

### Concepto

- En el modelo de cajas se utilizan las propiedades **height** y **width** (autoexplicativas).
- Estas propiedades no incluyen paddings, borders o margins. Se refieren a la altura/anchura de lo que hemos llamado “*contenido*”. Para calcular el tamaño total de un elemento hay que sumar su relleno, bordes y márgenes.

Por ejemplo:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <title>Tamaño total</title>
7   <link rel="icon" type="image/png" href="../imagenes/favicon-16x16.png">
8   <style type="text/css">
9     p{width:320px;
10      padding:10px;
11      border: solid 5px tomato;}
12   </style>
13 </head>
14 <body>
15   <h2>Tamaño total</h2>
16   
17   <p>Párrafo inferior con anchura de 350px: width de 320 + 20px de padding +
18     10px de bordes</p>
19 </body>
20 </html>
```

Vista en el navegador:

### Tamaño total



Párrafo inferior con anchura de 350px: width de 320 + 20px de padding + 10px de bordes

- El valor de ambas propiedades se indica con cualquiera de las unidades de medida que hemos visto. Si no se indica ninguna medida, el tamaño es “auto”, es decir, el navegador le asigna el espacio necesario.
- No se aplican a los elementos en línea, salvo a las imágenes

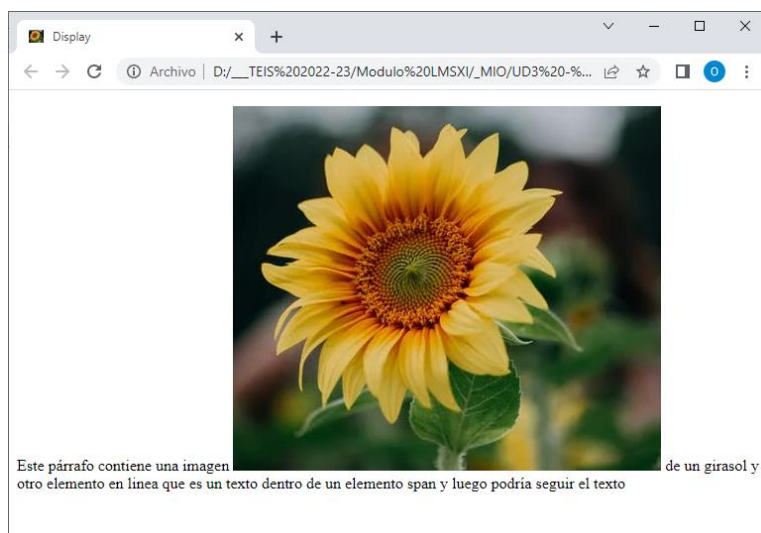
Por ejemplo:

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6     <title>Display</title>
7     <link rel="icon" type="image/png" href="../imagenes/favicon-16x16.png">
8     <style type="text/css">
9     </style>
10 </head>
11 <body>
12     <p>Este párrafo contiene una imagen  de
13     un girasol y otro elemento en línea <span> que es un texto dentro de un
14     elemento span</span> y luego podría seguir el texto</p>
15 </body>
16 </html>

```

Vista en el navegador:

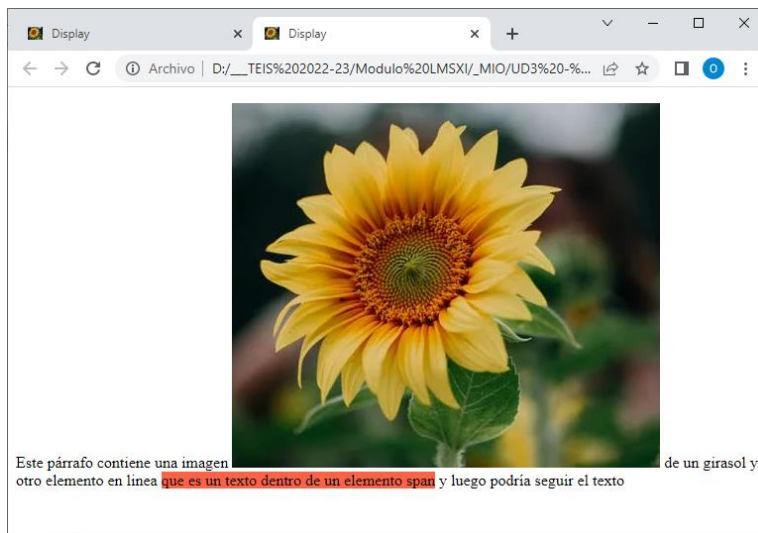


El elemento <span> es un elemento en línea y no se ve afectado aunque intentemos dimensionar su altura y/o anchura, como se ve en el código siguiente:

```

8     <style type="text/css">
9         span{background-color:tomato;
10             height:200px;
11             width:400px;}
12     </style>

```



Vemos que el elemento `<span>` sí que ha adquirido el color de fondo, pero su dimensión sigue siendo la automática.

### Propiedad max-width

- Se utiliza para establecer el ancho máximo de un elemento.
- Si usamos la propiedad **width** y la ventana del navegador se hace más pequeña que el ancho indicado para el elemento, el navegador añade una barra de scroll horizontal.
- En su lugar, la propiedad **max-width** mejora el manejo de ventanas pequeñas por parte del navegador.

Por ejemplo:

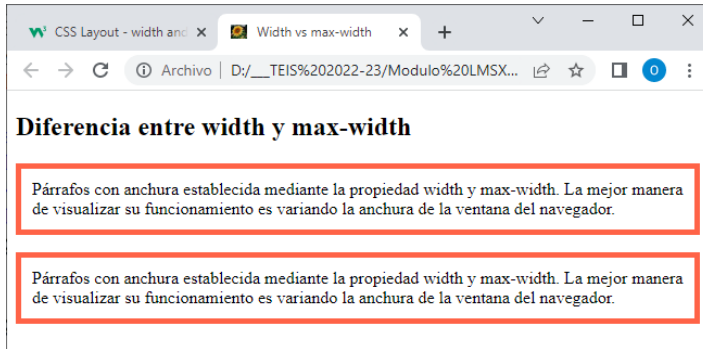
```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <title>Width vs max-width</title>
7   <link rel="icon" type="image/png" href="../imagenes/favicon-16x16.png">
8   <style type="text/css">
9     p.max{max-width:600px;
10      padding:10px;
11      border: solid 5px tomato;}
12     p.width{width:600px;
13      padding:10px;
14      border: solid 5px tomato;}
15   </style>
16 </head>
17 <body>
18   <h2>Diferencia entre width y max-width</h2>
19   <p class="width">Párrafos con anchura establecida mediante la propiedad width y max-width. La mejor manera de
20   visualizar su funcionamiento es variando la anchura de la ventana del navegador.</p>
21   <p class="max">Párrafos con anchura establecida mediante la propiedad width y max-width. La mejor manera de
22   visualizar su funcionamiento es variando la anchura de la ventana del navegador.</p>
23 </body>
24 </html>

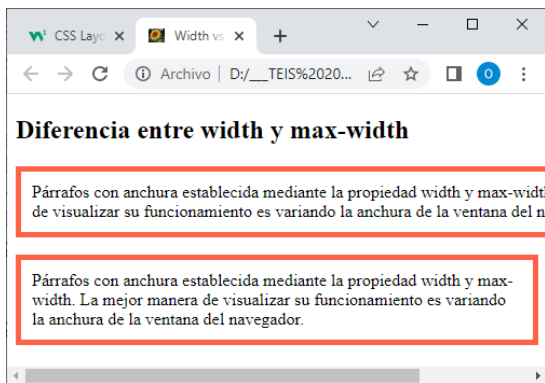
```

El primer párrafo está dimensionado con la propiedad **width** mientras que el segundo párrafo está dimensionado con la propiedad **max-width**.

Vista en el navegador cuando la anchura de la ventana es suficiente:



Vista en el navegador cuando la anchura de la ventana ya no es suficiente:



## PROPIEDAD display

(J. Eguíluz, pág. 97 y sgts)

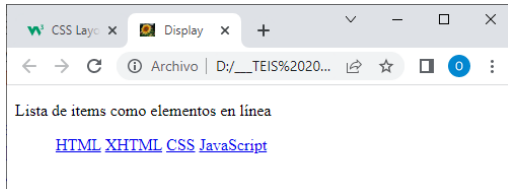
### Concepto

- La propiedad display (junto con la propiedad visibility) sirve para controlar la visualización de un elemento.
- Todo elemento (X)HTML tiene un valor defectivo para la propiedad display, dependiendo de qué tipo de elemento se trate. Los valores defectivos para la mayoría de los elementos son **block** (como para los elementos `<p>` o `<div>`) e **inline** (como para los elementos `<a>` o `<strong>`).
- Estos valores defectivos se pueden modificar; es decir, un elemento que por defecto es en línea puede mostrarse como si fuese un elemento en bloque y viceversa.
- **Importante:** cambiar el valor de la propiedad *display* de un elemento **sólo cambia la forma en que dicho elemento es visualizado**, pero no cambia el tipo del elemento en sí. Por ejemplo, un elemento en línea puede ser visualizado como un bloque, pero sigue siendo un elemento en línea y, por tanto, no podrá contener elementos en bloque en su interior.

- Ejemplo (muy utilizado para convertir una lista de enlaces en un menú horizontal)

```
<p>Lista de items como elementos en línea</p>
<ul>
  <li><a href="#">HTML</a></li>
  <li><a href="#">XHTML</a></li>
  <li><a href="#">CSS</a></li>
  <li><a href="#">JavaScript</a></li>
</ul>
```

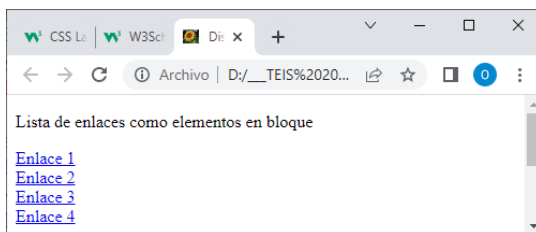
Mediante **li {display: inline;}**



- Ejemplo inverso: visualizar elementos en línea como si fuesen elementos en bloque

```
<p>Lista de enlaces como elementos en bloque</p>
<a href="#">Enlace 1</a>
<a href="#">Enlace 2</a>
<a href="#">Enlace 3</a>
<a href="#">Enlace 4</a>
```

Mediante **a {display: block;}**



- Otro valor de la propiedad display es **inline-block**

Es una mezcla de los dos anteriores:

- Comparado con *inline* permite establecer los valores de width/height para el elemento en cuestión (la mayoría de los elementos en línea no se ven afectados por estas propiedades).

También se respetan los márgenes superior e inferior (lo que no ocurre con display:inline)

- Comparado con *block*, la mayor diferencia es que no genera un salto de línea después del elemento y, por tanto, el elemento puede mantener su posición al lado de otros elementos

Ver ejemplo en [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_inline-block\\_span1](https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_span1)

- El último de los valores importantes de la propiedad display es **none**:

El valor **none** oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar dicho elemento.

***Este valor se utiliza habitualmente junto con JavaScript*** para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

- Ejemplo:

```
<p>Lista de enlaces como elementos en bloque</p>
<p class="oculto">Párrafo oculto</p>
<a href="#">Enlace 1</a>
<a href="#">Enlace 2</a>
<a href="#">Enlace 3</a>
<a href="#">Enlace 4</a>
```

Mediante **.oculto {display: none;}**

Lista de enlaces como elementos en bloque

[Enlace 1](#)  
[Enlace 2](#)  
[Enlace 3](#)  
[Enlace 4](#)

Enlace interesante: <https://javadesde0.com/diferencias-entre-display-block-inline-inline-block-y-none/>