

FUNDAMENTOS DEL DESARROLLO DE PÁGINAS WEB



Módulo 1





CLASE 5

Selectores avanzados

Pseudo-selectores

Modelo de caja

Propiedad width

Box Sizing

Position

Introducción



Hasta aquí tenemos muy avanzado los conocimientos de las estructuras de HTML para poder realizar una página simple. Pero, para dar diseños y estilos más específicos es necesario que veamos esta unidad donde profundizaremos conceptos de CSS para poder posicionar elementos a través del modelo de cajas.

Así que pondremos manos al código y atención a estas explicaciones.

Selectores avanzados

En CSS existen muchos selectores complejos que sirven para seleccionar partes específicas de nuestros documentos HTML.

Selector Universal ()*

Selecciona todos los elementos del archivo HTML. Para darle un borde rojo a todos los elementos de la página usamos el siguiente código:

```
* { border: 2px solid red}
```

Selector De Descendiente (elemento, espacio, otro elemento)

Selecciona todos los elementos del segundo tipo que están contenidos dentro del primero. Así le damos un color azul al texto a todos los `<a>` que se están dentro de un ítem de una lista ``

```
li a {color: blue}
```

Selector De Hijo (>)

El selector de hijo (o child selector) solo aplica para los elementos que están inmediatamente dentro de otros. Es decir, para sus hijos directos. Por ejemplo

Si en HTML tenemos un span dentro de un párrafo que, a su vez, está contenido por un div. Escribiremos el código de la derecha...

Y en el CSS escribiremos...

Es así como quedará seleccionado y se pintará el texto de amarillo ya que es un hijo inmediato del elemento <p>.

```
<!--HTML  -->

<p>Yo no tengo color asignado </p>
<div>
  <p>
    <span>
      Yo tengo color verde
    </span>
  </p>
</div>
```

```
/* CSS */

p > span {color: green}
```

Yo no tengo color asignado

Yo tengo color verde

Selectores avanzados

Agrupación De Selectores (elemento1, elemento2, etc.)

Para seleccionar varios elementos de un mismo HTML, se pueden agrupar separándolos con comas (.). Así, se le aplicará una misma regla a todos ellos. Por ejemplo:

```
h1, p, div {font-size: 30px}
```

Así aplicará un tamaño de letra de 30px a todos los elementos h1, p y div del documento HTML.

También podemos poner otros selectores separados con coma (como ID o clases).

Pseudo-selectores

Los **pseudo-selectores** son **palabras clave** que se añaden a los selectores para especificar diferentes estados. Es decir, se activan dependiendo de lo que haga el usuario. Por lo tanto, si los elementos a los que se aplican estos pseudo-selectores cumplen con estas condiciones o estados, se aplicarán los estilos dados.

Regla general: Los pseudo-selectores se escriben en la Hoja de Estilos, seleccionando un elemento HTML seguido por ":" y el nombre del pseudo selector. Por ejemplo:

```
elemento:hover {color: red}
```

Los pseudo-selectores más frecuentes

- **Hover** (elemento:**hover**): La regla CSS se aplicará únicamente cuando el usuario pase el mouse por encima del elemento indicado. Por ejemplo:

```
h1: hover {color:red;}
```

En este caso, cada vez que se pase el mouse por encima de un <h1> se activará regla CSS expresada entre las llaves, haciendo que el color de la letra cambie a rojo.

- **Focus** (elemento:**focus**): La regla CSS se aplicará cuando el elemento esté en foco (como el input de un formulario que está seleccionado para completar). Por ejemplo:

```
.formulario:focus {border: 2px solid gold;}
```

En este caso, cuando se seleccione un elemento con class="formulario" se activará la regla CSS expresada entre las llaves, haciendo que el borde se ponga color dorado.

Los pseudo-selectores más frecuentes

- **Active** (elemento:active): La regla CSS se aplicará cuando el elemento esté siendo activado. Por ejemplo:

```
#ejemplo:active {background-color: green;}
```

En este caso, cuando se haga click en el elemento con id="ejemplo" se activará la regla CSS expresada entre las llaves, haciendo que el color de fondo se ponga verde.

¿Cómo se activan los elementos?

La activación, generalmente, comienza cuando el usuario presiona el botón primario del mouse y termina cuando se suelta. :active se usa frecuentemente en los elementos <a> y <button>, pero también se puede aplicar a otros.

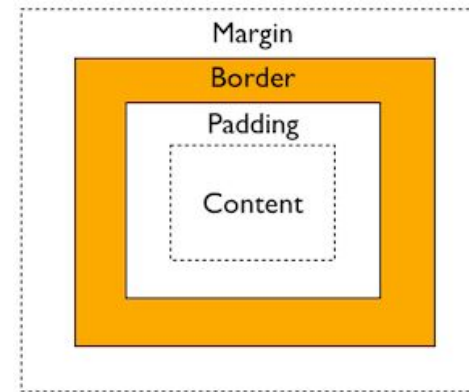
Importante: Los estilos definidos por :active serán anulados por cualquier otro pseudo-selector posterior, relacionado con el enlace, que tenga al menos la misma especificidad (por ejemplo, :link, :hover o :visited).

Box Model

Hay un concepto fundamental en HTML y CSS que es el del Modelo de Caja o Box Model.

En un documento HTML, cada elemento es representado como una caja rectangular con 4 capas de profundidad: **contenido, padding, borde y margen**. Gracias a CSS, podrá manipularlas para tener un diseño más atractivo.

Yendo desde afuera hacia adentro, las capas son:



- **Margen** (margin): es el espacio que separa un elemento de los otros. En este caso, es el espacio entre las cajas.
- **Borde** (border): es la línea de la caja que marca su límite. Podemos hacerlo visible con diferentes grosores, estilos y colores.
- **Padding**: es el espacio que separa el borde de la caja de su contenido.
- **Contenido** (content): es todo lo que está contenido en el elemento. Puede ser, por ejemplo, un texto en un `<h1>`, una imagen o cualquier tag anidado.



Las propiedades margin, border y padding, tienen 4 lados que se pueden modificar individualmente:
Top - Right - Bottom - Left

Para darle diferentes valores a cada uno de los lados, no hace falta escribir:

```
div {  
    margin-top: 5px;  
    margin-right: 2px;  
    margin-bottom: 10px;  
    margin-left: 15px;  
}
```

Podemos ser más eficientes y escribir de forma abreviada:

```
div {  
    margin: 5px 2px 10px 15px;  
}
```

Así, le daremos los 4 valores a los distintos lados siguiendo las agujas del reloj: arriba, derecha, abajo e izquierda.

Importante: También puede usar dos valores en vez de cuatro. En ese caso, el primero se aplicará para los lados de arriba y de abajo (top y bottom) y el segundo para los costados derecho e izquierdo (right y left).

La propiedad *Width*



El **width** se utiliza para cambiar el ancho de un elemento, por ejemplo el de un div.



En esta propiedad, podrá darle a ese elemento un **valor absoluto** (una cantidad de píxeles) o **valor relativo** a la vista del dispositivo (un porcentaje de la pantalla). La diferencia entre uno y el otro es que, usando el relativo se asegura que el diseño sea **responsive** y se adecue a todos los dispositivos.

¿Cómo centrar un elemento usando el valor auto?

El **valor auto** hace que la caja deje el mismo espacio de margen de cada lado. Es decir que, eligiendo este valor, quedará centrada la caja.

Para ver cómo se modifica el posicionamiento de un elemento, seguiremos estos pasos. Pero antes de avanzar al siguiente, miraremos su posicionamiento usando el Inspector de Chrome.

1. En un div color rojo, daremos el valor de margin por margin: 10px 30px 50px 80px; .
2. Luego, cambiamos por margin: 0px 100px;.
3. Por último, le damos un margin: 0px auto; . Así, tendrá 0px para arriba y abajo, y auto para la derecha y para la izquierda. Es decir, dejará el mismo espacio en cada costado.

Box sizing

Es una propiedad que determina qué elemento de la caja se considerará para determinar el ancho final del elemento. Los valores pueden ser content-box o border-box.

Content-Box

Para poder definir el ancho de un elemento, podríamos usar la propiedad width, que especifica el ancho del content. Sin embargo, por default, **CSS sumará el border y el padding** para determinar el ancho final del elemento. A este valor se lo llama Content-Box.

Border-Box

Para ser más precisos y tener más control, conviene usar la propiedad **box-sizing: border-box;**. De esta manera, el browser entiende que **el ancho indicado es el final, incluyendo los bordes y el padding**.

[En este ejemplo](#) verá la diferencia entre border-box y content-box.

Juegue con el CSS Demo y mire cómo cambia el tamaño de las distintas cajas según el criterio que aplique.

La propiedad *Position*

Hasta aquí, usamos los elementos HTML según su posicionamiento por default. Es decir, si son de tipo block, se ubicaran uno debajo del otro y, si son de tipo inline, se posicionarán de izquierda a derecha.

Ahora, aprenderá la propiedad CSS position, para ubicar elementos en distintas partes del sitio.

Hay 4 valores posibles para la propiedad Position:

- **Static:** es el valor por default de todos los elementos HTML. Técnicamente, si un elemento tiene `position: static` no estaría posicionado ya que ocupará su posición por defecto.
- **Relative:** La propiedad `position: relative` nos permite mover el elemento en relación a su ubicación dada por default. Este valor va acompañado de las propiedades `left` o `right` que indican cuánto queremos moverlo hacia los costados y `top` o `bottom` para moverlo hacia arriba o hacia abajo.

Importante: si se posiciona un elemento usando `position: relative`, el espacio ocupado por default, permanecerá bloqueado aunque el elemento se haya movido. Es decir, los demás elementos HTML no ocupan ese lugar.

Valores contradictorios en Position: relative

No deben usarse los atributos `left` y `right` simultáneamente porque sería una orden contradictoria (o se mueve a la izquierda de donde está o hacia la derecha). Lo mismo ocurre con `top` y `bottom`. Hay que elegir uno de cada par.

Por ejemplo:

```
p {  
    position: relative;  
    top: 50px;  
    left: 100px;  
}
```

Importante: Si declaramos la propiedad `position: relative`; y no lo acompañamos de ningún atributo, el elemento permanecerá en el mismo lugar porque no sabrá cuánto se tiene que mover en relación al punto de referencia.

- **Fixed:** fija el elemento en un punto del viewport de la pantalla. Por lo tanto, si hacemos scroll, este elemento no se moverá y quedará por debajo o por encima del contenido.

Valores contradictorios del Position: fixed

Para posicionar un elemento con position: fixed usamos, al igual que en position: relative, los valores contradictorios top o bottom y right o left.

Importante: Los elementos con position: fixed salen del flow de la página. Es decir, quedan volando y, a diferencia de position: relative, su espacio será ocupado por los otros elementos de la web.

- **Absolute:** ubica al elemento en un punto que está en relación a su ancestro posicionado más cercano. Al igual que en fixed, el elemento queda flotando sobre el elemento que lo contiene, ocupando solo el ancho que necesita.

¿Qué sucede si el elemento con position: absolute no está dentro de ningún elemento posicionado?
Si el elemento con position: absolute no está dentro de ningún otro elemento posicionado, se ubicará en relación al body, que es su único ancestro posicionado.

Importante: Dicho ancestro deberá tener declarada la propiedad position con un valor relative o fixed, ya que static es un valor que se toma por default.

MÓDULO 1

+

o



.

GRACIAS

Aquí finaliza la clase n°5 del módulo 1

OTEC PLATAFORMA 5 CHILE