



CEU

| *Centro de Estudios
Profesionales*

LENGUAJE DE MARCAS
1º DAW – 1º DAM

TEMA 14

Meta etiquetas en Head y responsive



Elemento “meta” en la cabecera

El elemento <meta> se utiliza normalmente para especificar el juego de caracteres, la descripción de la página, las palabras clave, el autor del documento y la configuración de la ventana gráfica.

Los metadatos no se muestran en la página, pero serán utilizados por los navegadores (cómo mostrar el contenido, cómo recargar la página), los motores de búsqueda (palabras clave) y otros servicios web.

Veamos las meta etiquetas más usadas:

1. Charset

Define el juego de caracteres a usar:

```
<meta charset="UTF-8">
```

2. Keywords

Define palabras clave para los motores de búsqueda:

```
<meta name="keywords" content="Bormujos, Ceu, Noticias, informática ">
```

3. Description

Define una descripción de tu página web:

```
<meta name="description" content="Noticias CEU en Grado Informática ">
```



4. Autor

Define el autor de la página:

```
<meta name="author" content="Alicia Parrado">
```

5. Refresh

Refresca la página por un período de segundos configurados:

```
<meta http-equiv="refresh" content="30">
```

Esta página se refrescaría cada 30 segundos.

6. Viewport

Configura la página web para que se vea bien en todos los dispositivos.

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

Width=device-width ---→ establece que el ancho de página siga al ancho del dispositivo en el que se esté visualizando la página que puede variar.

Initial-scale=1.0 --→ establece el zoom inicial cuando el navegador carga la página por primera vez.

Elemento “base” en la cabecera

El elemento <base> especifica la URL base y/o el destino de todas las URL relativas de una página.

La etiqueta <base> debe tener presente un atributo href o target, o ambos.



¡Solo puede haber un único elemento <base> en un documento!

Ejemplo:

```
<base href="https://www.diariodesevilla.com/" target="_blank">
```

Vamos a crear un archivo denominado tema14.html:

Escribiremos el siguiente código:

```
<?> tema14.html > html > body > hr
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <!-- ESTABLECE EL JUEVO DE CARACTERES-->
6      <meta charset="UTF-8">
7      <!-- ETIQUETA PARA HACER RESPONSIVE LA PÁGINA, ADAPTACIÓN DE ESCALAS -->
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
9      <!-- ESTABLECE UNA DESCRIPCIÓN DE LA PÁGINA -->
10     <meta name="description" content="Curso Html Tema 14">
11     <!-- ESTABLECE PALABRAS CLAVE PARA SER LOCALIZADA LA PÁGINA POR LOS MOTORES DE BÚSQUEDA -->
12     <meta name="keywords" content="HTML, CSS, JavaScript, Meta, Responsive">
13     <!-- ESTABLECE EL AUTOR DEL ARCHIVO-->
14     <meta name="author" content="Fernando Gallego">
15     <title>Tema 14 html</title>
16     <!-- ICONO DE VENTANA -->
17     <link rel="icon" type="image/x-icon" href="imagenes/favicon.ico">
18 </head>
19
20 <body>
21     <!-- LINK EXTERNOS ABSOLUTOS -->
22     <a href="https://www.diariodesevilla.es/">Abrir diario de Sevilla</a>
23     <hr>
24     <a href="https://www.diariodesevilla.es/aljarafe/">Abrir diario de Sevilla noticias Aljarafe</a>
25     <hr>
26     <a href="https://www.diariodesevilla.es/economia">Abrir diario de Sevilla apartado economía</a>
27     <hr>
28 </body>
29
30 </html>
```

Vemos que hemos puesto las meta etiquetas que hemos visto anteriormente.

Ahora tenemos 3 enlaces con url's absolutas que abren distintos apartados de la web del diario de Sevilla.



Si probamos nuestro archivo:

[Abrir diario de Sevilla](#)

[Abrir diario de Sevilla noticias Aljarafe](#)

[Abrir diario de Sevilla apartado economía](#)

Vemos que cuando pulsamos cualquier link lo abre en la misma ventana en la que estamos.

Ahora vamos a hacer uso del elemento `<base>` que hemos visto antes:

```
<!-- ICONO DE VENTANA -->
<link rel="icon" type="image/x-icon" href="imagenes/favicon.ico">
<!-- PONEMOS URL BASE -->
<base href="https://www.diariodesevilla.es/" target="_blank">
</head>

<body>
  <!-- LINK EXTERNOS ABSOLUTOS -->
  <a href="">Abrir diario de Sevilla</a>
  <br>
  <a href="aljarafe">Abrir diario de Sevilla noticias Aljarafe</a>
  <br>
  <a href="economia">Abrir diario de Sevilla apartado economía</a>
  <br>
</body>

</html>
```

Si probamos nuestro ejercicio, podéis comprobar que se abre en una nueva pestaña, y hemos ahorrado a la hora de especificar las url's poniéndolas de forma relativa a partir de la url Base.



Hemos visto como hemos utilizado la URL Base para rutas absolutas, vamos a utilizarla ahora para rutas relativas dentro de nuestro equipo:

Vamos a crear un directorio en el mismo directorio donde se encuentra nuestro archivo tema14.html denominado “paginastema14” y en este directorio tendremos 2 archivos html:

Tema14_1.html

Tema14_2.html

Los contenidos de estos archivos son los siguientes:

Tema14_1.html:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <!-- ESTABLECE EL JUEVO DE CARACTERES-->
  <meta charset="UTF-8">
  <title>Tema 14_1 html</title>
</head>

<body>
  <p>ESTAMOS EN LA PÁGINA TEMA 14_1</p>
</body>

</html>
```

Tema14_2.html:



```
<!DOCTYPE html>
<html lang="es">

<head>
  <!-- ESTABLECE EL JUEVO DE CARACTERES -->
  <meta charset="UTF-8">
  <title>Tema 14_2 html</title>
</head>

<body>
  <p>ESTAMOS EN LA PÁGINA TEMA 14_2</p>
</body>

</html>
```

Ahora vamos a modificar nuestro archivo tema14.html para poner una ruta base relativa:

```
<!-- PONEMOS URL BASE -->
<!-- <base href="https://www.diariodesevilla.es/" target="blank"> -->
<base href="paginastema14/" target="_blank">
</head>

<body>
  <!-- LINK EXTERNOS ABSOLUTOS -->
  <!-- <a href="">Abrir diario de Sevilla</a>
  <hr>
  <a href="aljarafe">Abrir diario de Sevilla noticias Aljarafe</a>
  <hr>
  <a href="economia">Abrir diario de Sevilla apartado economía</a>
  <hr> -->

  <!-- LINKS RELATIVOS -->
  <a href="tema14_1.html">Abrir Tema 14_1</a>
  <hr>
  <a href="tema14_2.html">Abrir Tema 14_2</a>
  <hr>
</body>
```



Hemos comentado los links de antes pues no funcionarían al cambiar la url base, podemos comprobar si ejecutamos nuestro archivo que también abre nuestros archivos locales con la ruta base relativa.

[Abrir Tema 14_1](#)

[Abrir Tema 14_2](#)

Responsive: adaptación del ancho de imágenes

Ya hemos visto en temas anteriores que, para adaptar una imagen a la pantalla del dispositivo, utilizamos el atributo “width” de la imagen y le ponemos el valor del 100%, esto hace que la imagen sea responsive y se escale hacia arriba o hacia abajo en tamaño.

Vamos a realizar el siguiente ejercicio en nuestro archivo tema14.html:

Accederemos al siguiente link y nos descargaremos la imagen:

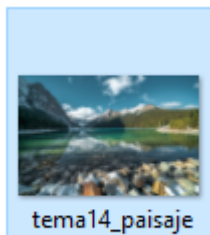
https://img.freepik.com/foto-gratis/impresionante-foto-hermosas-piedras-agua-turquesa-lago-colinas-fondo_181624-12847.jpg?w=996&t=st=1707677637~exp=1707678237~hmac=3090f98c6de47d8dbc74a3160cbcd403ea809c0ee17eedaeb2f13ef8872695a

la alojaremos dentro del directorio “imagenes” dentro del propio directorio en el que se encuentra nuestro archivo tema14.html.



Le llamaremos a la imagen:

https://img.freepik.com/foto-gratis/impresionante-foto-hermosas-piedras-agua-turquesa-lago-colinas-fondo_181624-12847.jpg?w=996&t=st=1707677637~exp=1707678237~hmac=3090f98c6de47d8dbc74a3160cbcdd403ea809c0ee17eedaeb2f13ef8872695a



El tamaño de nuestra imagen es (en pixeles):

Ancho		Altura
<input type="text" value="996"/>	↔	<input type="text" value="664"/>

Ahora escribimos en nuestro archivo html:

```
<!-- LINKS RELATIVOS -->
<a href="tema14_1.html">Abrir Tema 14_1</a>
<hr>
<a href="tema14_2.html">Abrir Tema 14_2</a>
<hr>
<!-- USO RESPONSIVE DE IMÁGENES -->

</body>
```

Y probamos nuestro ejercicio.


Ups!!! Vemos que no carga la imagen. ¿A qué se debe? Razonar la pregunta.


Pues a que aún enemos una url base para nuestras rutas, al igual que a los links a href, también afecta a las rutas de cargas de cualquier tipo de archivo.


Para que pueda encontrar la imagen, tendremos que moverla de carpeta.

Vamos a copiar nuestra imagen a la misma carpeta que teníamos los archivos: tema14_1.html y tema14_2.html.

Nombre

 tema14_1

 tema14_2

 tema14_paisaje

Y en nuestro archivo ya solamente tenemos que poner el nombre de la imagen:



```
<!-- LINKS RELATIVOS -->
<a href="tema14_1.html">Abrir Tema 14_1</a>
<hr>
<a href="tema14_2.html">Abrir Tema 14_2</a>
<hr>
<!-- USO RESPONSIVE DE IMÁGENES -->

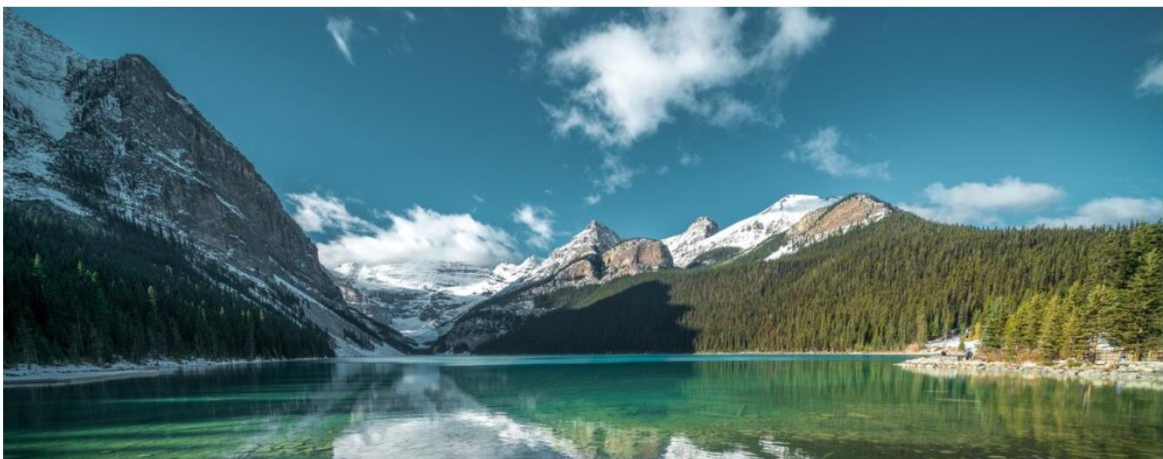
</body>

</html>
```

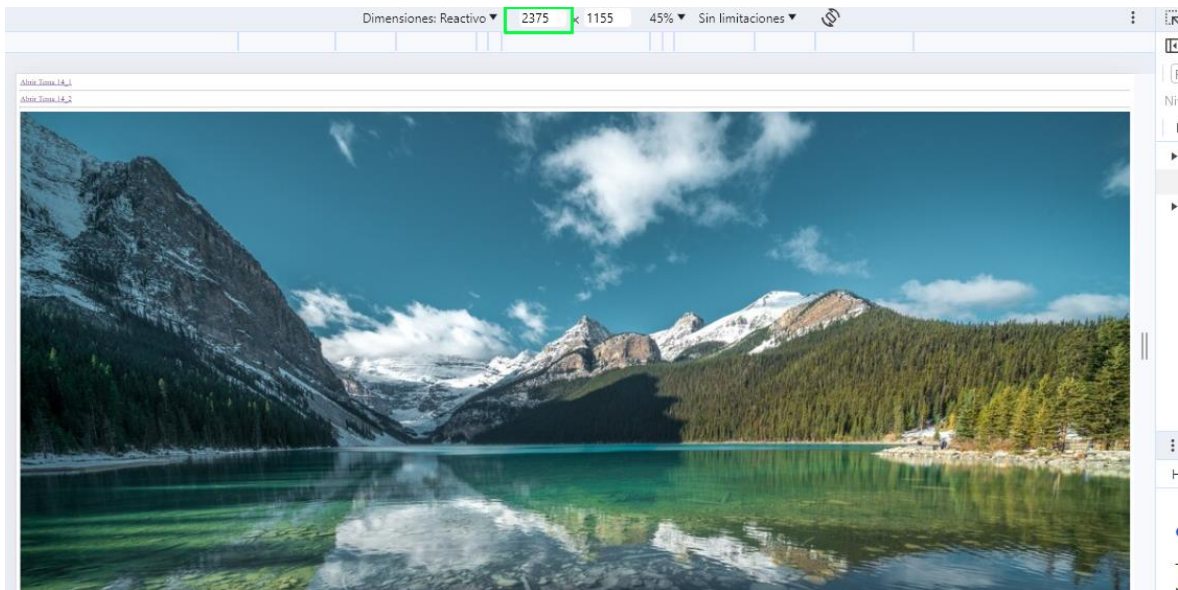
Si probamos ahora, debería de encontrar la imagen sin problema.

[Abrir Tema 14_1](#)

[Abrir Tema 14_2](#)



Si visualizamos el inspeccionador de elementos de Google, podemos ver que si movemos la imagen a más pequeña o más grande, siempre se adapta:



En este ejemplo, podemos observar que la imagen la podemos ampliar mucho más ancha de su tamaño original, en muchos casos, esto puede romper la calidad de la misma. Para poder arreglar esto, vamos a utilizar la propiedad: “max-width”.

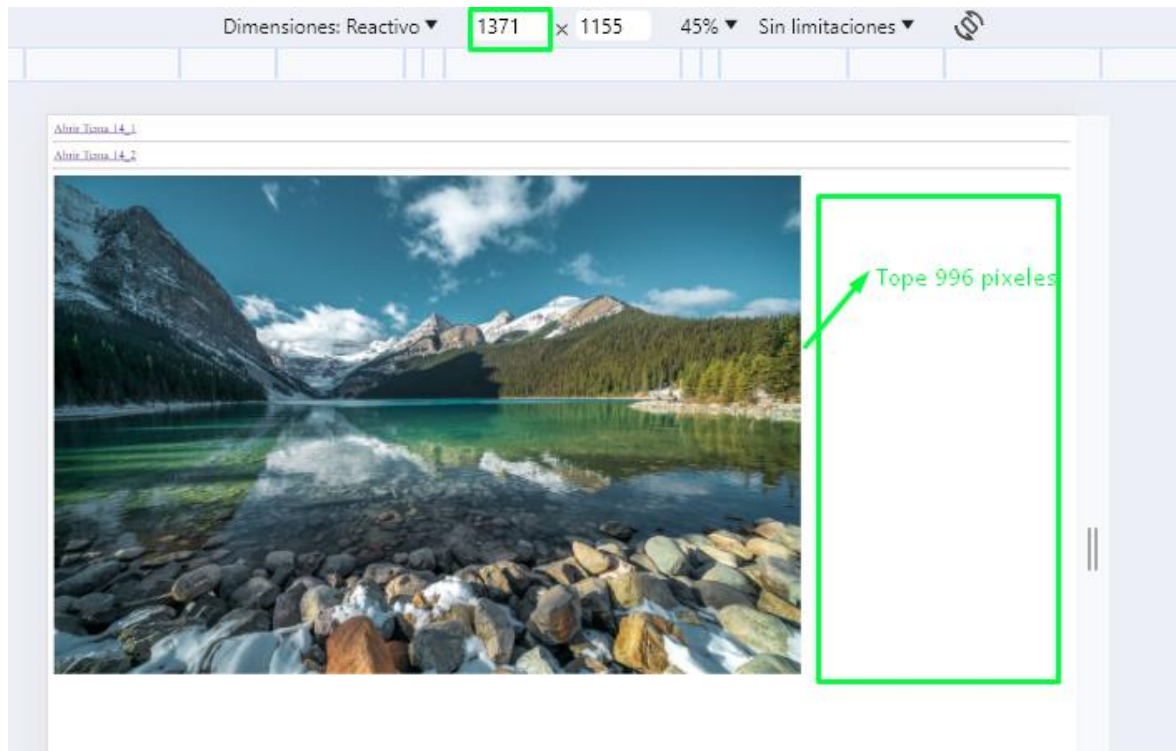
Si la propiedad max-width se establece en 100%, la imagen se reducirá si es necesario, pero nunca se ampliará para ser mayor que su tamaño original.

Pongamos ahora esta propiedad en nuestro ejercicio:

```
<hr>
<!-- USO RESPONSIVE DE IMÁGENES -->
<!---->
<!-- PROPIEDAD MAX-WIDTH PARA DELIMITAR EL ESCALADO HORIZONTAL DE UNA IMAGEN -->

</body>
```

Si probamos de nuevo con el inspeccionador, podremos observar que la imagen podemos escalarla como mucho a su ancho original que era de 996 píxeles.



Texto responsivo

Hasta ahora hemos visto que las medidas del tamaño de los textos las hemos establecido a pixeles: px.

```
<h1 style="font-size:10px">Hola página web</h1>
```

Pero si quisiéramos que el texto también se adaptara al ancho de la pantalla del dispositivo, podríamos poner la medida del texto en: vw (Viewport Width) Ancho del dispositivo.

```
<h1 style="font-size:10vw">Hola página web</h1>
```

Vamos a probar el siguiente ejercicio para que podáis observar la diferencia de estas medidas:



```
<!-- PROPIEDAD MAX-WIDTH PARA DELIMITAR EL ESCALADO HORIZONTAL DE UNA IMAGEN -->  
  
<!-- TEXTO EN PÍXELES -->  
<h1 style="font-size:16px;">TEXTO NORMAL EN PÍXELES</h1>  
<p style="font-size:12px;">Este párrafo está en tamaño de 12 píxeles.</p>  
<!-- TEXTO EN VIEWPORT WIDTH -->  
<h1 style="font-size:10vw;">TEXTO RESPONSIVE EN VIEWPORT WIDTH</h1>  
<p style="font-size:5vw;">Este párrafo está en responsive a 12 viewport width.</p>  
</body>
```

Si observamos nuestro archivo con el inspeccionador de Chrome abierto:





Media Queries

Además de cambiar el tamaño del texto y las imágenes, también es común utilizar “Media Queries” (Consulta de tamaños de dispositivos) en páginas web responsivas.

Con las consultas de tamaños (Media Queries) puedes definir estilos completamente diferentes para diferentes tamaños de navegador.

Vamos a ver un ejemplo en nuestro archivo html:

En el head de nuestro archivo vamos a poner un bloque de estilos <style>

```
<STYLE>
/* USO DE MEDIA QUERIE */
* {
  box-sizing: border-box;
}

.left {
  background-color: #2196F3;
  padding: 20px;
  float: left;
  width: 20%; /* Ancho a 20% por defecto */
}

.main {
  background-color: #f1f1f1;
  padding: 20px;
  float: left;
  width: 60%; /* Ancho a 60% por defecto */
}

.right {
  background-color: #04AA6D;
  padding: 20px;
  float: left;
  width: 20%; /* Ancho a 20% por defecto */
}
```



```
/* Uso de media query para añadir un cambio cuando se llega a 800px o
menos: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /*Pone en vertical las cajas cuando baja o llega a
los 800px la resolución de pantalla */
  }
}
</STYLE>
```

Podemos apreciar al principio del todo el selector “*” significa que coge el estilo para todos los elementos de nuestra página.

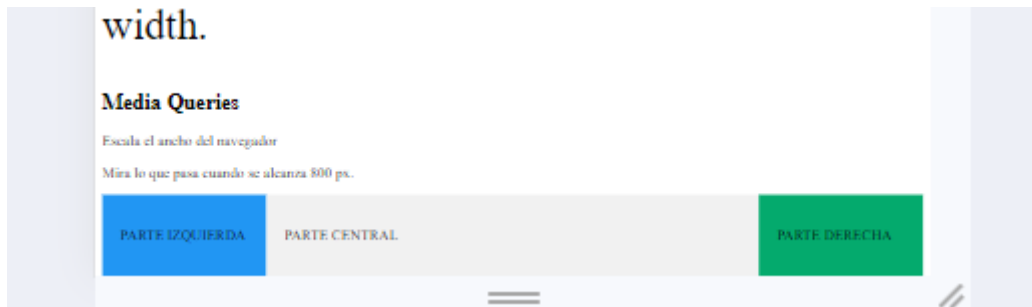
El `box-sizing: border-box;`

Lo que hace es contar como ancho de elementos también el padding en las cajas.

Y en el body de nuestro archivo añadimos:

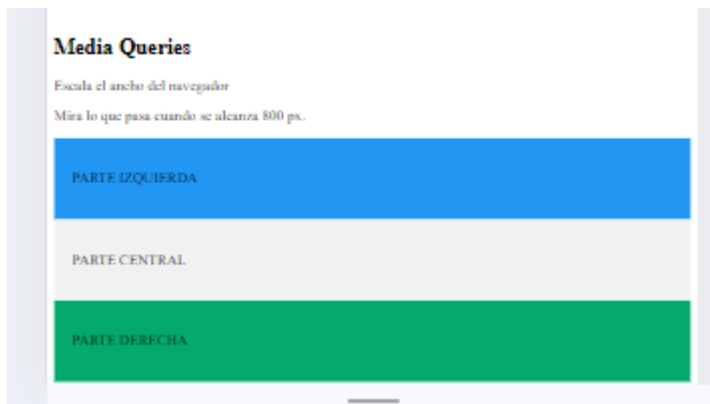
```
<h1 style="font-size:10vw;">TEXTO RESPONSIVE EN VIEWPORT WIDTH</h1>
<p style="font-size:5vw;">Este párrafo está en responsive a 12 viewport width.</p>
<!-- USO DE MEDIA QUERIE-->
<h2>Media Queries</h2>
<p>Escala el ancho del navegador</p>
<p>Mira lo que pasa cuando se alcanza 800 px.</p>
<hr>
<div class="left">
  <p>PARTE IZQUIERDA</p>
</div>
<div class="main">
  <p>PARTE CENTRAL</p>
</div>
<div class="right">
  <p>PARTE DERECHA</p>
</div>
</body>
```

Si observamos como aparece inicialmente:



Mantiene las medidas iniciales para cada estilo.

Pero si bajamos o llegamos a 800px:



Los coloca a 100% de ancho y aparecen en vertical.