

DAWEB

Práctica 7, día 21 de Marzo 2018

Introducción

El capítulo dedicado a CSS consta de dos prácticas, que corresponden a las sesiones de clase de teoría.

Esta segunda práctica trata de resolver aspectos concretos del diseño de una página, en CSS.

Objetivos

- Familiarizarse con algunas funciones avanzadas de CSS.
- Creación de menús habituales en las páginas web..

Menús empleando CSS

1.- Se plantea resolver ejercicios propuestos en el libro [Introducción a CSS](#)

http://librosweb.es/libro/css/capitulo_15/ejercicio_10.html

http://librosweb.es/libro/css/capitulo_15/ejercicio_11.html

2.- Menú multinivel

Se propone diseñar, usando exclusivamente CSS y HTML, un menú horizontal con cuatro opciones, en una de ellas desarrollar un menú vertical, en el que exista la posibilidad de un menú de tercer nivel en vertical.

Seguidamente se soluciona el problema, explicando cada una de las partes de código en HTML y hojas de estilo CSS.

HTML

El código HTML es muy simple. Se utiliza <nav> como div y se inserta una lista desordenada ().

```
<nav>
  <ul>
    <li><a href="inicio.html">Inicio</a></li>
    <li><a href="asignaturas.html">Asignaturas</a></li>
    <li><a href="umu.html">UMU</a></li>
  </ul>
</nav>
```

Esto es un menú de navegación muy simple con una lista de niveles individuales. Para crear un menú desplegable hay que agregar una lista anidada dentro de uno de los elementos de la lista.

```
<nav>
  <ul>
    <li><a href="inicio.html">Inicio</a></li>
    <li><a href="asignaturas.html">Asignaturas<span
class="caret"></span></a>
      <div>
        <ul>
          <li><a href="asignaturas.html#DAWeb">DAWeb</a></li>
          <li><a href="asignaturas.html#IACCSS">IACCSS</a></li>
          <li><a href="IATEI.html">IATEI</a></li>
        </ul>
      </div>
    </li>
    <li><a href="escepticismo.html">Escepticismo</a></li>
    <li><a href="umu.html">UMU</a></li>
  </ul>
</nav>
```

Si se desea crear más niveles en el menú, entonces es muy fácil, se siguen insertando menús dentro de los elementos de la lista. Lo lógico es usar cuatro niveles como máximo, ya que con más los usuarios se perderían muy fácilmente.

CSS

En el ejemplo se usa el siguiente estilo de cuerpo:

```
body {
  background-color: #EBE8E4;
  color: #222;
  font-family: "HelveticaNeue-Light", "Helvetica Neue Light",
  sans-serif;
  font-weight: 300;
  font-size: 15px;
}
```

En primer lugar hay que diseñar el elemento nav. Se hace de la misma forma que el estilo de una división (div) en CSS.

```
nav {
  background-color: #fff;
  border: 1px solid #dedede;
  border-radius: 4px;
  box-shadow: 0 2px 2px -1px rgba(0, 0, 0, 0.055);
  color: #888;
  display: block;
```

```
margin: 8px 22px 8px 22px;
overflow: hidden;
width: 90%;
}
```

Son de destacar los siguientes elementos CCS3:

Border-radius: Genera esquinas redondeadas que solían ser tan difíciles de crear en versiones previas de CSS. Se ha establecido un radio de 4px en todas las esquinas.

Box-shadow: Añade una sombra alrededor del elemento. Sintaxis; – *box-shadow: h-shadow v-shadow blur spread color inset*

RGBA: RGB significa simplemente Rojo, Verde, Azul, es una alternativa al uso de colores en hexadecimal. Al poner RGBA se ha especificado un valor "Alpha" o de opacidad al sombreado.

Margin: Realizado usando la secuencia habitual (arriba, derecha, abajo, izquierda), especifica el espacio alrededor del elemento.

Overflow: Evitará que cualquier elemento salga fuera, se utiliza para fijar los elementos de la lista.

La siguiente parte del código HTML es la lista, y se diseña el CSS correspondiente.

```
nav ul {
  margin: 0;
  padding: 0;
}

nav ul li {
  display: inline-block;
  list-style-type: none;

  -webkit-transition: all 0.2s;
  -moz-transition: all 0.2s;
  -ms-transition: all 0.2s;
  -o-transition: all 0.2s;
  transition: all 0.2s;
}

nav > ul > li > a {
  color: #aaa;
  display: block;
  line-height: 56px;
  padding: 0 24px;
  text-decoration: none;
}
```

Hay dos aspectos importantes en el código previo:

Inline-block: Al establecer los elementos de lista de esta forma, se pueden organizar en una línea horizontal sin utilizar float.

CSS transition: Se ha usado una transición muy básica, "all" significa todos los atributos CSS y "0.2s" se refiere a la cantidad de tiempo que llevará la transición.

Creación de un triángulo CSS

CSS permite crear diferentes formas cambiando el tamaño de los bordes. Esto significa que se puede crear un icono desplegable / flecha hacia abajo usando solo CSS.

En [CSS-TRICKS](#) muestran en un vídeo la forma de lograr estos triángulos, y en <http://apps.eky.hk/css-triangle-generator/> se puede acceder a un generador de código CSS.

```
nav > ul > li > a > .caret {  
  
border-top: 4px solid #aaa;  
border-right: 4px solid transparent;  
border-left: 4px solid transparent;  
content: "";  
display: inline-block;  
height: 0;  
width: 0;  
vertical-align: middle;  
  
-webkit-transition: color 0.1s linear;  
-moz-transition: color 0.1s linear;  
-o-transition: color 0.1s linear;  
transition: color 0.1s linear;  
}
```

Como se puede ver, se ha establecido el límite de arriba a 4px y se emplean los bordes izquierdo y derecho para formar el borde superior en la forma deseada.

Crear triángulos con CSS es simple y fácil de hacer. Aquí hay algunas combinaciones simples:

Up Arrow: Borde inferior: 5px sólido, borde-izquierdo: 5px transparente, borde-derecho: 5px transparente

Down Arrow: Borde superior: 5px sólido, borde-izquierdo: 5px transparente, borde-derecho: 5px transparente

Left Arrow: Borde-izquierdo: 5px sólido, borde-superior: 5px transparente, borde-abajo: 5px transparente

Right Arrow: Borde-derecho: 5px sólido, borde-superior: 5px transparente, borde-inferior: 5px transparente

Estados Hover

Una regla importante cuando se crean diseños de interfaces de usuario es asegurarse de que las cosas cambien cuando se esperan, esto implica que los estados de Hover en los enlaces son muy importantes. A continuación se muestran los estados de reposo básicos para los elementos de lista en la navegación de primer nivel.

```
nav > ul > li:hover {
  background-color: rgb( 40, 44, 47 );
}

nav > ul > li:hover > a {
  color: rgb( 255, 255, 255 );
}

nav > ul > li:hover > a > .caret {
  border-top-color: rgb( 255, 255, 255 );
}
```

Menú CSS 3 Drop-down

Finalmente se ha de diseñar la div que contiene el segundo menú. Hay dos partes importantes para este estilo:

Position Absolute: Esto colocará el menú fuera de la barra de navegación y permitirá mostrarlo en la parte superior de otros contenidos.

Display None: Por defecto no debe mostrar nada, el menú está oculto hasta que alguien pase el ratón sobre el enlace padre.

```
nav > ul > li > div {
  background-color: rgb( 40, 44, 47 );
  border-top: 0;
  border-radius: 0 0 4px 4px;
  box-shadow: 0 2px 2px -1px rgba(0, 0, 0, 0.055);
  display: none;
  margin: 0;
  opacity: 0;
  position: absolute;
  width: 165px;
  visibility: hidden;
  z-index: 100;

  -webkit-transition: opacity 0.2s;
  -moz-transition: opacity 0.2s;
  -ms-transition: opacity 0.2s;
  -o-transition: opacity 0.2s;
  -transition: opacity 0.2s;
}
```

```

nav > ul > li > div ul > li {
  display: block;
}

nav > ul > li > div ul > li > a {
  color: #fff;
  display: block;
  padding: 12px 24px;
  text-decoration: none;
}

nav > ul > li > div ul > li:hover > a {
  background-color: rgba( 255, 255, 255, 0.1);
}

```

¿Cómo se hace que aparezca el menú desplegable?

```

nav > ul > li:hover > div {
  display: block;
  opacity: 1;
  visibility: visible;
}

```

Menú de tercer nivel

Creación de un nuevo recuadro

Como el cursor actual indica una lista desplegable, se necesita diseñar un cursor para indicar un movimiento lateral. Esto se hace así:

```

nav > ul > li > a > .caret,
nav > ul > li > div ul > li > a > .caret {
  border-top: 4px solid #aaa;
  border-right: 4px solid transparent;
  border-left: 4px solid transparent;
  content: "";
  display: inline-block;
  height: 0;
  width: 0;
  vertical-align: middle;

  -webkit-transition: color 0.1s linear;
  -moz-transition: color 0.1s linear;
  -o-transition: color 0.1s linear;
  transition: color 0.1s linear;
}

```

```

}

nav > ul > li > div ul > li > a > .caret {
  border-bottom: 4px solid transparent;
  border-top: 4px solid transparent;
  border-right: 4px solid transparent;
  border-left: 4px solid #f2f2f2;
  margin: 0 0 0 8px;
}

```

Es importante destacar que se establece el nuevo cursor para tener estilo idéntico al previo. Se modifica la flecha indicadora a través del uso de bordes transparentes.

Contenedor de tercera capa

Al igual que el cursor, se necesita copiar el estilo anterior para el contenedor y luego modificarlo para el tercer nivel.

```

nav > ul > li > div,
nav > ul > li > div ul > li > div {
  background-color: rgb( 40, 44, 47 );
  border-top: 0;
  border-radius: 0 0 4px 4px;
  box-shadow: 0 2px 2px -1px rgba(0, 0, 0, 0.055);
  display: none;
  margin: 0;
  opacity: 0;
  position: absolute;
  width: 165px;
  visibility: hidden;

  -webkit-transition: opacity 0.2s;
  -moz-transition: opacity 0.2s;
  -ms-transition: opacity 0.2s;
  -o-transition: opacity 0.2s;
  -transition: opacity 0.2s;
}

nav > ul > li > div ul > li > div {
  border-radius: 0 4px 4px 4px;
  box-shadow: inset 2px 0 5px rgba(0,0,0,.15);
  margin-top: -42px;
  right: -165px;
}

nav > ul > li:hover > div,
nav > ul > li > div ul > li:hover > div {

```

```
display: block;
opacity: 1;
visibility: visible;
}
```

Donde:

border-radius: 0 4px 4px 4px; - esto fija un radio de 4px a todas las esquinas excepto a la izquierda superior, que da a la caja un buen flujo.

Box-shadow - la sombra está configurada para mostrarse por el lado izquierdo de la caja, por dentro, haciendo que la caja se vea debajo del segundo nivel

margin-top - esto tira de la caja para que esté nivelado con el elemento contenedor

right - tira el contenedor derecho por 165px (el ancho de la caja de segundo nivel)

Anexos:

Prefijos en CSS3

Son modificaciones que se anteponen a las propiedades de CSS3 que no son correctamente interpretadas por los navegadores, de forma que cada navegador, por medio de este modificador actúa de forma correcta en este fragmento de la hija de estilo en cascada.

Los prefijos son específicos de cada navegador, para los principales navegadores del mercado son:

-moz-: Para navegadores basados en Gecko como Mozilla.

-webkit-: Para navegadores basados en Webkit, como Chrome, Vivaldi o Safari.

-o-: Para Opera.

-ms-: Para Internet Explorer y Edge.

En shouldiprefix.com se puede ver qué propiedades necesitan prefijos y cuáles no lo requieren.

Dirección del código HTML y CSS:

<http://www.um.es/docencia/barzana/DAWEB/practicadas/DAWEB-P4-menu-tres-niveles.html>

En el blog WebGenio hay una selección de 100 menús:

<http://webgenio.com/2014/05/22/100-increibles-menus-css/>