Las imágenes representan una pieza fundamental al momento de mostrar contenido para web. Aquí conoceremos los principales tipos de imágenes web y sus formatos.

**Tipos de imágenes para web**

**Lossless (sin pérdida):**

* Capturan todos los datos del archivo original.
* No se pierde nada del archivo original.
* Puede comprimirse, pero podrá reconstruir su imagen al estado original

**Lossy (con pérdida):**

* Se aproximan a su imagen original.
* Podría reducir la cantidad de colores en su imagen o analizar la imagen en busca de datos innecesarios.
* Por consiguiente puede reducir su tamaño, lo que mejora el tiempo de carga de la página, pero pierde su calidad.
* Los archivos tipo lossy son mucho más livianos que los archivos tipo lossless, por lo que son ideales para usar en sitios en donde el tamaño del archivo y la velocidad de descarga son importantes.

**Formatos de imagen para web**

* **GIF** (Graphics Interchange Format): Formato de imagen sin pérdida, no se puede comprimir
* **PNG 8** (Portable Network Graphics): Formato de imagen sin pérdida, uso de colores de 256, se utiliza para logotipos e iconos para la página.
* **PNG 24** (Portable Network Graphics): Formato de imagen sin pérdida, utilización de colores ilimitados, alta calidad, si intentamos comprimir no ayudará demasiado por la gran cantidad de colores.
* **JPG / JPEG** (Photographic Experts Group): Formato de imagen con pérdida, perdemos calidad a la hora de comprimirlas, pero llegan a ser óptimas para la carga en la página web.
* **SVG - Vector** (Scalable Vector Graphics): Formato de imagen muy ligero sin pérdida, con svg no perdemos calidad, ya que está compuesta por vectores.
* **WebP**: Es un formato gráfico en forma de contenedor que sustenta tanto compresión con pérdida como sin ella. ​​Fue desarrollado por Google.

Las imágenes png-8 tienen fondo transparente, y las png24 también solo que esta tienen mas colores por ende es mas pesado.

SVG esta imagen no se ve pixelada al ampliar ya que hay un algoritmo que hace que la imagen cresca de acuerdo a la pantalla.

Tamaño máximo recomendado para una imágen:

* **70kb**  
  .  
  Herramientas para optimizar imágenes:
* [**Tiny PNG:**](https://tinypng.com/) Comprime el tamaño de una imagen, para hacerla más ligera.
* [**Verefix:**](https://www.verexif.com/) Elimina los metadatos de una imagen, para reducir su tamaño.

**RECOMENDACIÓN PERSONAL**  
Yo he usado estas páginas y no se logra optimizar a esta capacidad promedio de 70kb, pero usando Photoshop cambiando el ancho de la imagen y guardandolo como una imagen para web, he podido optimizar de 12MB a 92KB, que en mí opinión ha sido mejor que usar algún sitio en internet.



Optimizar nuestro sitio web con las imagenes o iframes, es colocando en la etiqueta img el atributo loading con el valor lazy.

<**img** loading="lazy" src="manzana.png" alt="People">

Clase 13

Una forma de optimizar nuestro sitio web con las imagenes o iframes, es colocando en la etiqueta img el atributo loading con el valor lazy.

<**img** loading="lazy" src="manzana.png" alt="People">

Atributo loading  
El atributo loading permite al navegador retrasar la carga de imáges y de iframes que están fuera de pantalla, hasta que el usuario haga scroll cerca de ellas. Éste atributo soporta 3 valores:

* lazy: Retrasa la carga de la imagen hasta que el usuario alcanza con el scroll una distancia calculada desde el viewport.
* eager: Carga la imagen inmediatamente, sin importar donde está situada o colocada en la pantalla. En resumen, no hace lazy-loading.
* auto: Implementa el comportamiento por defecto del navegador para la carga de las imágenes. En resumen, poner auto es lo mismo que no poner el atributo loading.

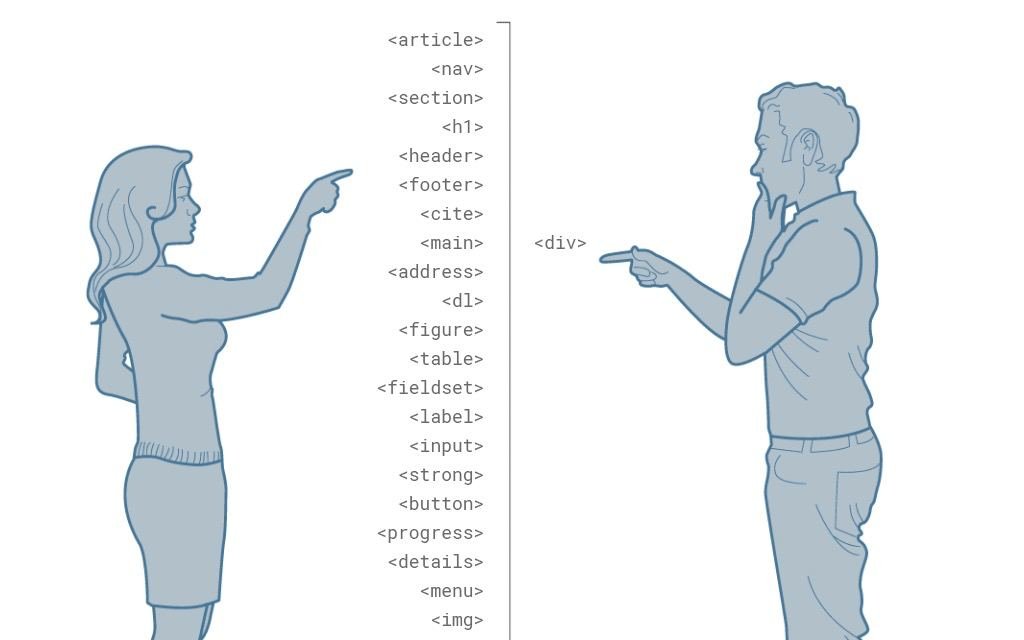
Clase 14

**Figure** <figure><img /> </figure> es una etiqueta que permite almacenar una imagen en su interior. Es una mejor práctica comparada con usar solamente un contenedor div. Como complemento al contenedor figure, se utiliza la etiqueta figcaption <figcaption></figcaption>, que permite darle una pequeña descripción a la imagen, como el autor, fuente o algo por el estilo, que se mostrará usualmente abajo de la imagen.

Figcaption se diferencia del atributo Alt porque esta última muestra su descripción en texto en el navegador solamente al pasar el mouse por encima de la imagen (de ahí su utilidad para personas con discapacidad visual).

Es importante considerar que la etiqueta figure no es únicamente para imágenes:  
[El elemento HTML <figure>](https://platzi.com/clases/1802-accesibilidad-web/26072-que-es-el-html-semantico-y-por-que-es-importante/) representa contenido independiente, a menudo con un título. Por lo general, se trata de una imagen, una ilustración, un diagrama, un fragmento de código, o un esquema al que se hace referencia en el texto principal, pero que se puede mover a otra página o a un apéndice sin que afecte al flujo principal.

## Ejemplo

clase 15 -etiqueta video

La etiqueta **<video>**, tiene algunos atributos como:  
.

1. **controls:** agrega al video los controles necesarios para reproducir, pausar y adelantar.
2. **preload = auto:** hace que el navegador descargue el video, en el momento en el que se acceda a la página.

.  
La etiqueta **<source>,** se puede colocar dentro de una etiqueta **<video>** varias veces, para especificar diferentes rutas. Esto para asegurar que cualquier navegador pueda mostrar el video.

Estructura

    <main>

        <section>

            <video src="./claseVideo/video.mp4#t=200,260" controls preload="auto"></video>

        </section>

    </main>

src="./claseVideo/video.mp4#t=200,260" al finalizar el src del video se agrega “#t=” para indicar q e video se reproduzca desde cierto minuto y también hasta que minuto finalice en este caso inicia en el minuto -200- y finaliza en el minuto-260-.

Se usa la etiqueta source dentro del tag video, para que se cargue el video con otro formato para que sea compatible con cualquier video.

<main>

        <section>

            <video controls preload="auto">

                <source src="./claseVideo/video.mp4#t=200,260" />

                <source src="./claseVideo/video.mp4#t=200,260" /> <!--video en otro formato-->

                <source src="./claseVideo/video.mp4#t=200,260" /> <!--video en otro formato-->

                <source src="./claseVideo/video.mp4#t=200,260" /><!--video en otro formato-->

            </video>

        </section>

    </main>

El navegador renderizará el primer video q pueda y no renderizará todos. Esto se hace por motivo de compatibilidad con otros navegadores y otros formatos.

Clase 16

Formularios---El mejor formulario es cuando no lo hay.

**Resumen**

**- <Form>; Es una etiqueta fundamentalmente para tener una mejor semántica.**

\*\*- <Label>; Etiqueta que representa la interfaz del usuario.

* for=" "; Permite asociar un control, en el se le dará un nombre personalizado
* **<Input>; Elemento donde el usuario da sus datos para el formulario.**
* type=“text”; Se aclara que los datos serán tipos textos (por ejemplo, nombre y apellido).
* type=“date”; Da una apariencia diferente al input más un calendario donde el usuario podrá poner una fecha mediante los números o facilitándose llenarlo con ayuda del calendario.
* type=“time”; También da una apariencia diferente pero para un horario.
* id=" "; Se le da un nombre (por así decirlo) al elemento para poder asociarlo junto al mismo nombre que el label en el que se le da al atributo de valor de for.
* ***<span>; Es para aplicar estilo al texto o agrupación de elementos. Lo que haría en si es que se colocará arriba o abajo (dependiendo donde lo pongas en el editor de código) donde obtendrá un texto.***

Estructura

<form action="">

    <label for="nombre">

        <span>Cual es tu nombre ?</span>

        <input type="text" id="nombre" placeholder="Tu nombre">

    </label>

    <label for="inicio-platzi">

        <span>Q dia inició en PLATZI ?</span>

        <input type="date" id="inicio-platzi">

    </label>

    <label for="horario">

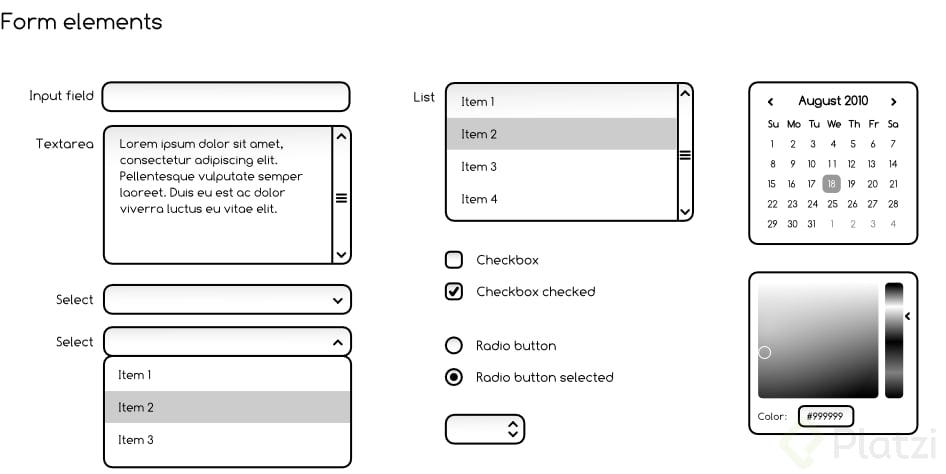
        <span>En que horario estudias ?</span>

        <input type="time" id="horario">

    </label>

</form>

Tipos de inputs



Como hacer formularios bonitos :

<https://medium.com/@juancaferraris/dise%C3%B1ando-formularios-m%C3%A1s-efectivos-estructura-inputs-labels-y-acciones-81ac011ea05f>

Calendar

 <form action="">

        <label for="hora">

            <span>Hora</span>

            <input type="time" id="hora" name="hora

            ">

        </label>

        <label for="dia">

            <span>Día</span>

            <input type="date" id="dia" name="dia

            ">

        </label>

        <label for="semana">

            <span>Week</span>

            <input type="week" id="semana" name="semana

            ">

        </label>

        <label for="mes">

            <span>Mes</span>

            <input type="month" id="mes" name="mes

            ">

        </label>

        <input type="submit">

    </form>

Se recomienda usar chrome ya que mozilla no renderiza correctamente los inputs month, week.

Calendario completo

<form action="">

    <label for="calendario">

        <span>calendario</span>

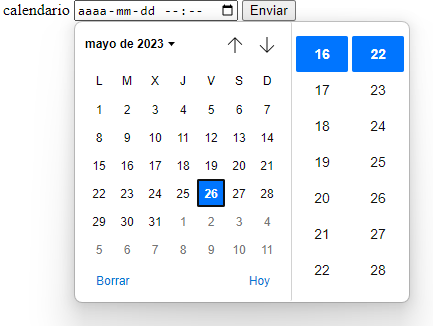
        <input type="datetime-local" id="calendario" name="calendario">

    </label>

    <input type="submit">

</form>

Así mismo se recomienda usarlo en Chrome en donde sale mas completo.



Tipos de inputs

<https://developer.mozilla.org/es/docs/Web/HTML/Element/input>