

Ejercicio 1: Responder

1. ¿ Qué es CSS y para qué se usa?

CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para describir la presentación visual de un documento escrito en un lenguaje de marcado, como HTML. CSS se encarga de cómo se deben mostrar los elementos en una página web, incluyendo aspectos como colores, fuentes, márgenes, alineación, y mucho más.

Principales usos de CSS:

Estilizar páginas web: CSS permite aplicar estilos visuales a los elementos HTML, como cambiar colores, fuentes, tamaños de texto, bordes, y sombras.

Diseño y distribución: Con CSS, puedes controlar la disposición de los elementos en la página, crear diseños responsivos que se adapten a diferentes tamaños de pantalla, y diseñar la estructura de la página, como barras laterales, encabezados, y pies de página.

Mejorar la accesibilidad: CSS facilita la mejora de la accesibilidad de las páginas web, permitiendo que los contenidos se adapten a diferentes dispositivos y preferencias de los usuarios, como aumentar el tamaño de fuente o proporcionar contrastes adecuados para personas con discapacidades visuales.

Separación de contenido y presentación: CSS permite mantener el contenido HTML separado de la presentación, facilitando el mantenimiento y la actualización de la página web.

Animaciones y transiciones: CSS también permite agregar animaciones y transiciones suaves entre estados de los elementos, mejorando la experiencia del usuario.

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

En CSS, las **reglas de estilo** se utilizan para aplicar estilos a los elementos HTML de una página web. Cada regla de estilo está compuesta por un **selector** y una o más **declaraciones**.

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

Selector: Es la parte que identifica el elemento HTML al que se aplicará el estilo. Puede ser una etiqueta (como p para párrafos, h1 para encabezados), una clase (como .rojo para un elemento con la clase "rojo"), un ID (como #miParrafo para un elemento con el ID "miParrafo"), o una combinación de estos.

Declaración: Es la parte donde se especifican las propiedades y valores que se aplicarán al elemento seleccionado. Cada propiedad es una característica visual que se puede modificar (como color, font-size, margin, etc.), y el valor asignado indica el estilo que se desea aplicar.

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

1. Estilos en línea (Inline CSS)

Los estilos en línea se aplican directamente a los elementos HTML utilizando el atributo style dentro de la etiqueta del elemento. Esta es la forma más específica de aplicar estilos, pero no es recomendable para grandes proyectos debido a la dificultad de mantenimiento.

2. Estilos en la cabecera (Internal/Embedded CSS)

Los estilos internos se definen dentro de la sección <head> de un documento HTML, utilizando una etiqueta <style>. Este enfoque es útil cuando necesitas aplicar estilos a una sola página.

3. Estilos externos (External CSS)

Los estilos externos se aplican mediante una hoja de estilo CSS separada, que se enlaza al documento HTML utilizando la etiqueta <link> en la sección <head>. Este es el método más común y recomendable para proyectos grandes, ya que facilita el mantenimiento y permite reutilizar la misma hoja de estilo en múltiples páginas.

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

1. Selector de Tipo (Type Selector)

Selecciona todos los elementos de un tipo específico, como todas las etiquetas <p>, <h1>, <div>, etc.

```
p {  
  color: blue;  
  font-size: 16px;  
}
```

2. Selector de Clase (Class Selector)

Selecciona elementos que tienen un atributo class específico. Las clases se definen con un punto (.) seguido del nombre de la clase

```
.mi-clase {  
    background-color: yellow;  
    padding: 10px;  
}
```

3. Selector de ID (ID Selector)

Selecciona un elemento con un atributo id específico. Los IDs son únicos dentro de una página y se definen con un símbolo de almohadilla (#) seguido del nombre del ID.

```
#mi-id {  
    color: red;  
    font-weight: bold;  
}
```

4. Selector de Atributo (Attribute Selector)

Selecciona elementos que tienen un atributo específico o un valor específico de un atributo.

```
input[type="text"] {  
    border: 1px solid black;  
    padding: 5px;  
}
```

5. Selector Universal (Universal Selector)

Selecciona todos los elementos del documento. Se define con un asterisco (*).

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Este selector eliminará los márgenes y rellenos de todos los elementos en la página.

6. Selector de Descendiente (Descendant Selector)

Selecciona los elementos que son descendientes (dentro) de un elemento específico. Se define con un espacio entre dos selectores.

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

Este selector aplicará color verde a todos los párrafos (<p>) que estén dentro de un <div>.

7. Selector de Hijo (Child Selector)

Selecciona los elementos que son hijos directos de un elemento específico. Se define con un símbolo ">" entre dos selectores.

```
ul > li {  
    list-style-type: square;  
}
```

8. Selector de Pseudo-clase (Pseudo-class Selector)

Selecciona elementos en un estado específico, como el estado :hover, :focus, :nth-child, entre otros.

```
a:hover {  
    color: orange;  
}
```

Este selector cambiará el color de un enlace a naranja cuando el usuario pase el cursor sobre él.

9. Selector de Pseudo-elemento (Pseudo-element Selector)

Selecciona y estiliza partes específicas de un elemento, como la primera letra o línea de un párrafo.

```
p::first-line {  
    font-weight: bold;  
}
```

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

Una **pseudo-clase** en CSS es una palabra clave que se añade a un selector y que especifica un estado especial de los elementos seleccionados. Las pseudo-clases permiten aplicar estilos a los elementos en situaciones específicas, como cuando un usuario interactúa con un elemento o cuando un elemento ocupa una posición particular en el árbol del documento.

Pseudo-clases aplicadas a vínculos

Los vínculos (<a>) tienen un conjunto de pseudo-clases comúnmente utilizadas que permiten aplicar estilos según el estado del vínculo. Las más utilizadas son:

1. **:link**: Selecciona los enlaces que aún no han sido visitados.

```
a:link {  
    color: blue;  
}
```

2. **:visited**: Selecciona los enlaces que ya han sido visitados por el usuario.

```
a:visited {  
    color: purple;  
}
```

3. **:hover**: Selecciona un enlace cuando el usuario pasa el cursor sobre él.

```
a:hover {  
    color: red;  
    text-decoration: underline;  
}
```

4. **:active**: Selecciona un enlace en el momento en que se está haciendo clic sobre él.

```
a:active {  
    color: orange;  
}
```

6. ¿ Qué es la herencia?

En CSS, la **herencia** es un concepto que describe cómo algunos valores de las propiedades de los elementos HTML se transmiten automáticamente a sus elementos "hijos". En otras palabras, ciertas propiedades aplicadas a un elemento

"padre" se heredan por los elementos "hijos" dentro de su estructura en el árbol del documento.

7. ¿ En qué consiste el proceso denominado cascada?

El **proceso de cascada** en CSS es el mecanismo que define cómo se aplican los estilos a los elementos de una página web cuando existen múltiples reglas que podrían afectar a un mismo elemento. La cascada determina cuál de las reglas tiene prioridad y, por lo tanto, cuál se aplica finalmente.

Proceso de Decisión de la Cascada

Cuando el navegador tiene que decidir qué estilo aplicar a un elemento, sigue estos pasos:

1. **Cálculo de especificidad:** Calcula la especificidad de todas las reglas que se aplican al mismo elemento.
2. **Comparación de orden:** Si hay empates en la especificidad, el navegador aplica la regla que aparece más tarde en el código CSS.
3. **Aplicación de !important:** Si una regla contiene **!important**, esa regla se aplica, independientemente de la especificidad o el orden de otras reglas, a menos que otra regla posterior con **!important** sea más específica.
4. **Herencia:** Considera los valores heredados y los aplica si no hay reglas que sobrescriban esos valores.

Ejercicio 2

Analizar el siguiente código señalando declaraciones y aplicaciones de reglas, y su efecto.

```
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;

}

<p id="normal">Este es un párrafo</p>

<p id="destacado">Este es otro párrafo</p>

<table id="destacado"><tr><td>Esta es una tabla</td></tr></table>

<p id="distinto">Este es el último párrafo</p>

Selector: p#normal

- Selecciona cualquier elemento <p> que tenga el atributo id="normal".

Aplicación:

- Aplica la familia de fuentes Arial o Helvetica (si Arial no está disponible).
- Establece el tamaño de fuente a 11 píxeles.
- Hace que el texto sea negrita (bold).

Selector: *#destacado

- Selecciona cualquier elemento que tenga el atributo id="destacado", independientemente del tipo de elemento (debido al selector universal *).

Aplicación:

- Aplica un borde sólido (solid).
- Establece el color del borde a azul (blue).
- Establece el ancho del borde a 2 píxeles.

Selector: #distinto

- Selecciona cualquier elemento que tenga el atributo id="distinto".

Aplicación:

- Establece el color de fondo a un tono azul claro (#9EC7EB).
- Establece el color del texto a rojo (red).

Ejercicio 3)

Analizar el siguiente código señalando declaraciones y aplicaciones de reglas, y su efecto.

```
p.quitar {
```

```
color: red;
```

```
}
```

```
*.desarrollo {
```

```
font-size: 8px;
```

```
}
```

```
.importante {
```

```
font-size: 20px;
```

```
}
```

```
<p class="desarrollo">
```

En este primer párrafo trataremos lo siguiente:

```
<br />xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
</p>
```

```
<p class="quitar">
```

Este párrafo debe ser quitado de la obra...

```
<br />xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
</p>
```

```
<p >
```

En este otro párrafo trataremos otro tema:


```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```


</p>

<p class="importante">

Y este es el párrafo más importante de la obra...

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

</ p>

<h1 class="quitar">Este encabezado también debe ser quitado de la obra</h1>

<p class="quitar importante">Se pueden aplicar varias clases a la vez</p>

Selector: p.quitar

- Selecciona cualquier elemento <p> que tenga la clase quitar.

Aplicación:

- Cambia el color del texto a rojo.

Selector: *.desarrollo

- Selecciona cualquier elemento con la clase desarrollo, independientemente del tipo de elemento (el selector universal * es opcional en este caso).

Aplicación:

- Establece el tamaño de la fuente a 8 píxeles.

Selector: .importante

- Selecciona cualquier elemento que tenga la clase importante.

Aplicación:

- Establece el tamaño de la fuente a 20 píxeles.

Análisis de los Elementos HTML

Primer párrafo (<p class="desarrollo">):

Reglas aplicadas: *.desarrollo

Efecto:

El párrafo tendrá un tamaño de fuente de 8 píxeles, haciéndolo visualmente pequeño en la página.

Segundo párrafo (<p class="quitar">):

Reglas aplicadas: p.quitar

Efecto:

Este párrafo tendrá el texto en color rojo.

Tercer párrafo (<p> sin clases):

Reglas aplicadas: Ninguna.

Efecto:

El párrafo se mostrará con los estilos por defecto del navegador, ya que no tiene clases asociadas y, por tanto, no se le aplica ninguna regla CSS definida en el código.

Cuarto párrafo (<p class="importante">):

Reglas aplicadas: .importante

Efecto:

Este párrafo tendrá un tamaño de fuente grande, de 20 píxeles, destacándolo visualmente.

Encabezado (<h1 class="quitar">):

Reglas aplicadas: Ninguna (solo se aplica a <p class="quitar">).

Efecto:

El encabezado no será afectado por las reglas CSS, ya que p.quitar solo aplica a los elementos <p> con clase quitar. Este encabezado se mostrará con los estilos por defecto del navegador para un <h1>.

Párrafo con múltiples clases (<p class="quitar importante">):

Reglas aplicadas: p.quitar, .importante

Efecto:

Este párrafo tendrá el texto en color rojo (por la regla p.quitar) y un tamaño de fuente de 20 píxeles (por la regla .importante). Ambas reglas se aplican sin conflicto, dado que afectan propiedades distintas.

Ejercicio 4

Dadas las siguientes declaraciones:

```
* {color:green; }
```

```
a:link {color:gray }
```

```
a:visited{color:blue }
```

```
a:hover {color:fuchsia }
```

```
a:active {color:red }
```

```
p {font-family: arial,Helvetica;font-size: 10px;color:black; }
```

```
.contenido{font-size: 14px;font-weight: bold; }
```

Analizar los siguientes códigos y comparar sus efectos. Explicar:

```
<body>
<p class="contenido" style="font-weight: normal;">
Este es un texto .....</p>
<table>
<tr>
<td>Y esta es una tabla.....</td>
</tr>
<tr>
<td><a href="http://www.google.com.ar">con un
enlace</a></td>
</tr>
</table>
</body>
```

Análisis y Efectos

- Elemento `<p class="contenido" style="font-weight: normal;">`:

Reglas aplicadas:

.contenido: Establece el tamaño de la fuente a 14 píxeles.

style="font-weight: normal;": Sobrescribe la regla de .contenido y establece el peso de la fuente a normal (no negrita).

Efecto:

El texto del párrafo se mostrará en tamaño 14 píxeles, con fuente no negrita, de color negro (por la regla p { color: black } que sobrescribe la regla * { color: green }).

Elemento <table> y sus contenidos:

Reglas aplicadas:

* {color:green; }: Aplica el color verde al texto de la tabla.

Efecto:

Todo el texto dentro de la tabla será de color verde.

Elemento :

Reglas aplicadas:

a:link {color:gray }: Aplica el color gris al enlace no visitado.

Efecto:

El enlace aparecerá en gris si no ha sido visitado. Cuando el usuario lo visite, cambiará a azul, fucsia al pasar el ratón, y rojo mientras se hace clic.

Segundo código HTML:

```
<body class="contenido">
```

```
<p> Este es un texto.....</p>
```

```
<table>
```

```
<tr>
```

```
<td>Y esta es una tabla.....</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="http://www.google.com.ar">con un enlace</a></td>
```

</tr>

</table>

</body>

Análisis y Efectos

1. Elemento <body class="contenido">:

Reglas aplicadas:

.contenido: Establece el tamaño de la fuente a 14 píxeles y la hace negrita para todos los elementos dentro del <body> (incluyendo los párrafos, la tabla, y los enlaces).

Efecto:

Todo el texto dentro del <body> será de tamaño 14 píxeles y negrita, además de los estilos específicos aplicados a los enlaces.

Debido a la regla global * {color:green; }, todo el texto será verde, a menos que sea sobrescrito por una regla más específica (como en los enlaces).

2. Elemento <p> dentro de <body class="contenido">:

Reglas aplicadas:

.contenido: El texto del párrafo será de tamaño 14 píxeles y negrita.

Efecto:

El párrafo se mostrará en tamaño 14 píxeles, negrita, y color verde (por la regla * {color:green; }).

3. Elemento <table> y sus contenidos:

Reglas aplicadas:

.contenido: Aumenta el tamaño de la fuente a 14 píxeles y la hace negrita.

* {color:green; }: Aplica el color verde.

Efecto:

Todo el texto dentro de la tabla será de tamaño 14 píxeles, negrita, y verde.

4. Elemento :

Reglas aplicadas:

.contenido: El texto del enlace será de tamaño 14 píxeles y negrita.

a:link {color:gray }: Aplica el color gris al enlace no visitado, sobrescribiendo el color verde.

Efecto:

El enlace aparecerá en gris (no visitado), azul (visitado), fucsia (hover) o rojo (activo), con un tamaño de 14 píxeles y negrita.

Ejercicio 5

En cada caso, declarar una regla CSS que produzca el siguiente efecto:

1. Los textos enfatizados dentro de cualquier título deben ser rojos.

h1 em, h2 em, h3 em, h4 em, h5 em, h6 em {

color: red;

}

2. Cualquier elemento que tenga asignado el atributo "href" y que esté dentro de cualquier párrafo que a su vez esté dentro de un bloque debe ser color negro.

div p a[href] {

color: black;

}

3. El texto de las listas no ordenadas que estén dentro del bloque identificado como "ultimo" debe ser amarillo pero si es un enlace a otra página debe ser azul.

#ultimo ul {

color: yellow;

}

#ultimo ul a {

```
color: blue;

}
```

4. Los elementos identificados como “importante” dentro de cualquier bloque deben ser verdes, pero si están dentro de un título deben ser rojos.

```
div .importante {

    color: green;

}
```

```
h1 .importante, h2 .importante, h3 .importante, h4 .importante, h5 .importante, h6
.importante {

    color: red;

}
```

Uso . en vez de # para seleccionar múltiples elementos con una misma característica.

5. Todos los elementos h1 que especifique el atributo title, cualquiera que sea su valor, deben ser azules.

```
h1[title] {

    color: blue;

}
```

6. El color de los enlaces en las listas ordenadas debe ser azul para los enlaces aún no visitados, y violeta para los ya visitados y, además, no deben aparecer subrayados.

```
ol a:link {

    color: blue;

    text-decoration: none;
```

```
}
```

```
ol a:visited {  
  
    color: violet;  
  
    text-decoration: none;  
  
}
```

Ejercicio 6

Dado los códigos de los documentos principal.html y estilo2.css, realizar las modificaciones necesarias en el documento HTML para reemplazar la hoja de estilo interna por la externa estilo2.css (sin modificarla) y obtener la misma salida en el navegador.

Pasos realizados:

- Eliminar hoja de estilo interna(<style>)

- Agregar enlace a hoja de estilo externa estilo2.css

- Ajustes en los IDs y clases: Se cambió id=titulo por id="encabezado" para que coincida con la declaración CSS externa, se agregaron las clases vineta a la lista no ordenada dentro de #menu y estilopie al div de pie de página.

Código HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd">
```

```
<HTML lang=es xmlns="http://www.w3.org/1999/xhtml">
```

```
<HEAD>
```

```
    <TITLE>Página Principal</TITLE>
```

```
    <META http-equiv=Content-Type content="text/html;  
charset=iso-8859-1"></META>
```

```
    <!-- Enlace a la hoja de estilo externa -->
```

```
    <link rel="stylesheet" type="text/css" href="estilo2.css">
```

```
</HEAD>
```


<BODY>

<DIV id="encabezado"><H1>ASIGNATURA ELECTIVA</H1></DIV>

<DIV id="contenido">

<H2>Contenido</H2>

<p>En esta asignatura
</p>

<P>El objetivo fundamental</P>

<P>etc., etc.,</P>

<P>etc., etc.,</P>

<P>Adem´s,</P>

<P>etc., etc.,</P>

<P class="resaltado">Pero, lo más importante es</P>

<P class="resaltado">etc., etc.,</P>

<P>Y, entonces, podemos continuar con</P>

<P>etc., etc.,</P>

<P> </P>

</DIV>

<DIV id="menu">

<H3>Enlaces</H3>

<UL class="vineta">

Curso Actual

Curso anterior

Contacto

</DIV>

<DIV id="pie" class="estilopie">Ingeniería en Sistemas de Información -
Universidad Tecnológica Nacional Rosario</DIV>

<p> </p>

</BODY>

</HTML>

En conclusión el código HTML queda mucho más simple y legible.

Ejercicio 7

Completar el juego para practicar sobre selectores <http://flukeout.github.io/>

