



HTML

Imágenes

Para trabajar con imágenes en web, podemos utilizar la etiqueta de HTML ``. Es una etiqueta autocontenido y se utiliza para insertar imágenes en una página web. Nos permite mostrar gráficos, fotos y otros contenidos visuales en nuestros proyectos web. Esta etiqueta no tiene etiqueta de cierre, lo que significa que sólo se necesita una etiqueta de apertura para utilizarla correctamente.

Un ejemplo de código que incluye la etiqueta `img` podría ser este con los atributos `src`, `alt` y `title`.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>Trabajar con imágenes en Web</title>
  </head>
  <body>
    
  </body>
</html>
```

El atributo `src` se utiliza para especificar la ruta o la URL de la imagen que deseamos mostrar.

El atributo `alt` proporciona un texto alternativo para la imagen en caso de que la imagen no pueda cargarse o para ayudar a los usuarios con discapacidades visuales que utilicen lectores de pantalla.

Recordad que el texto alternativo debe ser descriptivo y proporcionar información relevante sobre el contenido de la imagen.

El atributo `title` nos proporciona información adicional sobre la imagen cuando el usuario coloca el cursor sobre ella.

Algo esencial para la accesibilidad web a nivel de imagen es que proporcionemos el texto alternativo a través del atributo `alt` ya que permite a las personas con discapacidades visuales entender el contenido de la imagen mediante el uso de tecnologías asistidas como lectores de pantalla.

Además, el texto alternativo también es muy útil cuando las imágenes no se pueden cargar debido a problemas de conexión a internet o restricciones de ancho de banda.

Formatos de imagen

Los formatos más populares para utilizar en web son jpeg, png, gif y svg.

- El formato **jpeg** es adecuado para imágenes con muchas variaciones de color y detalles, como podrían ser fotografías.
- El formato **png** es ideal para imágenes con color sólido como logotipos, ilustraciones y gráficos. Además, admite transparencia, lo que es útil cuando se necesitan superponer imágenes.
- El formato **gif** se utiliza principalmente para animaciones simples.
- El formato **svg** es un formato vectorial que utiliza descripciones matemáticas, coordenadas cartesianas con formas y colores para representar imágenes. A diferencia del resto de formatos anteriores, que son imágenes de mapa de bits, los **svg** son de resolución independiente y se puede escalar sin pérdida de calidad, así que es ideal para gráficos, iconos y logotipos.

Algo que es crucial para reducir el tipo de carga de la página y mejorar la experiencia de usuario es optimizar las imágenes.

Las imágenes no optimizadas pueden aumentar significativamente el tamaño de la página y ralentizar su carga.

Hay varias herramientas y técnicas para optimizar imágenes.

Podemos comprimir imágenes en formatos como jpg y png con herramientas como [tinypng](#) y [JPEGmini](#).

Podemos ajustar el tamaño de las imágenes al tamaño real que se mostrará en la página web reduciendo significativamente el tamaño del archivo, por ejemplo, si una imagen solo necesita ser mostrada a 300x200 no tiene por qué tener una resolución de 2000x1500 píxeles.

Además, podemos utilizar formatos de imagen más eficientes, como WebP, que ofrece una mejor compresión sin pérdida de calidad.

Podemos utilizar CSS para controlar la posición, el tamaño y el diseño de las imágenes en una página web.

Por ejemplo, podemos utilizar la propiedad `width` para establecer el ancho de una imagen y la propiedad `height` para establecer su altura.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>Trabajar con imágenes en Web</title>
  </head>
  <body>
```

```
  
</body>  
</html>
```

```
/* style.css */  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
img {  
    width: 200px;  
}
```

También podemos utilizar porcentajes para dimensionar una imagen en función de su contenedor.

Para ello necesitamos obviamente poner un contenedor en nuestra HTML.

```
<!DOCTYPE html>  
<html lang="es">  
    <head>  
        <meta charset="UTF-8" />  
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />  
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
        <link rel="stylesheet" href="style.css" />  
        <title>Trabajar con imágenes en Web</title>  
    </head>  
    <body>  
        <div class="container">  
              
        </div>  
    </body>  
</html>
```

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    width: 500px;
    background-color: aqua;
}

img {
    width: 50%;
}
```

Si quisieramos centrar esta imagen en este contenedor, podríamos utilizar la propiedad `margin` y ponerle un valor de `auto` para los lados izquierdo y derecho.

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    width: 500px;
    background-color: aqua;
}

img {
    width: 50%;
    display: block;
    margin-left: auto;
    margin-right: auto;
}
```

Esa es una de las formas de poder centrar una imagen.

También si queremos centrar una imagen verticalmente, podemos utilizar la técnica de posición absoluta y transformación siempre que el contenedor tenga una posición relativa.

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
```

```
}

.container {
    position: relative;
    height: 600px;
}

img {
    width: 50%;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

A las imágenes también podemos añadirle margenes y paddings.

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    position: relative;
    height: 600px;
}

img {
    width: 50%;
    margin: 20px;
    padding: 10px;
    border: 2px solid black;
    background-color: aqua;
}
```

Con esas técnicas básicas podemos controlar posición, tamaño y el diseño de las imágenes en nuestro proyecto web.

Hemos cargado la imagen desde HTML con el tag image, pero también tenemos la posibilidad de utilizar la propiedad `background-image` de CSS para aplicar una imagen de fondo a un elemento HTML. Para aplicar una imagen de fondo a un elemento debemos especificar la ruta de la imagen como valor de la propiedad `background image`.

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8" />
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="stylesheet" href="style.css" />
<title>Trabajar con imágenes en Web</title>
</head>
<body>
    <div class="container"></div>
</body>
</html>
```

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    background-image: url("../assets/demo-img.jpg");
}
```

Para ajustar el tamaño de la imagen de fondo podemos utilizar propiedades como `background-size` e indicar qué tamaño tiene que tener el contenedor.

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    background-image: url("../assets/demo-img.jpg");
    background-size: content;
    width: 400px;
    height: 300px;
}
```

Si nos fijamos estamos viendo que la imagen se repite, para evitar que se repita podemos utilizar la propiedad `background-repeat` y decirle que no se repita.

```
/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

```
.container {  
    background-image: url("../assets/demo-img.jpg");  
    background-size: content;  
    background-repeat: no-repeat;  
    width: 400px;  
    height: 300px;  
}
```

Otro de los valores que tiene la propiedad `background-size` es `cover` y lo que hace es ajustar la imagen al tamaño del contenedor con lo que ya no es necesario indicar que la imagen no se repita.

```
/* style.css */  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
.container {  
    background-image: url("../assets/demo-img.jpg");  
    background-size: cover;  
    width: 400px;  
    height: 300px;  
}
```

También disponemos de la propiedad `background-position` para posicionar la imagen en un determinado lugar dentro del contenedor, en este caso vamos a indicar que se centre.

```
/* style.css */  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
.container {  
    background-image: url("../assets/demo-img.jpg");  
    background-size: cover;  
    background-position: center;  
    width: 400px;  
    height: 300px;  
}
```

Es una buena práctica establecer un color de fondo para que el elemento se vea en caso de que la imagen no se cargue o no esté disponible en ese momento.

```

/* style.css */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.container {
    background-image: url("../assets/demo-img.jpg");
    background-size: cover;
    background-position: center;
    width: 400px;
    height: 300px;
    background-color: blue;
}

```

Al igual que en el diseño responsive debemos ajustar nuestras imágenes para que se vean y carguen de una manera optima en distintos tamaños de dispositivos. Para ello podemos utilizar la etiqueta `<picture>` la cual nos permite servir distintas versiones de una imagen en función del tamaño de pantalla por ejemplo. Utilizada en conjunto con `<picture>` está la etiqueta `<source>` en la que establecemos las condiciones mediante su atributo `media`, muy similar a los `media-queries`, para mostrar una determinada imagen en según que momento.

```

<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <link rel="stylesheet" href="style.css" />
        <title>Trabajar con imágenes en Web</title>
    </head>
    <body>
        <picture>
            <source media="(min-width: 1200px)" srcset="large.jpg" />
            <source media="(min-width: 800px)" srcset="medium.jpg" />
            
        </picture>
    </body>
</html>

```

También desde la propia etiqueta `` podemos proporcionar múltiples versiones de una misma imagen mediante las propiedades `srcset` y `sizes`.

```

<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8" />

```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="stylesheet" href="style.css" />
<title>Trabajar con imágenes en Web</title>
</head>
<body>

</body>
</html>
```

También es importante considerar la carga progresiva de imágenes, donde la imagen se carga gradualmente a medida que se muestra en la página. Esto evita que los usuarios tengan que esperar a que se cargue toda la imagen antes de ver algo.

En resumen, al trabajar con imágenes en la web, debemos utilizar la etiqueta `` de HTML para insertar las imágenes, proporcionar texto alternativo a través del atributo "alt" para la accesibilidad, elegir el formato adecuado según el tipo de imagen, optimizar las imágenes para reducir el tamaño de archivo y mejorar el rendimiento de la página, y considerar técnicas como el atributo "srcset" y la carga progresiva de imágenes para mejorar la experiencia del usuario.