

## PRÁCTICA3 HTML5-CSS-JS: RESPONSIVE WEB



### Páginas adaptables

El diseño web adaptable o adaptativo (en inglés, **Responsive Web Design**) es una filosofía de diseño y desarrollo web que, mediante el uso de estructuras e imágenes fluidas, así como de **@media queries** en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario.

Además, en **HTML** debemos utilizar la meta-etiqueta **viewport**. Esta meta-etiqueta fue creada por Apple, pero se ha convertido en todo un estándar que es soportado por la mayoría de los dispositivos móviles (smartphones, tablets y gran parte de móviles de gama media y baja).

Su uso es totalmente necesario, ya que sino el navegador establece el ancho con el que prefiere visualizar una página, en lugar de usar el ancho del que dispone, es decir, si la pantalla de nuestro móvil tiene 400px y el navegador detecta que lo óptimo sería visualizarla con 700px, así lo hará, a no ser que usemos este meta).

Este meta lo añadiremos en el **<head>** de nuestra página:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

Se pueden usar los siguientes parámetros (separados por comas):

- **width**: ancho de la página (se puede establecer en píxeles o como device- width y usará el ancho del que dispone el dispositivo).
- **height**: alto de la página, actúa igual que el width .
- **initial-scale**: escala o zoom inicial de la página (éste y los demás tipos de escala se establecen con valores como 1.0 para no tener zoom o 2.5 para tener un zoom de 2 aumentos y medio, por ejemplo).
- **minimum-scale**: zoom mínimo que podemos hacer en la página.
- **maximum-scale**: zoom máximo que podemos hacer en la página.
- **user-scalable**: establece si está permitido o no hacer zoom (yes/no).

**Más información:** [https://www.w3schools.com/css/css\\_rwd\\_viewport.asp](https://www.w3schools.com/css/css_rwd_viewport.asp)

### @media queries

Las **media queries** son un módulo **CSS3** que permite **adaptar** la representación del contenido a características del dispositivo como la resolución de pantalla, y es un principio básico de la tecnología de Diseño web adaptativo.

La **query** contiene **dos** componentes: Un media type, para qué dispositivos, y la consulta entre paréntesis, que contiene una característica a inspeccionar (max-device-width) seguida por el valor al que apuntamos (ejemplo, 480px). En otras palabras, le estamos preguntando al dispositivo, si su resolución horizontal ( max-device-width ) es igual o menor que 480px.

Además se utiliza **!important**, que funciona como una palabra clave para ignorar las reglas de cascada. Cualquier definición que vaya acompañada de un **!important** tendrá mayor importancia que cualquier otra.

Teniendo en cuenta que cuando tenemos una misma propiedad aplicada dos veces, el navegador hará caso a la última, usaremos **!important**. Funciona como una palabra clave para ignorar las reglas de cascada. Cualquier definición que vaya acompañada de un **!important** tendrá mayor importancia que cualquier otra.

### Ejemplo:

```
@media all and (max-width: 1024px) /* Para todos los dispositivos */
{
    #page
    {
        width:90% !important;
        min-width: 320px;
    }

    #page>header
    {
        font-size:0.7em !important;
    }

    #page>nav>ul>li
    {
        margin:0% 8% 2% 8% !important;
    }
}
```

Esta @media query está definida para un ancho máximo de 1024 píxeles. Hará los siguientes cambios:

- El <div id="page"> tendrá un ancho del 90% y un ancho mínimo de 320 píxeles.
- El <header> tendrá un tamaño de fuente de 0.7em.
- Los <li> del menú existente en la barra de navegación (#page>nav>ul>li) tendrán los siguientes márgenes externos: 0% el superior, 8% el izquierdo y el derecho y 2% el inferior.

**Más información:** [https://www.w3schools.com/css/css\\_rwd\\_mediaqueries.asp](https://www.w3schools.com/css/css_rwd_mediaqueries.asp)

## PRÁCTICA RESPONSIVE WEB

En esta práctica vamos a partir del diseño de la práctica1 de vídeo. La estructura de carpetas será: carpeta **css** con el fichero estilos.css, carpeta **scripts** con el fichero menu.js, una carpeta **imgs** con la imágenes necesarias, y el fichero **index.html** en el directorio raíz.

Añadiremos un **menú** con las siguientes opciones: Contenidos, Manuales y Contacto.

Y vamos a sustituir los elementos del main por tres **artículos**. El diseño quedará como se muestra en la siguiente imagen:



Y el fichero de estilos contendrá, **inicialmente** las siguientes propiedades:

CSS:

```
/* Estilos generales */
article, section, aside, nav, header, footer, figure, figcaption, main
{
display: block;
}
*
{
margin: 0;
padding: 0;
}
```

```
body
{
  background-color: white;
  padding-top: 20px;
  text-align:center;
}

#page
{
  width:960px;
  margin:0 auto;
  text-align:left;

  border: 1px solid black;

  -moz-border-radius: 20px;
  -webkit-border-radius: 20px;
  border-radius: 20px;
}

/* Estilos para la cabecera */

#page>header
{
  color:white;
  border-bottom:6px black solid;

  background: #4D94FF;

  -moz-border-radius: 20px 20px 0px 0px;
  -webkit-border-radius: 20px 20px 0px 0px;
  border-radius: 20px 20px 0px 0px;
}

#page>header h1
{
  padding:20px 100px;
  text-align: center;
  font-family:Courier;
  font-size:2em;
}

#page>nav
{
  font-family:'Verdana';
  font-size:1.4em;
  color: black;
  text-align:center;
}
```

```
#page nav ul li
{
  padding:20px 0px 0px 0px;
  margin:0px 100px 15px 90px;
  list-style:none;
  display:inline-block;
  text-align:center;
  cursor:default;
}
```

```
#page>nav>ul>li>a
{
  text-decoration:none;
  color:#4D94FF;
}
```

```
#page>nav>ul>li>a:hover
{
  color:#005c99;
}
```

```
.clearfix:before,
.clearfix:after,
main
{
  font-family:verdana;
  font-size:0.8em;
  background-color: white;
  width:100%;
  display: table;
  border-top: 2px solid black;
}
```

```
main article
{
  width:240px;
  border:1px lightgray solid;
  background-color:white;
  text-align:center;
  line-height: 1.8em;
  padding:5px 5px 22px 12px;
  margin: 30px;
  float:left;
}
```

```
article header
{
  border-bottom: 1px solid #999999;
}

article header h2
{
  padding: 10px;
  font-size: 1.2em;
}

article header img
{
  width: 150px;
  height:100px;
}

/* estilos para el pie de página */
footer
{
  clear: both;
  font-size:0.7em;
  color:white;
  text-align: center;
  padding:15px 0px;
  border-top:6px solid black;

  background: #4D94FF;

  -moz-border-radius: 0px 0px 20px 20px;
  -webkit-border-radius: 0px 0px 20px 20px;
  border-radius: 0px 0px 20px 20px;
}
```

En este ejercicio vamos a preparar nuestra página index.html para que se adapte al **tamaño** de la ventana. Trataremos de que nuestra página se vea correctamente en cualquier resolución de pantalla.

En primer lugar añadiremos:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">
```

al fichero index.html en el lugar correspondiente, tal y como se ha explicado anteriormente.

Crearemos **tres** @media queries, en el fichero **estilos.css**.

- La **primera** con un **ancho máximo de 1024 píxeles**. Corresponde al **ejemplo** anterior.

- La **segunda** con un **ancho máximo de 700 píxeles**. En esta @media query haremos los siguientes cambios:

- Los <li> del menú existente en la barra de navegación (#page>nav>ul>li) tendrán los siguientes márgenes externos: 0% el superior, 4% el izquierdo y el derecho y 1% el inferior.



- La **tercera** corresponde a la **barra de navegación**. En las versiones para menores resoluciones o tamaños de pantalla, el menú suele convertirse hoy en día, casi un estándar de facto, en un **icono con tres rayitas** que se despliega. Pasamos de tener un menú lineal y visible a un menú desplegable.

Antes de nada prepararemos nuestro documento:

#### HTML:

- Añadiremos un **enlace vacío** dentro del `<nav>`, justo antes del `<ul>`: `<a href="#"></a>`
- Enlazaremos un fichero de script `menu.js` que estará en la carpeta **scripts**:  
`<script src="scripts/menu.js"></script>`  
Con la función de javascript `iniciarMenu()`, descrita más abajo.

En la **tercera** @media query (CSS), **con un ancho máximo de 475 píxeles**, prepararemos nuestra barra de navegación para que se **oculte** y se pueda **desplegar**. Haremos los siguientes cambios:

- Ocultaremos con la propiedad **display:none** el menú de navegación (que es una lista no numerada `<ul>` de la barra de navegación `<nav>`).
- Los márgenes de los `<li>` de la barra de navegación estarán a 0. Y aplicaremos la propiedad `display:block !important;`
- El **enlace** vacío que hemos introducido (`#page>nav>a`) tendrá las siguientes características:
  - Tendrá la imagen de **fondo** `menu.png` que podéis encontrar en la carpeta **imgs** del fichero de recursos.
  - La imagen **no se repetirá** y estará centrada y a 10 píxeles de distancia del borde superior.
  - El **ancho** y **alto** del enlace será de 80px.
  - El enlace tendrá unos **márgenes externos** superior e inferior de 0 y el izquierdo y derecho serán automáticos.
  - Usaremos la propiedad: **display:block !important;**
- Definir la clase **.desplegado**, esta clase la usaremos en javascript, en el código que está más abajo. Debe contener la siguiente propiedad: **display:block !important;**

Ya sólo nos queda hacer que al pulsar el enlace, con la imagen del icono con tres rayitas, el **menú** se **despliegue**. Esto lo haremos mediante una función, `despliegaMenu`, en javascript.

```
'use strict';
```

```
let enlaceMenu;
```

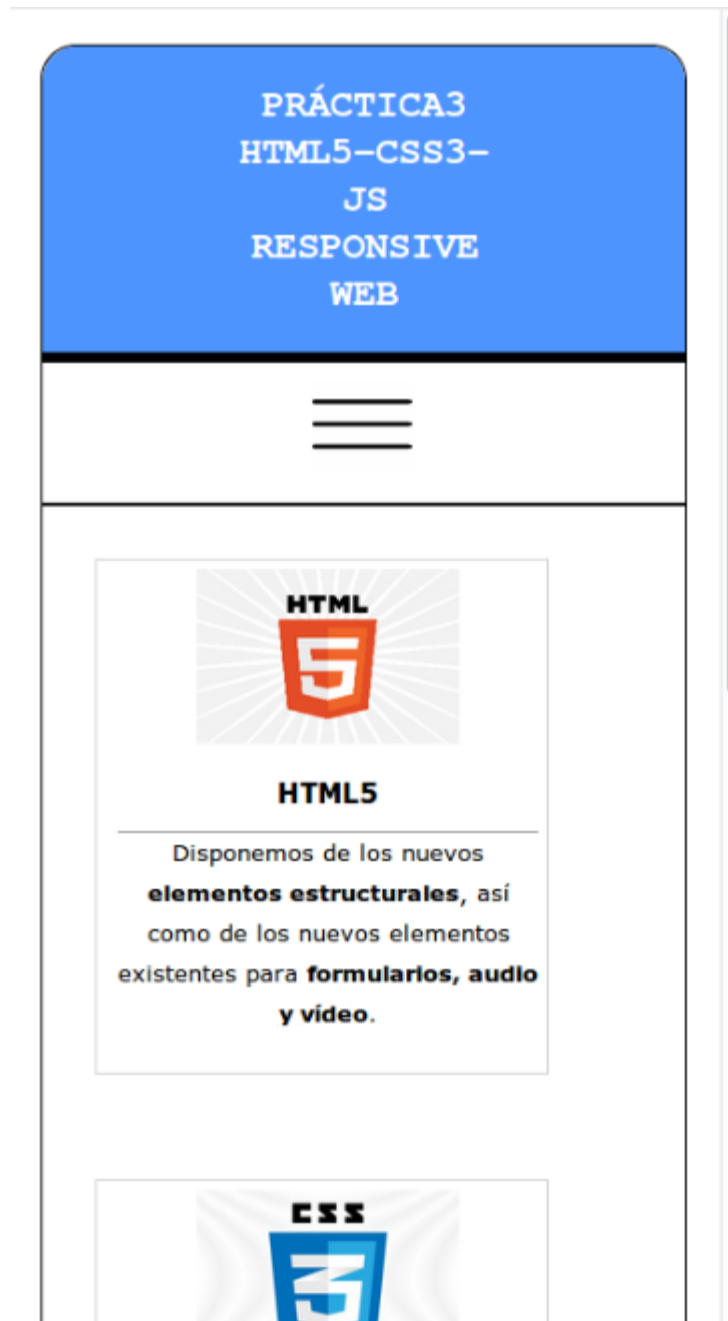
```
function iniciarMenu()
{
    enlaceMenu = document.querySelector("#page>nav>a");
    enlaceMenu.addEventListener("click", despliegaMenu, false);
}
```

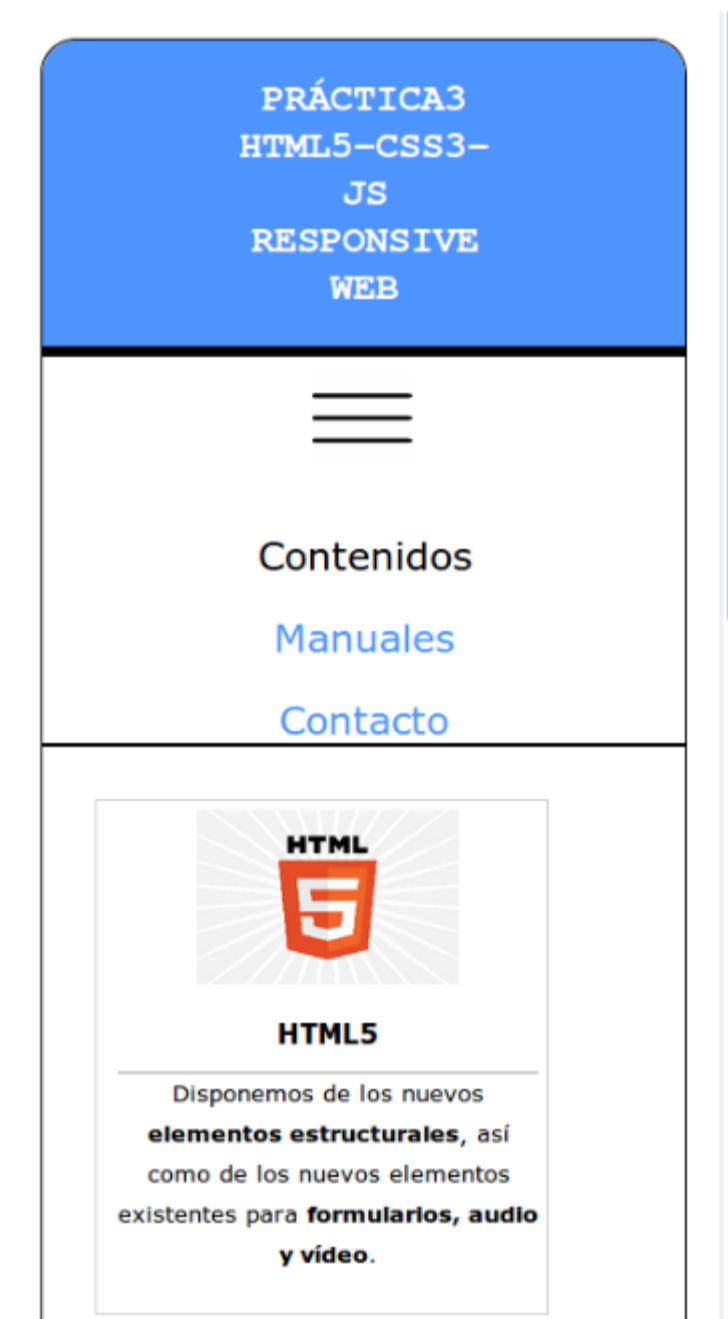
```
function despliegaMenu()
{
    document.querySelector("#page>nav>ul").classList.toggle('desplegado');
}
window.addEventListener("load", iniciarMenu, false);
```



En el código anterior, al pulsar en el enlace menú se le aplicará la **clase** desplegado al <ul> si no la tiene aplicada, o se eliminará dicha clase, si ya la tiene aplicada. Esto lo hacemos con el método **classList.toggle('desplegado')**.

Funcionará del siguiente modo:





### ENTREGA DE LA PRÁCTICA:

- No olvides pasar el **validador** de HTML y CSS.
- Comprime todos los ficheros generados en un uno solo con el nombre: **NombreAlumno-P3ResponsiveWeb.zip**
- Sube el fichero a la entrega correspondiente del curso del moodle.