre los paréntesis.

También existen pseudoclases referentes a los hipervínculos, como son:

- **:link** : Refiere a un enlace que no ha sido visitado.
- **:visited** : Refiere a un enlace ya visitado.

Y también existen pseudoclases referentes a acciones del usuario:

- **:focus** : Representa al elemento que posee el foco (sobre el cual se ha hecho click, o se selecciona mediante la tecla tab).
- **:hover** : Refiere a un elemento sobre el que se posiciona el cursor encima.
- **:active** : Refiere a un elemento que ha sido activado por el usuario, por ejemplo un botón mientras se está presionando el click sobre el.

7. ¿Qué es la herencia?

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos.

Mediante la herencia, por ejemplo, se pueden especificar las propiedades de las fuentes de los elementos html o body y todo el resto de elementos los heredarán. Se pueden especificar los colores de fondo y de primer plano de un elemento contenedor concreto y todos los elementos hijos de este contenedor heredarán automáticamente el color de primer plano. El color de fondo no se hereda, pero el valor inicial de background-color (color de fondo) es transparent, lo cual significa que el fondo del padre se verá a través de él.

8. ¿En qué consiste el proceso denominado cascada?

Es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.

Este proceso se basa en la importancia, especificidad y orden de las declaraciones.

La **importancia** es determinada por la hoja de estilos en las que se encuentran las declaraciones (navegador, usuario o autor). También se puede utilizar **!important** para que un estilo sea seleccionado por sobre otros.

Si la importancia no elimina la ambigüedad, se pasa a determinar la **especificidad** de los selectores CSS, que es uno de los criterios más importantes de la cascada de CSS.

La especificidad es el modo que tiene el navegador de decidir qué regla se aplica si diversas reglas tienen selectores diferentes pero podrían aplicarse a un mismo elemento. Básicamente, la especificidad mide cuán específica es la selección de un selector:

Un selector de elemento es menos específico (selecciona todos los elementos de aquel tipo que aparecen en la página) por lo que presenta una puntuación más baja en especificidad.

Un selector de clase es más específico (selecciona solo los elementos de una página que tienen un valor de atributo class dado), y por tanto recibe una puntuación mayor.

Serán seleccionados con mayor prioridad los estilos aplicados mediante el atributo style, luego dependerá de la cantidad de veces que aparezca e id, clase, o elemento en el selector.

Cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el **CSS**.

Por último, el **orden** señala donde están ubicados los estilos. Por lo tanto se seleccionara de la siguiente forma:

1. CSS en línea, estilos incluidos en una etiqueta HTML a través del atributo style.
2. CSS interno, definido a través de bloques <style> en el propio documento HTML.
3. CSS externos, son aquellos que se relacionan en un documento HTML a través de la etiqueta <link>.

EJERCICIO 2:

Reglas:

```
p#normal {  
font-family: arial, helvetica;  
font-size: 11px;  
font-weight: bold;  
}
```

```
*#destacado {  
border-style: solid;  
border-color: blue;  
border-width: 2px;  
}
```

```
#distinto {
```

```
background-color: #9EC7EB;  
color: red;  
}
```

Efectos:

- **Normal:** Aplica la familia de tipo de letra “arial, helvetica”, un tamaño de 11 pixeles y negrita a todo párrafo con id=”normal”.
- **Destacado:** Aplica un borde de tamaño de 2 pixeles color azul a todo lo que tenga id=”destacado”
- **Distinto:** Aplica color rojo al texto y el color “#9EC7EB” al fondo a todo lo que tenga id=distinto.

EJERCICIO 3:

```
p.quitar {  
    color: red;  
}  
/*En la primera regla el selector p.quitar nos indica que será aplicada  
a los elementos p dentro del HTML cuyo atributo class sea igual a  
"quitar". Por su parte la declaración { color: red; } nos dice que el  
efecto que produce la regla al ser aplicada es poner el color del texto  
en rojo.*/
```

```
*.desarrollo {  
font-size: 8px;  
}  
/*En la segunda regla el selector *.desarrollo nos indica que será  
aplicada a cualquier elemento dentro del HTML cuyo atributo class sea  
igual a "desarrollo". Por su parte, la declaración { font-size: 8px; }  
nos dice que el efecto que produce la regla al ser aplicada es que el  
tamaño de la fuente (texto) sea de 8 pixeles.*/
```

```
.importante {  
font-size: 20px;  
}  
/*En la tercera regla el selector .importante nos indica que será  
aplicada a cualquier elemento dentro del HTML cuyo atributo class sea
```

```
igual a "importante". Por su parte, la declaración { font-size: 20px; }
nos dice que el efecto que produce la regla al ser aplicada es que el
tamaño de la fuente (texto) sea de 20 pixeles.*/
<!-- A este elemento se le aplicará la segunda regla -->
<p class="desarrollo">
  En este primer párrafo trataremos lo siguiente:
  <br />xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</p>
```

```
<!-- A este elemento se le aplicará la primera regla -->
<p class="quitar">
  Este párrafo debe ser quitado de la obra...
  <br />xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</p>
```

```
<!-- A este elemento no se le aplicará ninguna de las 3 reglas -->
<p>
  En este otro párrafo trataremos otro tema:<br/>
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</p>
```

```
<!-- A este elemento se le aplicará la tercera regla -->
<p class="importante">
  Y este es el párrafo más importante de la obra...
  <br />xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</p>
```

```
<!-- A este elemento no se le aplicará ninguna de las 3 reglas. Si bien
su atributo class es "quitar", dicha clase está definida solamente para
el elemento p y no para el elemento h1 -->
<h1 class="quitar">Este encabezado también debe ser quitado de la
obra</h1>
```

```
<!-- A este elemento se le aplicará la primera y la tercera regla
simultáneamente -->
<p class="quitar importante">Se pueden aplicar varias clases a la
vez</p>
```

EJERCICIO 4:

```
<!-- Todos los elementos tendrán color de texto verde. A excepción de
aquellos que redefinan dicha propiedad -->
```

```
<body>
<!-- El elemento p es el único que aplica la clase "contenido". Su
apariencia tomará lo descripto para p primero (redefine el color del
texto, cambiandolo a negro) a excepción del font-size ya que este
también está definido en la clase contenido y al estar posteriormente
definido es lo que se toma. Además, tampoco tomará el font-weight de la
clase contenido ya que este está definido también en su atributo style
y, por las mismas razones, es lo que se tomará para su apariencia. -->
  <p class="contenido" style="font-weight: normal;">
    Este es un texto .....</p>
  <table>
    <tr>
      <td>Y esta es una tabla.....</td>
    </tr>
    <tr>
      <td>
<!-- Aplica todo lo definido para a. Redefine el color del texto
dependiendo de si el link fue visitado o no, si está actualmente
seleccionado o si se hace foco en él. -->
        <a href="http://www.google.com.pe">con un enlace</a>
      </td>
    </tr>
  </table>
</body>
```

En el otro caso:

```
<!-- Todos los elementos tendrán color de texto verde y lo definido en
la clase "contenido" debido a que se aplicará en cascada el estilo de
body al resto de elementos. A excepción de aquellos que redefinan esas
propiedades. -->
</body>
<body class="contenido">
<!-- Su apariencia tomará lo definido en p (redefine el color,
cambiandolo a negro; redefine el font-size desechando el de la clase
contenido; mantiene el font-weight: bold. -->
  <p> Este es un texto.....</p>
  <table>
    <tr>
      <td>Y esta es una tabla.....</td>
    </tr>
    <tr>
      <td>
```



```
<!-- Aplica todo lo definido para a. (Igual al del caso anterior en  
color aunque cambiando font-size y font-weight ya que aplica la clase  
contenido). -->  
    <a href="http://www.gogle.com">con un enlace</a>  
  </td>  
</tr>  
</table>  
</body>
```