

Creación de una página web accesible sencilla

Lic. Iris Fernández

Unidad 3: Creación de un menú

Objetivos:

- Definir regiones en la pantalla a través del uso de DIVs
- Usar un validador de código HTML

Contenidos:

- Uso de divs
- Cambiar el aspecto de los divs usando hojas de estilo
- Creación de un menú accesible
- Uso de menús prediseñados
- El validador del W3C

Actividad 1

Objetivos

- ◆ Crear un menú mediante el uso de DIVs.

Contenidos

- ◆ Creación de divisiones en la pantalla
- ◆ Definición de las características de los DIVs en la hoja de estilo

Creación de un menú para la página

La mayoría de las páginas tiene un menú gracias al cual los usuarios podemos navegar por sus diferentes secciones y a la vez tener una idea de su estructura desde el primer momento, simplemente viendo las opciones y su jerarquía.

Las normas de usabilidad de una página nos indican que se debe tener una barra de navegación –en la parte superior o a la izquierda de la pantalla- que sea muy clara y demuestre la jerarquía de enlaces, un mapa del sitio, un buscador si el sitio es grande, un acceso a la página de inicio en cada página (preferentemente arriba a la izquierda), etc.

[Lista de comprobaciones para las reglas de usabilidad básica:](http://www.nocodestudio.com.ar/articulos/checklist/07.htm)
<http://www.nocodestudio.com.ar/articulos/checklist/07.htm>

A esas normas de usabilidad se le agregan las reglas de accesibilidad:

- La barra de menú no debe estar en un marco, como en muchos sitios antiguos. De hecho, **no se deben usar marcos** en ninguna página.

- La barra de menú no debe ser una tabla. **No se deben usar tablas para maquetar** una web.
- La primera opción de un menú debe ser “Saltar al contenido”, para que los invidentes no necesiten escuchar todas las opciones del menú cada vez que entran a una página. Generalmente se usa el signo # para que se trate de un enlace que se escuche, sin necesidad de verse.
- Todas las opciones del menú deben tener una tecla de acceso rápida, para que los visitantes habituales las memoricen y no necesiten esforzarse pasando entre todas las opciones (en el caso de un problema motriz) o escuchar todas las opciones (en el caso de un invidente).

El primer paso: crear divisiones en la pantalla

Antes de estudiar la manera de cumplir con todas las normas de accesibilidad para un menú, vamos a crear y a diseñar el espacio correspondiente al mismo. Para ello, crearemos una división en la pantalla con el nombre “opciones” a la que después daremos formato a través de la hoja de estilo, y otra división con el nombre “contenido”. El tag **DIV** se utiliza para crear estas divisiones en la pantalla. Una de estas secciones puede tener un nombre que la identifique (*ID= "nombre"*), gracias al cual definiremos el formato en la hoja de estilo.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
  <title>El sistema solar</title>
<link rel="stylesheet" type="text/css" href="miestilo3.css"/>
</head>

<body>
<div id="opciones">
  <ul>
    <li>Opción 1</li>
    <li>Opción 2</li>
  </ul>
</div>
<div id="contenido">
  <h1>El sistema solar</h1>
  <h2>Introducción</h2>
  <p>Aquí voy a reunir información sobre el sistema solar.</p>
</div>
</body>
</html>

```

Si miramos la página en un navegador sin definir el formato en la hoja de estilo, vamos a ver el contenido de la página como si las divisiones no existieran:

- Opción 1
- Opción 2

El sistema solar

Introducción

Aquí voy a reunir información sobre el sistema solar.

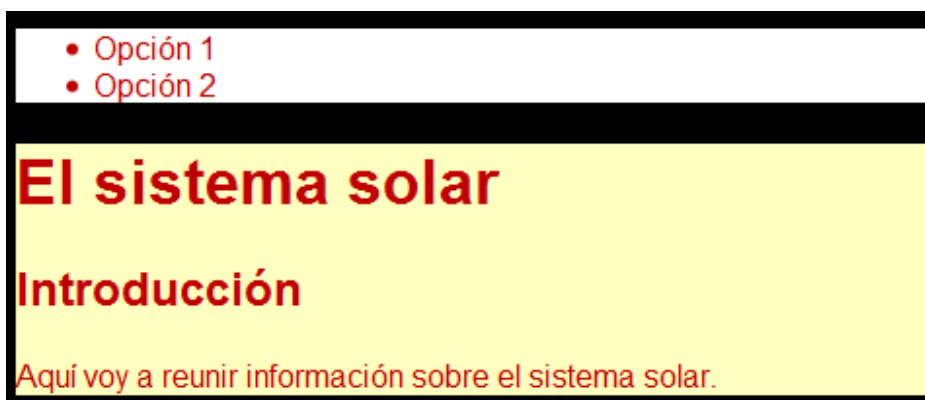
Vamos a crear la hoja de estilo que defina la manera en que se verán esas divisiones en la pantalla. Para empezar a entender un poco dónde está cada parte, vamos a definir colores de fondo (*background-color*), y de letra (*color*):

```
body {
  color: #800000;
  background-color: #000000;
}

#opciones {
  background-color: #FFFFFF;
  color: #C00000;
}

#contenido {
  background-color: #FFC000;
  color: #C00000;
}
```

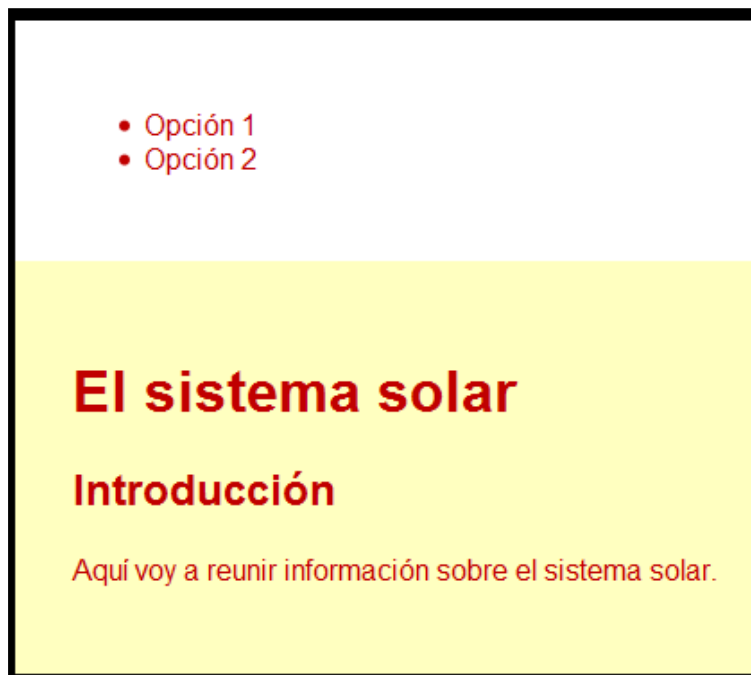
Ahora la misma página se ve de esta manera:



Para mejorar un poco su aspecto podemos agregarle un padding de 2em a cada DIV, distanciando el contenido del borde de la caja que lo contiene:

```
...  
  
#opciones {  
  background-color: #FFFFFFF;  
  color: #C00000;  
  padding: 2em;  
}  
  
#contenido {  
  padding: 2em;  
  background-color: #FFFFC0;  
  color: #C00000;  
}
```

Ahora se verá espacio libre entre el borde de cada caja y su contenido:



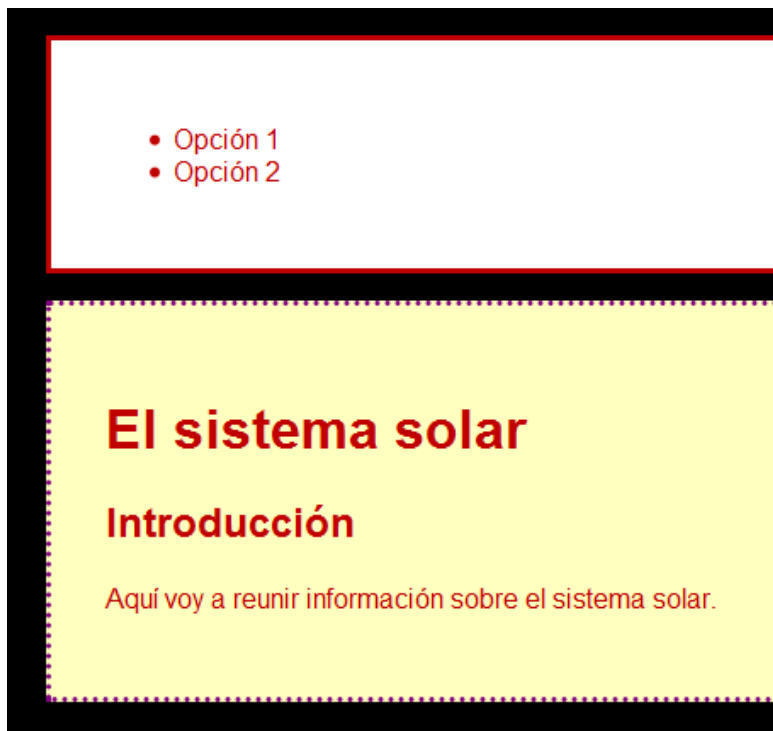
Para separar las cajas entre sí podemos agregar margen; con *border* podemos agregar un marco visible a cada sector de la pantalla:

```
...  
#opciones {  
  background-color: #FFFFFFF;  
  color: #C00000;
```

```
padding: 2em;  
margin: 1em;  
border: 0.2em solid;  
}
```

```
#contenido {  
padding: 2em;  
margin: 1em;  
border: 0.2em dotted #800080;  
background-color: #FFFFC0;  
color: #C00000;  
}
```

Ahora veremos una distancia entre cada caja y la caja que la contiene, además de los bordes:

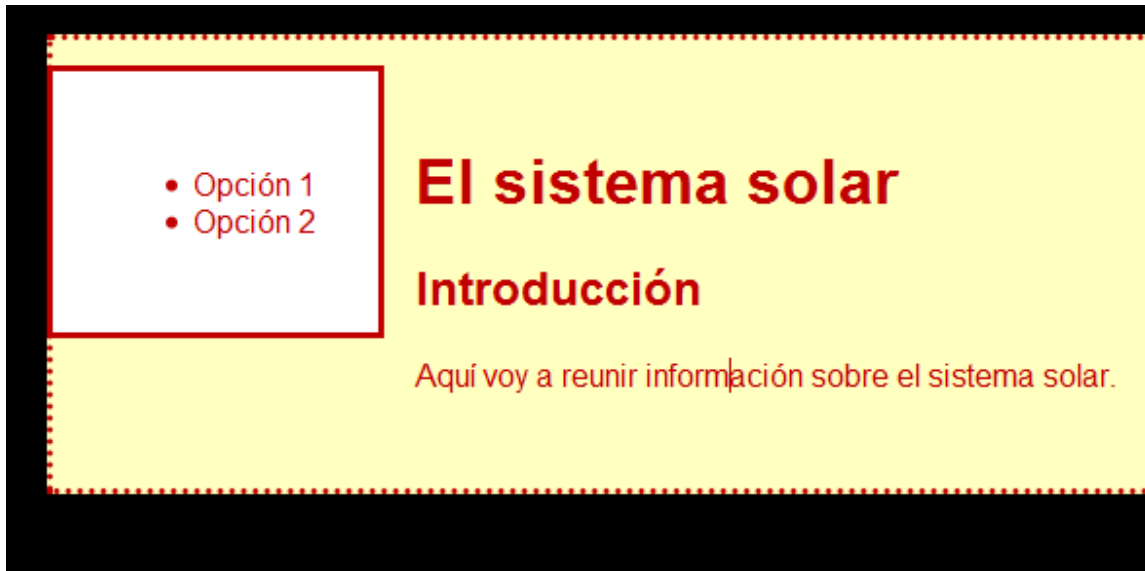


Por último, vamos a pedir que la división correspondiente al menú flote a la izquierda:

```
...  
#opciones {  
background-color: #FFFFFF;  
color: #C00000;  
padding: 2em;  
margin: 1em;  
border: 0.2em solid;
```

```
float: left;
}
```

Ahora el nuevo aspecto de nuestra página será este:



En cambio, si le pedimos que flote a la derecha (*float: right*), el resultado va a ser:



Por otra parte podemos indicarle al DIV que tiene el contenido, que también flote a la izquierda, sin acomodarse por debajo del primer DIV:



Si invertimos el orden de los DIVs en el archivo HTML obtendremos resultados diferentes:

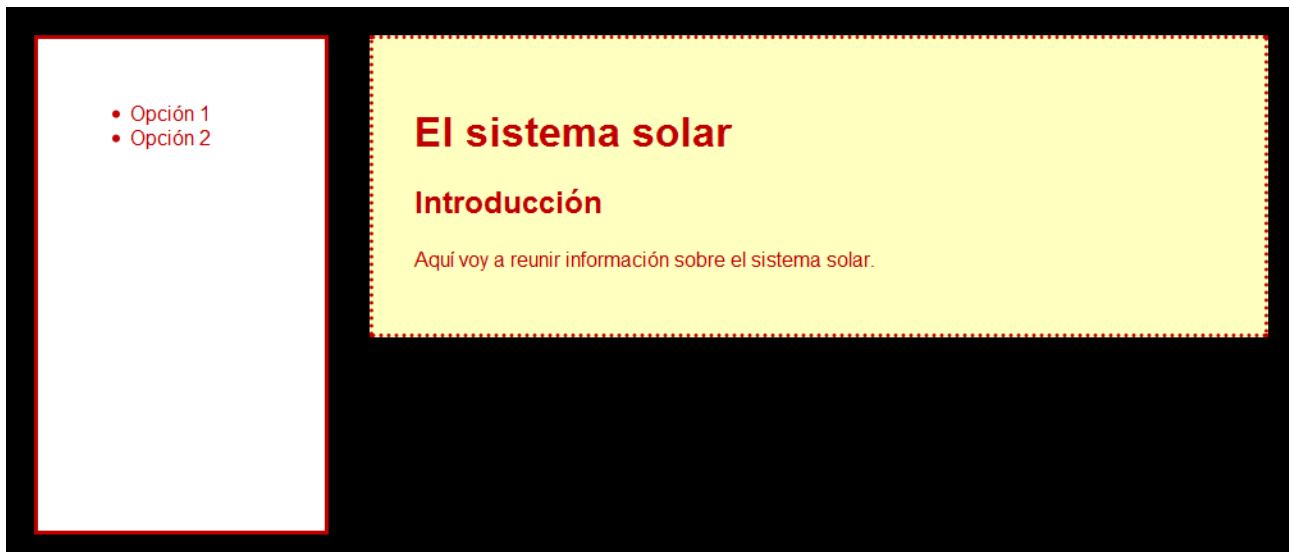


A un DIV se le pueden definir muchas otras propiedades, como el tamaño o la posición exacta donde debe estar, pero no todas serán buenas para todos los navegadores y dispositivos.

Por ejemplo, para definir el ancho (**width**) o el alto (**height**) de un DIV podemos usar medidas en porcentaje o *em* dependiendo del efecto que necesitamos lograr:

```
#opciones {  
  background-color: #FFFFFF;  
  color: #C00000;  
  padding: 2em;  
  margin: 1em;  
  border: 0.2em solid;  
  float: left;  
  width: 10em;  
  height: 20em;  
}
```

```
#contenido {
  padding: 2em;
  margin: 1em;
  float: left;
  border: 0.2em dotted #800080;
  background-color: #FFFFC0;
  color: #C00000;
  width: 50%;
}
```



Ejercicio 1:

Enviar al foro el archivo HTML y la hoja de estilo de una página que tenga una lista de elementos similar a una barra de navegación, y un sector con contenido que ocupe una parte predominante de la pantalla.

Actividad 2

Objetivos

- ◆ Crear un menú con ayudas para la accesibilidad

Contenidos

- ◆ Creación de un menú accesible:
 - Inclusión de una opción para saltar opciones
 - Teclas de acceso rápido

- ◆ Uso de menús prediseñados: List-a-matic

Creación de un menú accesible

Ahora que tenemos diseñada la caja que contendrá el menú, vamos a incluir una serie de opciones en forma de lista no numerada y con links. Cada opción debe tener definida una tecla de acceso rápido:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>El sistema solar</title>
<link rel="stylesheet" type="text/css" href="miestilo3.css"/>
</head>

<body>
<div id="opciones">
<ul>
<li><a href="#principal" title="Saltar al Contenido" accesskey="S">#</a></li>
<li><a href="sistema_solar.html" accesskey="E">El Sistema Solar</a></li>
<li><a href="quienes.html" accesskey="Q">Quiénes somos</a></li>
<li><a href="acerca_de.html" accesskey="A">Acerca del proyecto</a></li>
</ul>
</div>
<div id="contenido">
<h1>El sistema solar</h1>
<h2>Introducción</h2>
<p>Aquí voy a reunir información sobre el sistema solar.</p>
</div>
</body>
</html>
```

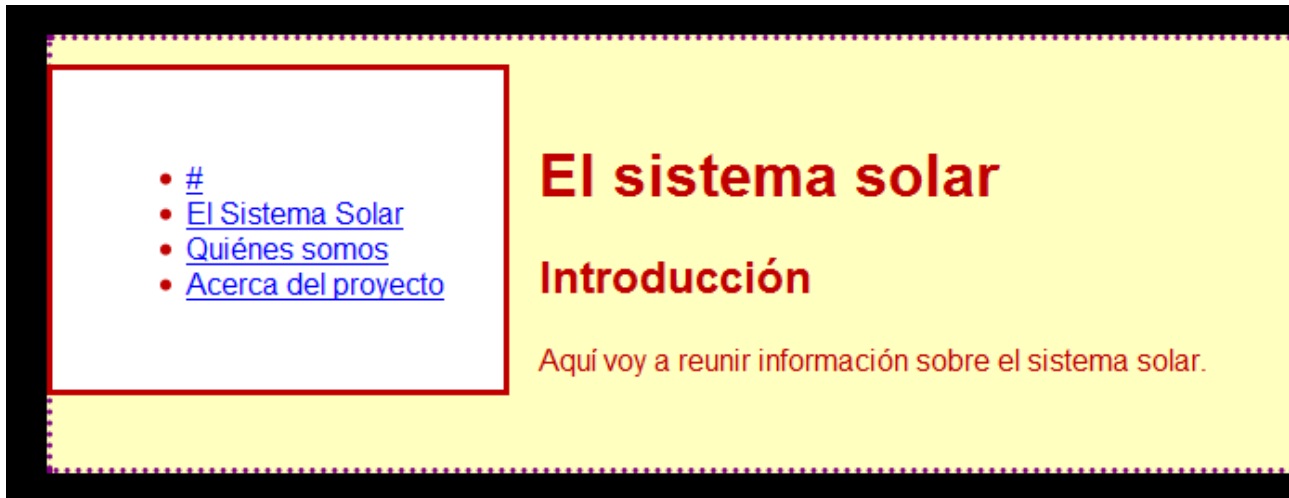
Para que el menú funcione, obviamente tendrán que existir los archivos que se mencionan:

sistema_solar.html
quienes.html
acerca_de.html

Sin embargo, en el caso de la primera opción, no existe un archivo que abrir sino que **el enlace debe llevar al lector de pantalla a un lugar de la misma página en que está el menú**. Para lograrlo, vamos a incorporar en el archivo HTML, antes del contenido:

```
<a name="principal"></a>
```

En principio el menú se verá como cualquier lista con links:



Para usar las tecla de acceso cada usuario deberá conocer el navegador que utiliza. Por ejemplo, en Mozilla Firefox se accede mediante *ALT+MAY+Tecla de Acceso rápido*.

Dar formato al menú

El menú está formado por una lista de enlaces dentro de un DIV. Por lo tanto, para darle formato, deberemos definir cómo serán los enlaces y las listas en ese DIV en particular.

Con respecto a la lista, vamos a sacarle el símbolo de la izquierda. Para ello utilizaremos la propiedad `list-style-type`, que define que el símbolo de la izquierda sea un círculo (*disc, circle*), un cuadrado (*square*), un número (*decimal, lower-roman, upper-roman*), una letra (*lower-alpha; upper-alpha*) o ningún símbolo (*none*).

Si quisiéramos definir el tipo de lista de toda la página lo haríamos de esta manera:

```
li { list-style-type: none; }
```

Sin embargo en este caso queremos definir el tipo de lista que tendrán únicamente las listas que estén dentro del DIV llamado “opciones”. Por lo tanto haremos lo mismo pero comenzando con el nombre del DIV:

```
#opciones li { list-style-type: none; }
```

Incluso podemos indicar que las listas que estén en el contenido lleven un cuadrado a su izquierda:

```
#contenido li { list-style-type: square; }
```

Ahora cada opción del menú se verá como un enlace normal pero sin identificador de ítem de lista:

[#
El Sistema Solar
Quiénes somos
Acerca del proyecto](#)

El sistema solar

Introducción

Aquí voy a reunir información sobre el sistema solar.

- Opción 1
- Opción 2
- Opción 3
- Opción 4

Para que no se vean como vínculos comunes, vamos a darle formato a los enlaces del DIV correspondiente al menú.

Podemos pedir que los enlaces de toda la página se vean de color rojo y sin subrayado:

```
a {  
    color: #800000;  
    text-decoration: none;  
}
```

Sin embargo, es recomendable que **sólo los enlaces del menú** tengan ese aspecto, así que le vamos a agregar el nombre del DIV correspondiente, antes del selector a.

```
#opciones a {  
    color: #800000;  
    text-decoration: none;  
}
```

Los enlaces del resto de la página se seguirán viendo subrayados y con los colores por defecto:

```
#  
El Sistema Solar  
Quiénes somos  
Acerca del proyecto
```

El sistema solar

Introducción

Aquí voy a reunir información sobre el sistema solar.

Este es [un enlace a la Wikipedia](#)

- Opción 1
- Opción 2
- Opción 3
- Opción 4

Para dar un aspecto todavía más parecido al de un menú, podemos cambiar el formato de las opciones cuando el enlace ya se visitó:

```
#opciones a:link {  
  color: #800000;  
}  
  
#opciones a:visited {  
  color: #FF0000;  
}
```

Cuando el enlace está activo (mientras el botón del mouse está presionado):

```
#opciones a:active {  
  color: #C08080;  
}
```

Podemos indicar que el enlace cambie de aspecto cuando el mouse pasa sobre él. En este ejemplo se transforman las letras en mayúscula y el color de fondo en amarillo:

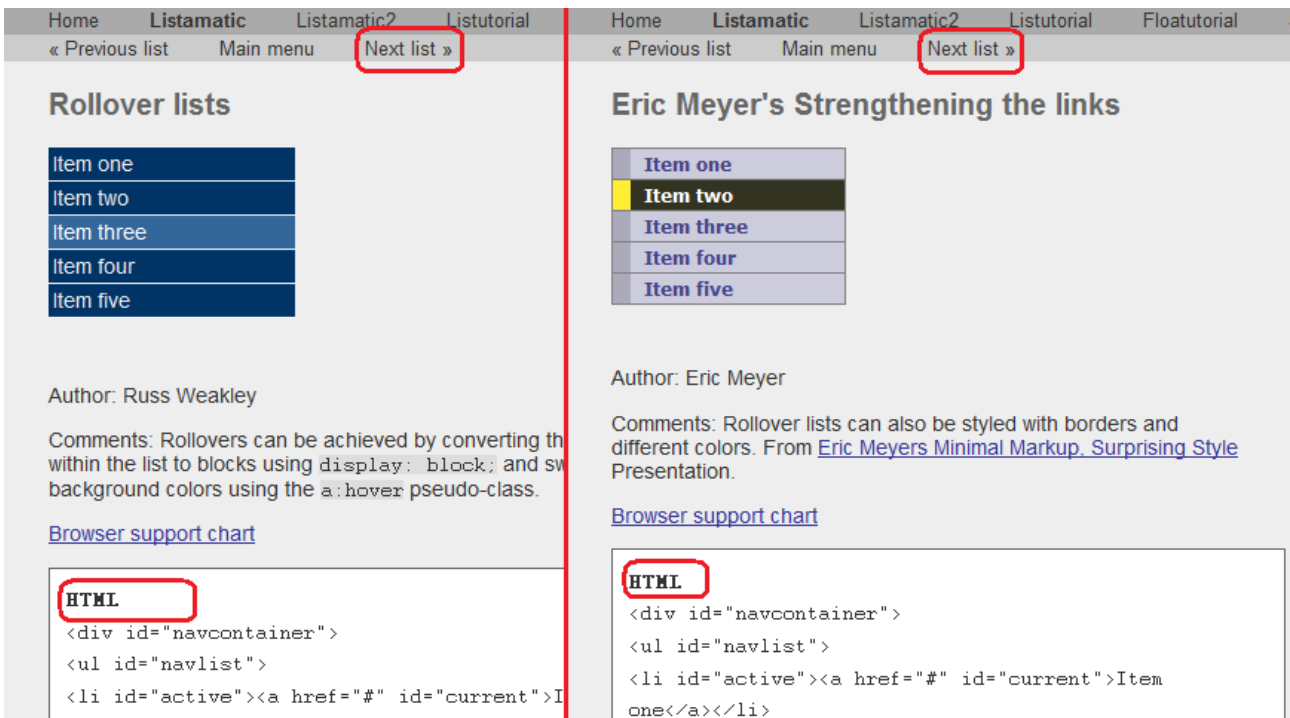
```
#opciones a:hover {  
  text-transform: uppercase;  
  background-color: #FFFC0;  
}
```



Usar un menú prediseñado

Para los que se animen a profundizar, en esta página pueden encontrar muchos ejemplos de menú con su respectivo código y su hoja de estilo:

<http://css.maxdesign.com.au/>



¡Atención! Si usas un menú prediseñado, no olvides agregarle la opción para saltar al contenido y las teclas de acceso, además de evaluar el correcto contraste de colores.

Ejercicio 2:

Enviar al foro el archivo HTML de tu página web, que tenga un menú con dos o tres opciones. Para que el menú funcione, copiar dos o tres veces la misma página y cambiarle el nombre (y si tienes ganas, también el contenido). De esa manera se podrán ver los vínculos ya visitados y activos.

Actividad 3

Objetivos

- ◆ Validar una página web utilizando el validador de W3C

Contenidos

- ◆ Para qué sirve el validador de código de W3C
- ◆ Chequeo y corrección de errores de una página

Validar el código

Es muy importante comprender que **los programas validadores, controlan únicamente el código, ya que no pueden acceder a los contenidos** (por ejemplo, si el texto alternativo a una imagen es “*imagen1*”, “*imagen2*” tenemos una falla de accesibilidad que no puede controlarse por medio de un validador automático).

Por otra parte, hay que tener muy claro que los validadores controlan el código de acuerdo a la declaración de tipo de documento que nosotros realizamos al comienzo del mismo.

Si decimos que usamos XHTML estricto, y en realidad usamos HTML transicional, tendremos muchísimos errores que se solucionan modificando el encabezado.

Por eso hay que saber **leer con inteligencia** los resultados que arroja el validador.

Tres formas de ingresar el código

El validador creado por el *Consortio de la triple W* (w3c), se encuentra en la siguiente dirección:

<http://validator.w3.org/>

Al entrar, vemos tres solapas:

Validate a document online:

- Validar por dirección (URI: *Uniform Resource Identifier*):
 - Copiamos la dirección de la página a validar y la pegamos en la casilla de texto *Adress*.
- Validar subiendo un archivo (*File Upload*):
 - Subimos un archivo de tipo HTML
- Validar mediante ingreso directo (*Direct Input*)
 - Copiamos el código de la página que queremos validar y lo pegamos en el cuadro de texto. Tiene la ventaja de poder hacer correcciones en el mismo cuadro de texto hasta lograr la validación.

Empecemos por validar páginas web que usemos normalmente.

Copiemos algunas direcciones y veamos si el código es correcto o cuántos errores tienen, por ejemplo, la página del diario Clarín, alguna página que hayas hecho o de algún conocido. **Lo más frecuente es que todas las páginas den error**, ya que hay muy poco cuidado de los estándares.

Errors found while checking this document as XHTML 1.0 Transitional!	
Result:	599 Errors, 230 warning(s)
Address :	<input type="text" value="http://www.clarin.com/"/>
Encoding :	iso-8859-1 <input type="text" value="(detect automatically)"/>
Doctype :	XHTML 1.0 Transitional <input type="text" value="(detect automatically)"/>
Root Element:	HTML

Cómo corregir errores en nuestro sitio

Vamos a empezar por validar algunos códigos de ejemplo, para comprender cómo se leen los mensajes de error y cómo se solucionan los problemas.

- Copia y pega este código en la tercer solapa (*Direct Input*), y hacé clic en “*Check*” para controlar el código.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Prueba del validador</title>

```

```
<link rel="stylesheet" type="text/css" href="miestilo4.css">
</head>
<body>
<h1>Estoy usando un validador</h1>
<p>Este es un párrafo.</p>
<p></p>
</body>
</html>
```

- Copia y pega este otro código en la tercer solapa (*Direct Input*), y hacé clic en “*Revalidate*” (o en *Check* si recién entrás)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Prueba del validador</title>
<link rel="stylesheet" type="text/css" href="miestilo4.css"/>
</head>
<body>
<h1>Estoy usando un validador</h1>
<p>Este es un párrafo.</p>
<p></p>
</body>
</html>
```

Ninguno de los dos códigos tiene errores, sin embargo son diferentes. ¿Cuáles son las diferencias entre ambos códigos? **El primero declara el tipo de documento HTML 4.01 estricto, y el segundo XHTML 1.1.**

En estos ejemplos, la única diferencia además del encabezado es que **al usar XHTML los elementos que no tienen tag de cierre (img, link, etc.) tienen que tener una barra de cierre antes del signo mayor:**

```

```

Ahora vamos a hacer otras pruebas, para empezar a entender los errores que nos va a dar el validador. Pega en el validador los siguientes códigos:

```
<head>
<title>Prueba del validador</title>
</head>
<body>
<h1>Estoy usando un validador</h1>
</body>
</html>
```

En este caso el primer error es el único que debemos solucionar:

Line 1, Column 1: no document type declaration; will parse without validation (No se declaró el tipo de documento)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Prueba del validador</title>
<link rel="stylesheet" type="text/css" href="miestilo4.css"/>
</head>
<body>
<h1>Estoy usando un validador</h1>
<p>Este es un párrafo.</p>
<p></p>
</body>
</html>
```

El error que nos da la página es que la imagen no tiene texto alternativo. Nos indica el número de línea y la columna:

Line 10, Column 27: required attribute "alt" not specified

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Prueba del validador</title>
<link rel="stylesheet" type="text/css" href="miestilo4.css"/>
</head>
<body>
<h1>Estoy usando un validador</h1>
<p>Este es un párrafo.
<p></p>
</body>
</html>
```

En este caso el validador nos dice que hay cinco errores. Sin embargo, hay un único error que provoca los otros cuatro:

Line 10, Column 3: document type does not allow element "p" here; missing one of "button", "map", "object", "ins", "del", "noscript" start-tag (“Este tipo de documento no permite un tag “p” aquí”)

Line 11, Column 7: end tag for "p" omitted, but OMITTAG NO was specified (“Se ha omitido el tag de cierre para “p”)

...



Si agregamos el cierre del párrafo donde corresponde, ya no habrá más errores.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Prueba del validador</title>
<link rel="stylesheet" type="text/css" href="miestilo4.css"/>
</head>
<body>
<h1>Estoy usando un validador</h1>
<p>Este es un <strong>párrafo</strong>.</p>
<p></p>
</body>
</html>
```

En este caso el validador nos dice que hay tres errores, pero hay un único error que provoca los otros dos:

Line 9, Column 37: end tag for element "strong" which is not open (se está cerrando un elemento llamado "strong" que no fue abierto)

Line 9, Column 42: end tag for "strong" omitted, but OMITTAG NO was specified (no se encuentra el cierre del elemento "strong")

Line 9, Column 38: XML Parsing Error: Opening and ending tag mismatch: strong line 9 and stong (No coinciden los tipos de tags: "strong" y "stong")

Este error se soluciona corrigiendo el error de tipeo en el tag "Stong", que debería ser "Strong".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Prueba del validador</title>
</head>
<body>
<p>Este es un <u>párrafo</u>.</p>
</body>
</html>
```

En este caso se está usando un elemento antiguo (<u> se utilizaba para subrayar textos):



Line 9, Column 17: element "u" undefined. Did you mean "input" or "ul"?

Lista de códigos válidos y tags caducos (*deprecated*):

<http://www.w3schools.com/tags/default.asp>

Incorporar el sello a la página

Una vez que la página valida, podemos poner en ella el sello de “*Código válido*”, copiando y pegando el código que se nos brinda en la misma página del validador cuando el código no da errores:

	<pre><p> </p></pre>
	<pre><p> </p></pre>

Ejercicio 3: Validar alguna página que hayas hecho en el curso y agregale el sello de XHTML correcto. Si validás un sitio (un conjunto de páginas), cada una de ellas debe pasar por el validador.