

Hojas de estilos **CSS3**

Autor: Kostadin Kirilov
DAW2:Diseño de interfaces web

Hojas de estilos CSS3

1.- Hojas de estilos.....	5
1.1.- <i>Formas de aplicar estilos.....</i>	5
1.2.- <i>Prioridades a la hora de aplicar estilos.....</i>	7
1.3.- <i>Comentarios en CSS.....</i>	7
2.- Selectores.....	8
2.1.- <i>Selector de elemento.....</i>	8
2.2.- <i>Selector de id.....</i>	8
2.3.- <i>Selector de clase.....</i>	10
2.4.- <i>Selector de clase para un tipo de elemento específico.....</i>	11
2.5.- <i>Aplicar varias clases a un elemento.....</i>	12
2.6.- <i>Agrupar selectores.....</i>	13
3.- Colores.....	15
3.1.- <i>color.....</i>	15
3.2.- <i>Background-color.....</i>	16
4.- background-image.....	17
4.1.- <i>background-repeat -.....</i>	18
4.2.- <i>background-position.....</i>	20
4.3.- <i>background-attachment.....</i>	21
5.- border.....	22
5.1.- <i>Border-style.....</i>	22
5.2.- <i>border-width.....</i>	24
5.3.- <i>border color.....</i>	24
5.4.- <i>Estilos individuales para bordes.....</i>	25
5.5.- <i>Bordes formas de ahorrar código.....</i>	25
5.6.- <i>border-radius.....</i>	25
5.7.- <i>box-sizing.....</i>	26
6.- margin.....	27
6.1.- <i>Valor auto:.....</i>	29
6.2.- <i>Valor heredado:.....</i>	30
7.- padding.....	33
8.- height.....	36
8.1.- <i>max-height.....</i>	37
8.2.- <i>min-height.....</i>	38
9.- Modelo de cajas.....	39
9.1.- <i>outline.....</i>	40
10.- Trabajo con texto.....	41

<i>10.1.- Color del texto:</i>	41
<i>10.2.- Alineación del texto:</i>	41
<i>10.3.- Decoración del texto:</i>	42
<i>10.4.- Transformación del texto:</i>	43
<i>10.5.- Tabulado del texto:</i>	44
<i>10.6.- Espaciado entre letras:</i>	45
<i>10.7.- Altura de línea:</i>	45
<i>10.8.- Dirección del texto:</i>	46
<i>10.9.- Espaciado entre palabras:</i>	47
<i>10.10.- Sombreado del texto:</i>	47
11.- Fuentes.....	48
<i>11.1.- Repositorio de Google Fonts: https://fonts.google.com/.....</i>	48
<i>11.2.- Tamaño de fuente:</i>	52
<i>11.3.- Ancho de fuente:</i>	54
12.- Iconos.....	56
13.- Favicon.....	57
14.- Enlaces.....	58
<i>14.1.- Decoración del texto:</i>	60
<i>14.2.- Color de fondo de los enlaces:</i>	61
15.- Listas.....	62
<i>15.1.- Imagen como ícono de lista:</i>	65
<i>15.2.- Posición de los íconos de lista:</i>	65
<i>15.3.- Listas con colores:</i>	66
16.- Combinadores- Selectores.....	67
<i>16.1.- Selectores descendentes.....</i>	67
<i>16.2.- Selectores hijos.....</i>	68
<i>16.3.- Selectores adyacentes hermanos.....</i>	69
<i>16.4.- Selectores generales hermanos.....</i>	70
17.- Pseudoclases.....	71
<i>17.1.- pseudoclases-combinas con CSS.....</i>	72
<i>17.2.- Caso interesante:</i>	73
18.- Pseudoelementos.....	75
<i>18.1.- Pseudoelemento first-letter:</i>	75
<i>18.2.- Pseudoelemento before:</i>	76
<i>18.3.- Pseudoelemento after:</i>	76
<i>18.4.- Pseudoelemento selection:</i>	77
19.- Display.....	78

19.1.- Elementos de bloque:.....	80
19.2.- Elementos inline:.....	80
19.3.- Display: none.....	80
20.- Visibility.....	81
21.- Position.....	82
21.1.- Position static:.....	82
21.2.- Position relative:.....	83
21.3.- Position fixed:.....	84
21.4.- Position absolute:.....	85
21.5.- Position sticky:.....	86
21.6.- z-index.....	87
22.- Overflow.....	89
22.1.- Overflow:auto.....	91
23.- Float.....	92
24.- Clear.....	93
24.1.- Clearfix.....	94
25.- Alinear elementos.....	96
25.1.- Centrado vertical utilizando padding:.....	99

1.- Hojas de estilos

Definición: mediante una hoja de estilos se pueden definir todas las propiedades visuales del documento en general y de sus elementos en particular y aplicarlos a un documento HTML.

1.1.- Formas de aplicar estilos

1. Hoja de estilos interna.

- Sirve para definir estilos y clases en un documento .css
- Hoja de estilos interna La aplicamos con <style></style> dentro de la sección <head> de la web

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Pocos</title>
    <style>
      table, td, th{
        border: 1px solid black ;
        border-collapse: collapse;
      }
    </style>
  </head>
  <body>
```

2. Hoja de estilos externa.

- Hoja de estilos externa, se aplica con:
 - <link rel="stylesheet" type="text/css" href="HojaEstilos1.css">

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Ejercicio 1 Sesión 12</title>
  <link rel="stylesheet" href="css/HojaEstilos1.css">
</head>
<body>
```

3. Estilos en línea (utilizar con moderación).

- Sirve para definir estilos y clases en el mismo documento .html
- Estilos en línea Atención: utilizar con moderación.
- Lo aplicamos con <h1 style="color:red; margin-left:15px;">Esto es un encabezado</h1>

```
<body>
  <h1 style="background-color: red; color: white; margin-left: 15px;">Ejercicio 1 - Sesión 12</h1>
  <div class="contenedor">
```

Ejercicio 1 - Sesión 12

1.2.- Prioridades a la hora de aplicar estilos

1. Estilo en línea.
2. Hojas de estilos internas o externas.
3. Por defecto el navegador.

1.3.- Comentarios en CSS

- Se utilizan para comentar el código y hacerlo más fácil de entender.
- Ejemplo:
 - /*Esto es un comentario de una línea*/
 - /*Esto es un comentario
 - de varias líneas*/

```
4
5  /*Esto es un comentario de una línea*/
6  /*Esto es un comentario
7   de varias líneas*/
8
```

2.- Selectores

Sirven para localizar un elemento en el documento HTML.

2.1.- Selector de elemento

Se utiliza para seleccionar un elemento basándonos en su nombre de elemento. Mediante esto podemos aplicar un estilo concreto a todos los <p>.

Ejemplo:

```
p {
    color:orange;
    text-align:right;
}
```

En este caso aplicaríamos color naranja y alineación derecha a todos los párrafos del documento.

Ejercicio 1 - Sesión 12

Estos es un parrafo de color naranja centrado a la derecha

2.2.- Selector de id

- Se utiliza para seleccionar un elemento único mediante su **id** (identificador que no puede empezar por un número).
- para estilos particulares.
 - Ejemplo:

```
#texto1 {
    color:yellow;
    text-align:center;
}
```

```
#texto1 {
    color: yellow;
    text-align:center;
}
```

Ejercicio 1 - Sesión 12

Estos es un parrafo de color naranja centrado a la derecha

 Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt molestiae cum ea ratione! Minima aut quia non est animi expedita at tempore eveniet esse! Perferendis blanditiis voluptate maiores explicabo quae? Vel dolorem amet voluptatum possimus ullam quisquam, distinctio repellat nisi omnis in minus inventore provident adipisci, ipsam numquam natus commodi est et reprehenderit doloribus nostrum accusamus ea? Magnam, numquam vero?

En este caso aplicaríamos color amarillo y alineación centrada un elemento con el identificador texto1.

En <body> lo aplicaríamos de la siguiente manera:

```
<span id="text1">Texto con estilo particular</span>
```

2.3.- Selector de clase

- Se utilizar para seleccionar elementos que cumplan con un atributo de clase específico.

Ejemplo:

```
.izquierda{
```

```
    text-align:left;
```

```
}
```

The screenshot shows a code editor interface with two files open:

- EXPLORER**: Shows the project structure with files like "ejemplosMiniproyecto.html" and "ejemplosMiniproyecto.css".
- OPEN EDITORS**: Shows the content of the files:

 - ejemplosMiniproyecto.html**: An HTML document with a title, meta tags, and a paragraph with the class "izquierda".
 - ejemplosMiniproyecto.css**: A CSS file containing a single rule ".izquierda { text-align: left; }".

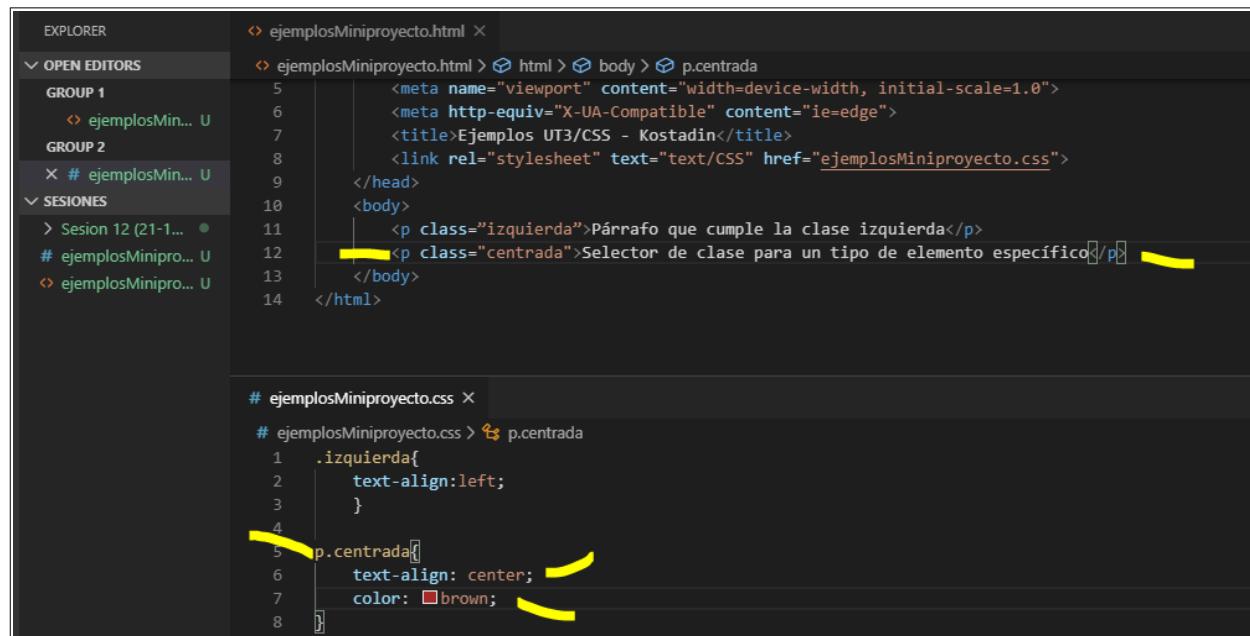
2.4.- Selector de clase para un tipo de elemento específico

Se utilizar para especificar unos determinados elementos que se vean afectados por una clase específica.

Ejemplo:

```
p.centrar {
    text-align: center;
    color: brown;
}
```

En este ejemplo únicamente los elementos <p> que cumplan la clase “centrar”, quedarían alienados de manera centrada.



The screenshot shows the Eclipse IDE interface. On the left, the 'EXPLORER' view lists several open editors and sessions. In the center, the code editor displays two files: 'ejemplosMiniproyecto.html' and '# ejemplosMiniproyecto.css'. The HTML file contains:

```
<p class="izquierda">Párrafo que cumple la clase izquierda</p>
<p class="centrada">Selector de clase para un tipo de elemento específico</p>
```

The CSS file contains:

```
.izquierda{
    text-align:left;
}
.p.centrada{
    text-align: center;
    color: brown;
```



2.5.- Aplicar varias clases a un elemento

Es posible aplicar varias clases a un elemento:

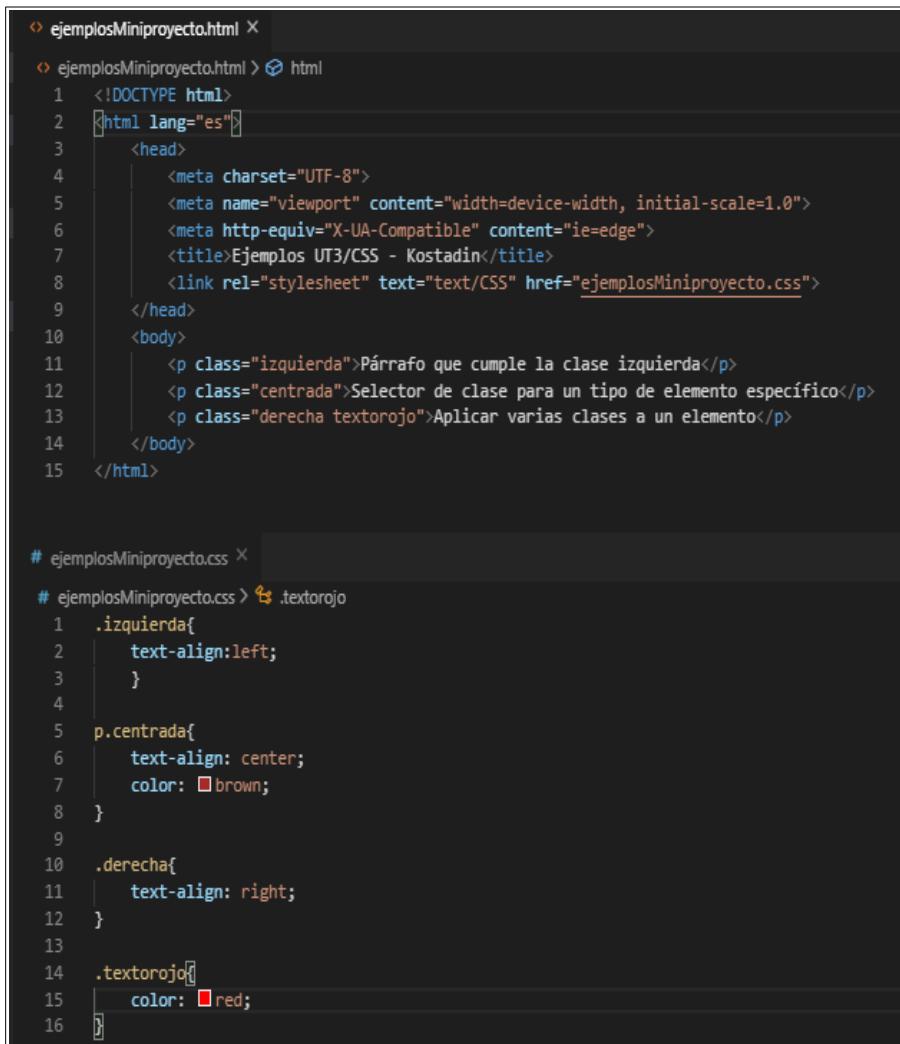
Ejemplo:

body:

```
<p class="derecha textorojo">Aplicar varias clases a un elemento</p>
```

css:

```
.derecha{  
    text-align: right;  
}  
  
.textorojo{  
    color: red;  
}
```



The screenshot shows a code editor with two files open. The top file is 'ejemplosMiniproyecto.html' containing the following HTML code:

```
<!DOCTYPE html>  
<html lang="es">  
    <head>  
        <meta charset="UTF-8">  
        <meta name="viewport" content="width=device-width, initial-scale=1.0">  
        <meta http-equiv="X-UA-Compatible" content="ie=edge">  
        <title>Ejemplos UT3/CSS - Kostadin</title>  
        <link rel="stylesheet" type="text/css" href="ejemplosMiniproyecto.css">  
    </head>  
    <body>  
        <p class="izquierda">Párrafo que cumple la clase izquierda</p>  
        <p class="centrada">Selector de clase para un tipo de elemento específico</p>  
        <p class="derecha textorojo">Aplicar varias clases a un elemento</p>  
    </body>  
</html>
```

The bottom file is 'ejemplosMiniproyecto.css' containing the following CSS code:

```
.izquierda{  
    text-align: left;  
}  
  
.centrada{  
    text-align: center;  
    color: brown;  
}  
  
.derecha{  
    text-align: right;  
}  
  
.textorojo{  
    color: red;  
}
```



2.6.- Agrupar selectores

En este ejemplo agruparíamos los selectores para los tipos de elemento h1, h2 y p, que tendrían color gris.

```
h1, h2, p {
    color:gray;
}
```

```

ejemplosMiniproyecto.html
ejemplosMiniproyecto.html > html > body > h1
3     <head>
4         <meta charset="UTF-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1.0">
6         <meta http-equiv="X-UA-Compatible" content="ie=edge">
7         <title>Ejemplos UT3/CSS - Kostadin</title>
8         <link rel="stylesheet" text="text/CSS" href="ejemplosMiniproyecto.css">
9     </head>
10    <body>
11        <h1>Ejercicios de selectores</h1>
12        <!-- <p class="izquierda">Párrafo que cumple la clase izquierda</p>
13        <p class="centrada">Selector de clase para un tipo de elemento específico</p>
14        <p class="derecha textorox">Aplicar varias clases a un elemento</p> -->
15        <h2>Agrupar selectores</h2>
16        <p>En este ejemplo agruparíamos los selectores para los tipos de elemento h1, h2 y p, que tendrían color gris.</p>
17    </body>
18 </html>
# ejemplosMiniproyecto.css
# ejemplosMiniproyecto.css > h1, h2, p
17
18     h1, h2, p {
19         color:gray;
20     }
21
22
23

```



3.- Colores

3.1.- color

Especifican un determinado nombre de color o un valor **RGB, HEX, HSL, RGBA o HSLA**

HTML soporta 140 nombres de colores:

https://www.w3schools.com/colors/colors_picker.asp?colorhex=7FFFD4

HTML Color Picker

[◀ Previous](#) [Next ▶](#)

Pick a Color:



Selected Color:

Black Text
Shadow
White Text
Shadow

Aquamarine
#7FFFD4
rgb(127, 255, 212)
hsl(160, 100%, 75%)

Lighter / Darker:

100%	#FFFFFF
95%	#E6FFFF
90%	#CCFFEE
85%	#B3FFEE
80%	#99FFDD
75%	#7FFFD4
70%	#66FFCC
65%	#4DFFC3
60%	#33FFBB
55%	#1AFFB2
50%	#00FFAA
45%	#00E699
40%	#00CC88
35%	#00B377
30%	#009966
25%	#008055
20%	#006644
15%	#004D33
10%	#003322
5%	#001A11
0%	#000000

Or Enter a Color:

Or Use HTML5:

Hue

Hue	Hex	Rgb	Hsl
0	#ff8080	rgb(255, 128, 128)	hsl(0, 100%, 75%)
15	#ff9f80	rgb(255, 159, 128)	hsl(15, 100%, 75%)
30	#ffbfb80	rgb(255, 191, 128)	hsl(30, 100%, 75%)
45	#ffdff80	rgb(255, 223, 128)	hsl(45, 100%, 75%)
60	#fffff80	rgb(255, 255, 128)	hsl(60, 100%, 75%)
75	#dfff80	rgb(223, 255, 128)	hsl(75, 100%, 75%)
90	#bfff80	rgb(191, 255, 128)	hsl(90, 100%, 75%)

3.2.- Background-color

Uso: aplicar un color de fondo a un elemento HTML

Ejemplo:

```
<p style="background-color:Tomato">
    Este párrafo tiene Tomato como color de fondo
</p>
```

```

EXPLORER          ejemplosMiniproyecto.html X
OPEN EDITORS      ejemplosMiniproyecto.html > html > body
GROUP 1           ejemplosMiniproyecto.html
GROUP 2           ejemplosMiniproyecto.html
SESIONES          Sesion 12 (21-1... ●)
# ejemplosMiniproyecto.html
ejemplosMiniproyecto.html
1   <head>
2     <meta charset="UTF-8">
3     <meta name="viewport" content="width=device-width, initial-scale=1.0">
4     <meta http-equiv="X-UA-Compatible" content="ie=edge">
5     <title>Ejemplos UT3/CSS - Kostadin</title>
6     <!-- <link rel="stylesheet" text="text/CSS" href="ejemplosMiniproyecto.css"> -->
7   </head>
8   <body>
9     <h1>Ejercicios de colores</h1>
10    <p style="background-color: Tomato">Este párrafo tiene Tomato como color de fondo</p>
11    <!-- <p class="izquierda">Párrafo que cumple la clase izquierda</p>
12    <p class="centrada">Selector de clase para un tipo de elemento específico</p>
13    <p class="derecha textorojito">Aplicar varias clases a un elemento</p> -->
14    <h2>Agrupar selectores:</h2>
15    <p>En este ejemplo agruparíamos los selectores para los tipos de elemento h1, h2 y p, que tendrían color gris.</p>
16
17
18

```



4.- background-image

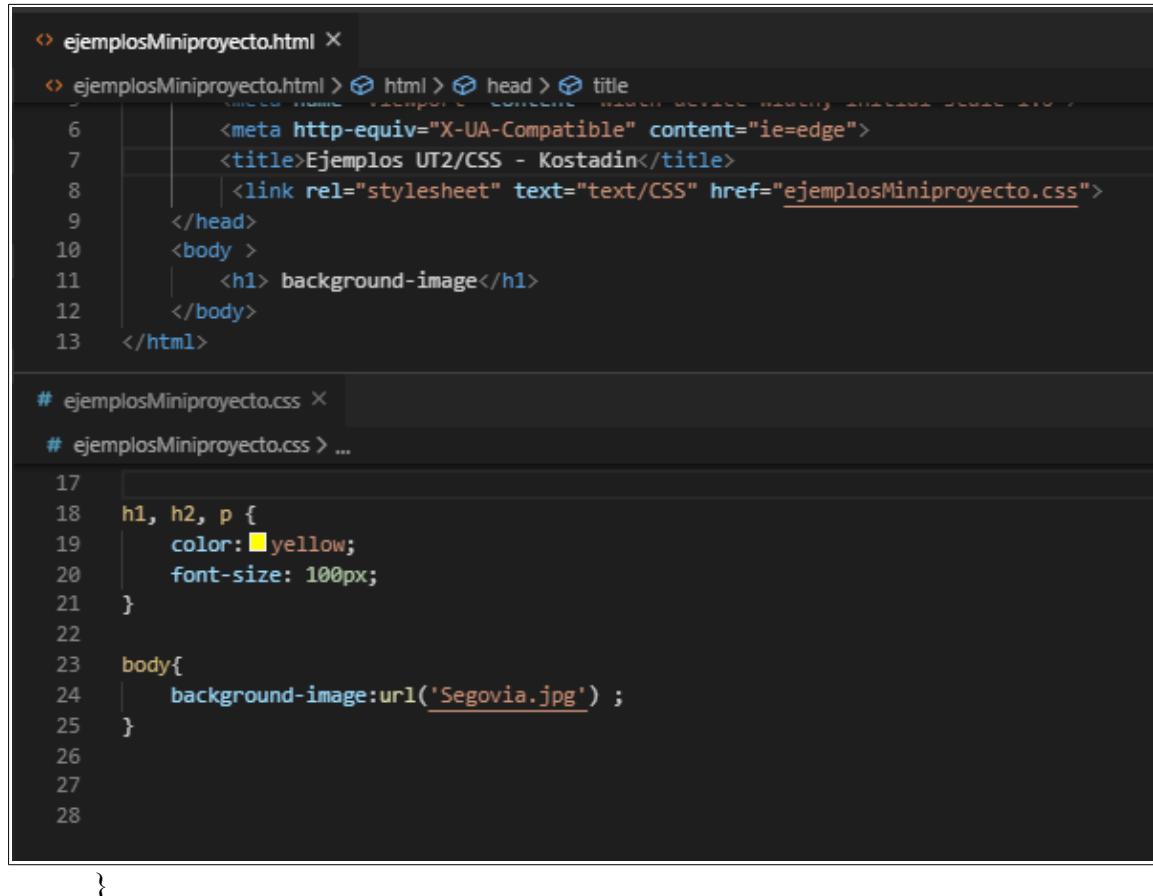
Uso: especifica una imagen de fondo para un elemento.

Ejemplo:

```
<body style="background-image:url('Img/Fondo1.jpg')">
```

Con selectores:

```
body {  
    background-image:url("Img/Halo1.jpg");
```



The screenshot shows a code editor with two files open:

- ejemplosMiniproyecto.html**: An HTML file containing the following code:

```
<!DOCTYPE html>  
<html>    <head>        <meta http-equiv="X-UA-Compatible" content="ie=edge">        <title>Ejemplos UT2/CSS - Kostadin</title>        <link rel="stylesheet" type="text/css" href="ejemplosMiniproyecto.css">    </head>    <body>        <h1> background-image</h1>    </body></html>
```
- ejemplosMiniproyecto.css**: A CSS file containing the following code:

```
h1, h2, p {    color: yellow;    font-size: 100px;}  
body{    background-image:url('Segovia.jpg') ;}
```



4.1.- background-repeat –

Se utiliza para indicar cómo debe repetirse la imagen de fondo

Ejemplos:

- `background-repeat:repeat-x; /*Para repetir la imagen de fondo horizontalmente*/`



Estilos con CSS

- background-repeat: repeat-y; /*Para repetir la imagen de fondo verticalmente*/



- background-repeat: no-repeat; /*Para no repetir la imagen de fondo*/



4.2.- background-position

Se utiliza para indicar la posición que deseamos que tenga la imagen de fondo.

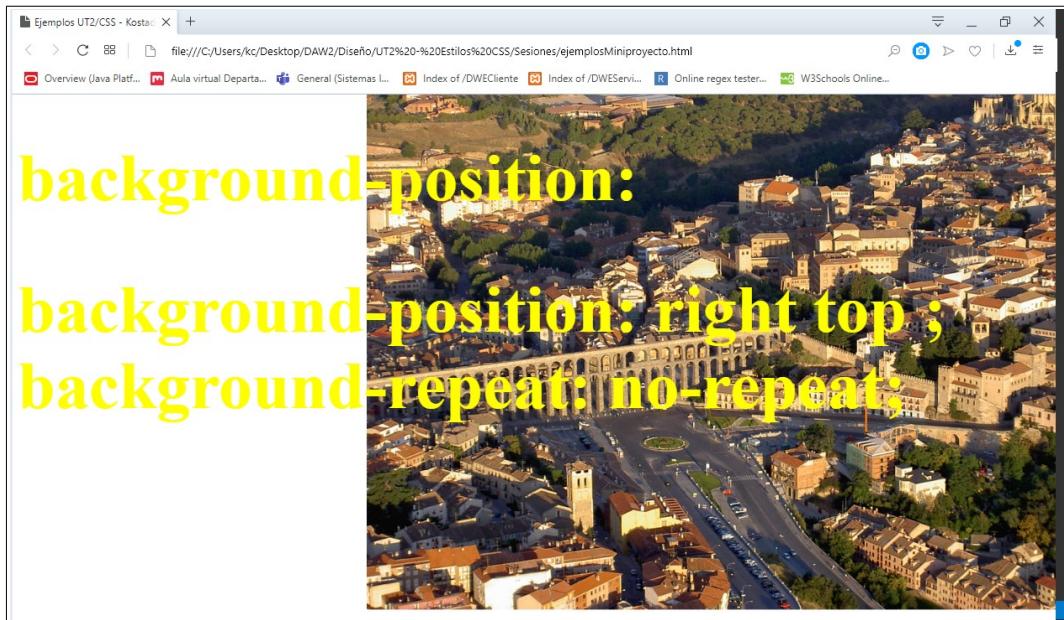
Ejemplos:

- background-position:right top;

```

↳ ejemplosMiniproyecto.html ×
ejemplosMiniproyecto.html > ⚡ html > ⚡ body > ⚡ h1
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Ejemplos UT2/CSS - Kostadin</title>
8   | <link rel="stylesheet" text="text/CSS" href="ejemplosMiniproyecto.css">
9   </head>
10  <body >
11  | <h1> background-position:[]</h1>
12  | <h2>background-position: right top ;
13  | | | background-repeat: no-repeat;</h2>
14  </body>
15 </html>

# ejemplosMiniproyecto.css ×
# ejemplosMiniproyecto.css > 📄 body
17
18 h1, h2, p {
19   color: yellow;
20   font-size: 100px;
21 }
22
23 body{
24   background-image:url('Segovia.jpg') ;
25   background-position: right top ;
26   background-repeat: no-repeat;
27 }
```



Posibles valores:

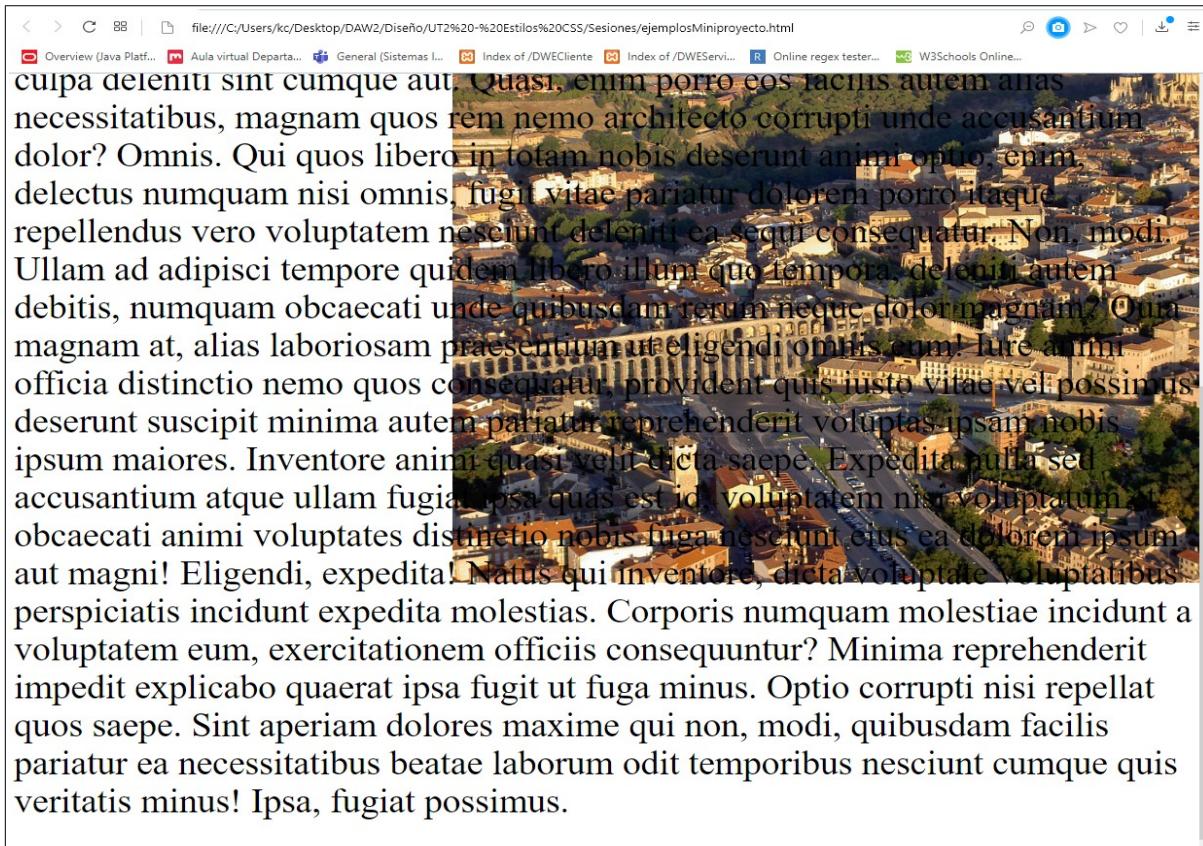
- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom

4.3.- **background-attachment**

Se utiliza para fija una imagen de fondo y que no se desplace con el scroll con el resto de la página.

La opción predeterminada es **scroll**.

Ejemplos:**background-attachment: fixed;**



5.- border

5.1.- Border-style

Uso: especifica un tipo de borde para un elemento en cuestión.

Ejemplo:

```
.claseborde1 {
    border-style:dashed;
}
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Ejemplos UT2/CSS - Kostadin</title>
    <link rel="stylesheet" text="text/CSS" href="ejemplosMiniproyecto.css">
</head>

<body>
    <h1>Bordes con CSS:</h1>
    <div class="contenedorGeneral bordesConCSS">
        <p>border-style:dashed;</p>
    </div>
</body>
```



Estilos con CSS

Posibles valores:

- none

border-style:dashed;

- hidden

border-style:dashed;

- dotted

border-style:dashed;

- dashed

border-style:dashed;

- solid

border-style:dashed;

- double

border-style:dashed;

- groove

border-style:dashed;

- ridge

border-style:dashed;

- inset

border-style:dashed;

- outset

border-style:dashed;

5.2.- border-width

Uso: indica el ancho del borde.

Ejemplo:

Border-width: 5px;

Border-width: 2px 10px 4px 20px;

(Arriba, derecha, abajo, izquierda)

Bordes con CSS:

Border-width: 2px 10px 4px
20px; /*(Arriba, derecha,
abajo, izquierda)*/ }

5.3.- border color

```

2 <body>
3   <h1>Bordes con CSS:</h1>
4   <div class="contenedorGeneral bordesConCSS anchoDeBorde">
5     <p>Border-width: 2px 10px 4px 20px; /*(Arriba, derecha, abajo, izquierda)*</p>
6   </div>
7 </body>
8
9 </html>

```

```

ejemplosMiniproyecto.css ×
ejemplosMiniproyecto.css > ⚒ .bordesConCSS
5
7 .bordesConCSS{
8   border-style:solid;
9   border-color: #chartreuse;
0 }

.anchoDeBorde{
3   Border-width: 2px 10px 4px 20px; /*(Arriba, derecha, abajo, izquierda)*/ 
4 }

```

Bordes con CSS:

Border-width: 2px 10px 4px
20px; /*(Arriba, derecha,
abajo, izquierda)*/

5.4.- Estilos individuales para bordes

```
border-top-style: dotted; /*Borde superior*/  
border-right-style: solid; /*Borde derecho*/  
border-bottom-style: dotted; /*Borde inferior*/  
border-left-style: solid; /*Borde izquierdo*/
```

Bordes con CSS:

Border-width: 2px 10px 4px
20px; /*(Arriba, derecha,
abajo, izquierda)*/

5.5.- Bordes formas de ahorrar código

De esta forma podemos especificar el ancho, estilo y color del borde en una sola línea.

Ejemplos:

```
border: 4px solid red;  
border-left: 4px solid red;
```

border:4px solid red;

5.6.- border-radius

Sirve para crear bordes redondeados. Cuanto mayor sea el valor, más redondeado será el borde.

Ejemplo:

```
border-radius: 50%;
```

border:4px solid red;

5.7.- box-sizing

Esta propiedad hace que el ancho del borde esté incluido en el ancho del elemento que tiene ese borde.

Ejemplo:

box-sizing: border-box

Realizar una práctica con 3 capas en paralelo: 1 con ancho 20%, otra con ancho 60% y otra con ancho 20%.

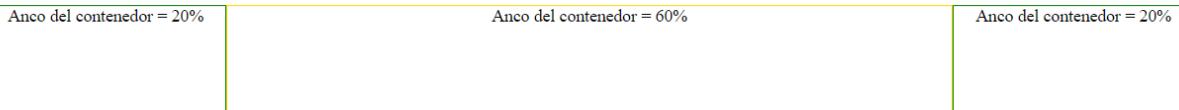
```

12  <body>
13      <h1>Bordes con CSS:</h1>
14          <div class="contenedor contenedorDe20">Anco del contenedor = 20%</div>
15          <div class="contenedor contenedorDe60">Anco del contenedor = 60%</div>
16          <div class="contenedor contenedorDe20">Anco del contenedor = 20%</div>
17  </body>
18
19  </html>

# ejemplosMiniproyecto.css ×

# ejemplosMiniproyecto.css > 📄 .contenedor
+++
12  .contenedorDe20{
13      float: left;
14      width: 20%;
15      border: solid 1px □green;
16      height: 100px;
17      box-sizing: border-box;
18 }
19 .contenedorDe60{
20     float: left;
21     width: 60%;
22     border: solid 1px □gold;
23     height: 100px;
24 }
```

Bordes con CSS:



6.- margin

Uso: permite especificar un margen en cada lado de los elementos.

Ejemplos:

```
#margenes1{  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

```
# ejemplosMiniproyecto.css > ...  
1   .margenes1{  
2       margin-top: 50px;  
3       margin-bottom: 75px;  
4       margin-right: 200px;  
5       margin-left: 200px;  
6   }  
7       border: solid 2px red;  
8 }
```

Margenes con CSS:

Latin text (lorem ipsum) describing various legal terms and conditions related to the use of margins in CSS.

```
#margenes1{ margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; }
```

Latin text (lorem ipsum) describing various legal terms and conditions related to the use of margins in CSS.

Se puede ahorrar código aplicando valores a la propiedad margin.

Ejemplo:

```
.margenes1{  
margin: 25px 50px 75px 100px; /* top, right, bottom, left*/  
}
```

```
# ejemplosMiniproyecto.css X  
# ejemplosMiniproyecto.css > .margenes1  
1   .margenes1{  
2     margin: 25px 50px 75px 100px; /* top, right, bottom, left*/  
3     border: solid 2px red;  
4   }  
5
```

quaerat corporis ipsa doloremque incidentum cum perferendis repudiandae aliquam, dicta sit ut consectetur aperiam ex vel assumenda!

```
margin: 25px 50px 75px 100px; /* top, right, bottom, left*/
```

Quaerat corporis ipsa doloremque incidentum cum perferendis repudiandae aliquam, dicta sit ut consectetur aperiam ex vel assumenda!

```
.margenes3{  
margin: 25px 50px; /*top y bottom 25px, right y left 50px*/
```

```
.margenes4{  
margin: 30px; /*Todos los márgenes 30px*/  
}
```

```
# ejemplosMiniproyecto.css X  
# ejemplosMiniproyecto.css > .margenes1  
1   .margenes1{  
2     margin: 25px 50px; /*top y bottom 25px, right y left 50px*/  
3     border: solid 2px red;  
4   }  
5
```

quaerat corporis ipsa doloremque incidentum cum perferendis repudiandae aliquam, dicta sit ut consectetur aperiam ex vel assumenda!

```
margin: 25px 50px; /*top y bottom 25px, right y left 50px*/
```

Quaerat corporis ipsa doloremque incidentum cum perferendis repudiandae aliquam, dicta sit ut consectetur aperiam ex vel assumenda!

6.1.- Valor auto:

Centra horizontalmente un elemento dentro de su contenedor. Sólo centra los elementos que se visualizan como bloque (propiedad display:block;)

```
#capa1 {  
    width: 200px;  
    margin: auto;  
    border: 3px dashed green;  
}
```

```
ejemplosMiniproyecto.html ×  
ejemplosMiniproyecto.html > html > body > div.contenedor > p.capa1  
12 <body>  
13     <h1>Margenes con CSS:</h1>  
14     <div class="contenedor">  
15         <p class="capa1">  
16             Lorem ipsum, dolor sit amet consectetur adipisicing elit. Illum distinctio ullam, corporis natus expedita  
17             eligendi accusamus repellat recusandae. Atque voluptatibus magnam et, ea ut architecto neque consequuntur  
18             asperiores ullam sequi?  
19         </p>  
20     </div>  
21 </body>  
22 </html>  
  
ejemplosMiniproyecto.css ×  
ejemplosMiniproyecto.css > .capa1  
1 .contenedor{  
2     width: 100%;  
3     min-height: 250px;  
4     border: 1px solid black;  
5     display: block;  
6 }  
7 .capa1{  
8     min-height: 150px;  
9     width: 200px;  
10    margin: auto;  
11    border: 3px dashed green;  
12    display: block;  
13 }  
14 }
```



6.2.- Valor heredado:

Permite heredar una propiedad del elemento padre.

Ejemplo:

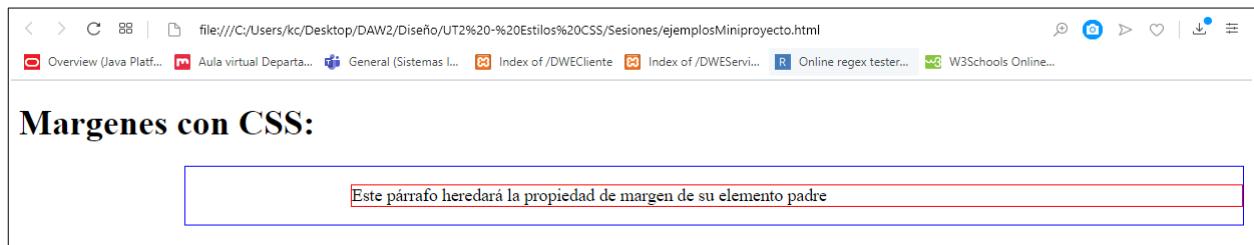
```
div {
    border: 1px solid blue;
    margin-left: 150px;
}
```

```
p.clasemargen1 {
    margin-left: inherit;
}
```

```
<div>
<p class="clasemargen1">Este párrafo heredará la propiedad de margen de su elemento padre</p>
</div>
```

```
ejemplosMiniproyecto.html
ejemplosMiniproyecto.html > html > body > div
12 <body>
13     <h1>Margenes con CSS:</h1>
14     <div>
15         <p class="clasemargen1">Este párrafo heredará la propiedad de margen de su elemento padre</p>
16     </div>
17 </body>
18
19 </html>

# ejemplosMiniproyecto.css
# ejemplosMiniproyecto.css > p.clasemargen1
1 div {
2     border: 1px solid blue;
3     margin-left: 150px;
4 }
5
6 p.clasemargen1 {
7     margin-left: inherit;
8     border: 1px solid red;
9 }
```



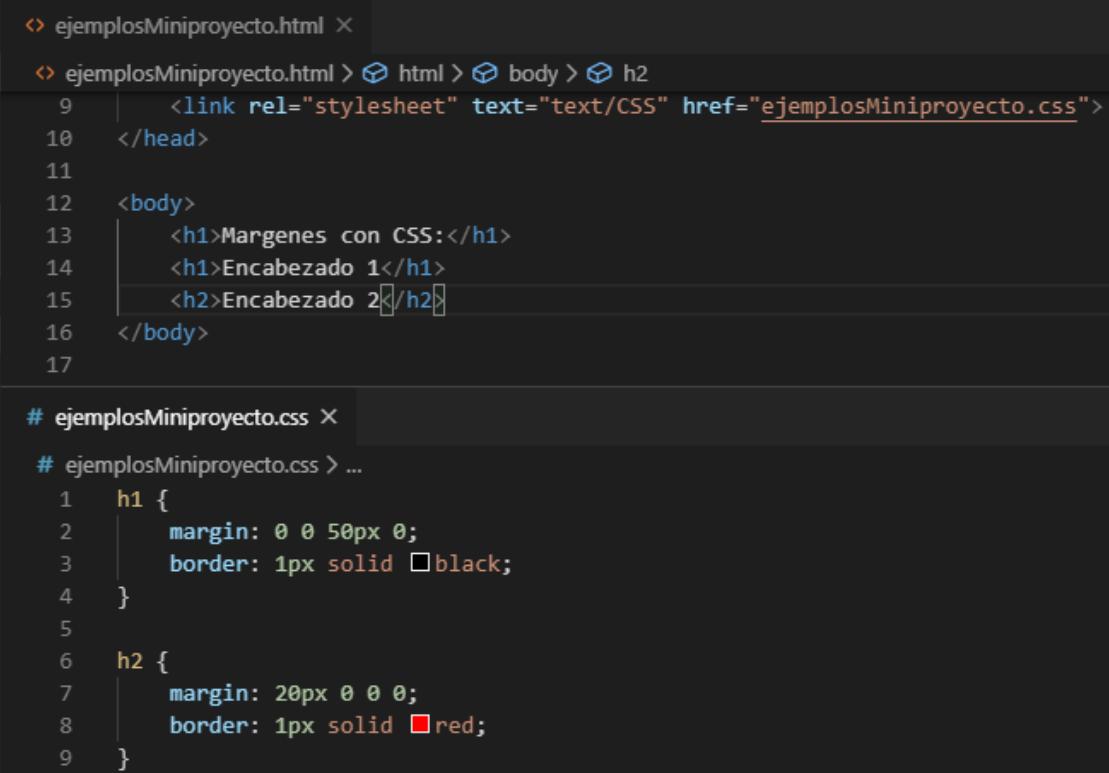
Estilos con CSS

Suma de márgenes (margin collapse):

```
h1 {  
    margin: 0 0 50px 0;  
}  
  
h2 {  
    margin: 20px 0 0 0;  
}
```

```
<h1>Encabezado 1</h1>  
<h2>Encabezado 2</h2>
```

Entre los dos elementos tendremos 50px de separación: 50 bottom margin y 20 top margin. Es decir, tomaría el mayor de los dos valores.



The screenshot shows a code editor with two files:

- ejemplosMiniproyecto.html**:
A snippet of HTML code. Lines 13 and 14 show an

element with the text "Margenes con CSS:" followed by an element with the text "Encabezado 1". Line 15 shows another element with the text "Encabezado 2".
- ejemplosMiniproyecto.css**:
A snippet of CSS code defining styles for

and elements. The style has a bottom margin of 50px and a black border. The style has a top margin of 20px and a red border.

6.-margin



7.- padding

Permite añadir espacio en el interior de un elemento con sus respectivos elementos hijos.

Ejemplo:

```
.clasecapa1{  
    padding-top: 50px;  
    padding-right: 30px;  
    padding-bottom: 50px;  
    padding-left: 80px;  
}
```

```
<div class="clasecapa1">
```

```
    Esta capa tendrá definidas varias propiedades padding  
</div>
```

```
↳ ejemplosMiniproyecto.html ×  
↳ ejemplosMiniproyecto.html > ⚒ html > ⚒ body > ⚒ div.clasecapa1  
10   </head>  
11  
12   <body>  
13       <h1>padding con CSS:</h1>  
14       <div class="clasecapa1">  
15           Esta capa tendrá definidas varias propiedades padding  
16       </div>  
17  
18   </html>  
  
# ejemplosMiniproyecto.css ×  
# ejemplosMiniproyecto.css > ⚒ .clasecapa1  
1   .clasecapa1{  
2       padding-top: 50px;  
3       padding-right: 30px;  
4       padding-bottom: 50px;  
5       padding-left: 80px;  
6       border: 1px solid □ black;  
7   }
```



Se puede ahorrar código aplicando valores a la propiedad padding.

```
padding: 25px 50px 75px 100px; /* top, right, bottom, left*/
```

Importante: el padding se suma al ancho de un elemento

```
#capapadding1 {  
    width: 300px;  
    padding: 25px; /*Izquierda y derecha*/  
}
```

The screenshot shows a code editor with two files: 'ejemplosMiniproyecto.html' and 'ejemplosMiniproyecto.css'.

ejemplosMiniproyecto.html:

```
<html>  
    <head>  
    </head>  
    <body>  
        <h1>padding con CSS:</h1>  
        <div class="capapadding1">  
            Importante: el padding se suma al ancho de un elemento  
        </div>  
    </body>  
</html>
```

ejemplosMiniproyecto.css:

```
.capapadding1 {  
    width: 300px;  
    padding: 25px; /*Izquierda y derecha*/  
    border: solid 1px black;
```

Tendremos una capa de 350px de anchura

Ver demostración: capa de 300px de ancho vs capa de 300px de ancho con padding:25px.

```
↳ ejemplosMiniproyecto.html ×
↳ ejemplosMiniproyecto.html > ⏷ html > ⏷ body > ⏷ div > ⏷ div.capapadding
13   <h1>padding con CSS:</h1>
14   <div>
15     <div class="capapadding">
16       |   anchura de la caja es de 300px sin padding
17     </div>
18     <div class="capapadding1">
19       |   caja de 300 px + padding left y right de 25px=>ancho de la caja 350px;
20     </div>
21   </div>

# ejemplosMiniproyecto.css ×
# ejemplosMiniproyecto.css > ⏷ .capapadding
1  .capapadding [
2    width: 300px;
3    border: solid 1px □black;
4  ]
5
6  .capapadding1 {
7    width: 300px;
8    padding: 25px; /*Izquierda y derecha*/
9    border: solid 1px □black;
10 }
```

padding con CSS:

anchura de la caja es de 300px sin padding

caja de 300 px + padding left y right de
25px=>ancho de la caja 350px;

8.- height

Permite establecer la altura de un elemento.

Ejemplo:

```
#capa1 {  
    height: 200px;  
    width: 30%;  
    background-color: blue;  
}
```

```
12 <body>  
13     <h1>padding con CSS:</h1>  
14     <div class="capa1">  
15     </div>  
16 </html>  
  
ejemplosMiniproyecto.css ×  
# ejemplosMiniproyecto.css > .capa1  
1 .capa1{  
2     height: 200px;  
3     width: 30%;  
4     background-color: blue;  
5     text-align: center;  
6 }
```

padding con CSS:



8.1.- **max-height**

Establecer una altura máxima con max-height. Si el contenido del elemento supera este máximo, el elemento activará la propiedad overflow (ver ejemplo).

Por defecto tenemos max-height: none;

Ejemplo:

```
p.parr1 {  
    max-height: 50px;  
}
```

The screenshot shows the browser's developer tools with the DOM tree and the CSS styles panel. The DOM tree shows the structure: ejemplosMiniproyecto.html > html > body > div.capa1 > p.parr1. The CSS styles panel shows the following rules:

```
# ejemplosMiniproyecto.css  
# ejemplosMiniproyecto.css > p.parr1  
1 .capa1{  
2     height: 200px;  
3     width: 30%;  
4     background-color: blue;  
5     text-align: center;  
6 }  
8 p.parr1[]  
9     max-height: 50px;  
10 }
```

padding con CSS:

Loreum ipsum dolor sit amet, consectetur adipisicing elit. Nostrum sed odio alias recusandae labore nulla, earum omnis autem, ratione adipisci in placeat repudiandae error magnam iste facere tenetur sequi esse. Nobis dolor adipisci laboriosam necessitatibus saepe amet sint recusandae voluptates beatae facere in repudiandae, vero minima nemo eaque atque incident quasi error obcaecati asperiores! Ratione quasi eum expedita facilis aut. Quisquam consectetur preferendis totam nostrum dicta corporis modi aut ab exercitationem praesentium distinctio similiqe ratione, non, inventore labore commodi vero animi expedita dolores, perspiciatis ad dolorem neque dolorum! Dolorem, dolores? Lorem ipsum dolor sit amet consectetur, adipisicing elit. Saepe nulla explicabo, harum repellat hic consequuntur ipsa quo quaerat atque nam vel voluptas neque, eveniet nihil alias! Rem odio eveniet consequatur.

8.2.- min-height

Establecer una altura mínima con min-height.

Ejemplo:

```
p.parr2{
    min-height: 50px;
}
```

```

<html>
<head>
<title>ejemplosMiniproyecto.html</title>
</head>
<body>
    <h1>padding con CSS:</h1>
    <div class="capa1">
        <p class="parr2">
            Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nostrum sed odio alias recusandae labore nulla, earum omnis autem, ratione adipisci in placeat repudiandae error magnam iste facere tenetur sequi esse. Nobis dolor adipisci laboriosam necessitatibus saepe amet sint recusandae voluptates beatae facere in repudiandae, vero minima nemo eaque atque incidunt quasi error obcaecati asperiores! Ratione quasi eum expedite facilis aut. Quisquam consectetur perferendis totam nostrum dicta corporis modi aut ab exercitationem praesentium distinctio similiqe ratione, non, inventore labore commodi vero animi expedita dolores, perspiciatis ad dolorem neque dolorum! Dolorem, dolores?
        </p>
    </div>
</body>
</html>

```

```

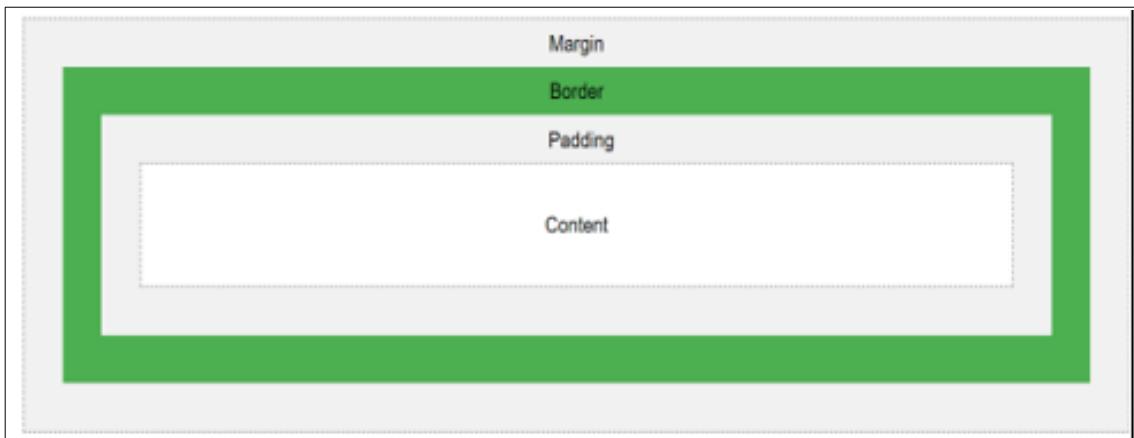
# ejemplosMiniproyecto.css
.capa1{
    width: 50%;
    background-color: blue;
    text-align: center;
    border: solid 1px red;
}
.parr2{
    min-height: 50px;
}

```

padding con CSS:

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nostrum sed odio alias recusandae labore nulla, earum omnis autem, ratione adipisci in placeat repudiandae error magnam iste facere tenetur sequi esse. Nobis dolor adipisci laboriosam necessitatibus saepe amet sint recusandae voluptates beatae facere in repudiandae, vero minima nemo eaque atque incidunt quasi error obcaecati asperiores! Ratione quasi eum expedite facilis aut. Quisquam consectetur perferendis totam nostrum dicta corporis modi aut ab exercitationem praesentium distinctio similiqe ratione, non, inventore labore commodi vero animi expedita dolores, perspiciatis ad dolorem neque dolorum! Dolorem, dolores? Lorem ipsum dolor sit amet consectetur, adipisciing elit. Saepe nulla explicabo, harum repellat hic consequuntur ipsa quo quaerat atque nam vel voluptas neque, eveniet nihil alias! Rem odio eveniet consequatur. Lorem ipsum dolor, sit amet consectetur adipisciing elit. Iure minus sapiente harum temporibus, asperiores quaerat in illo recusandae eaque eveniet hic laudantium? Suscipit necessitatibus temporibus numquam optio unde sunt enim! Consequuntur sint nihil enim assumenda officia illo quia itaque, dicta placeat provident porro quae amet, ducimus ipsa voluptatibus explicabo sunt! Consequuntur asperiores, aliquid perferendis aperiam dolor soluta magni quidem incidunt. Nemo alias doloremque hic suscipit ex aliquam voluptatum enim libero expedita a ratione delectus placeat eveniet, esse accusamus repellendus voluptate eligendi corporis, amet, doloribus non rem labore! Minima, quas nobis! Animi delectus reprehenderit porro et dolorem similiqe praesentium consequuntur vitae eum. Possimus atque sunt qui, exercitationem velit eligendi nemo sequi quod, illum minima numquam soluta. Quis laudantium officiis totam aspernatur

9.- Modelo de cajas



Contenido: Contenido del elemento.

- **Padding:** Espacio entre el límite superior del elemento y el contenido del elemento.
- **Borde:** Borde que rodea al elemento.
- **Margen:** Espacio entre el elemento y los elementos que lo rodean.

Calcular la altura y la anchura de un elemento:

```
div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

Para calcular la anchura total del elemento:

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

9.1.- outline



La propiedad `outline` se encuentra aún más exterior que el borde. En este caso, no añade anchura al elemento.

Ejemplos:

```
p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}  
p.double {outline-style: double;}  
p.groove {outline-style: groove;}  
p.ridge {outline-style: ridge;}  
p.inset {outline-style: inset;}  
p.outset {outline-style: outset;}
```

Propiedades individuales: `outline-style`, `outline-color`, `outline-offset` y `outline-width`.

Formas de ahorrar código: Podemos especificar en una sola línea el estilo, anchura y color del `outline`.

```
p.ej1 {outline: dashed;}  
p.ej2 {outline: dotted red;}  
p.ej3 {outline: 5px solid yellow;}  
p.ej4 {outline: thick ridge pink;}
```

Propiedad `offset`: añade un espacio entre `border` y `outline`.

```
p.clase1 {  
    margin: 30px;  
    border: 1px solid black;  
    outline: 1px solid red;  
    outline-offset: 15px;  
}
```

10.- Trabajo con texto

Tenemos varias formas de formatear el texto:

10.1.- Color del texto:

```
#colortexto1 {  
    color: green;  
}
```

Trabajo con texto, CSS:

Color del texto

10.2.- Alineación del texto:

izquierda, derecha y centrada.

```
h1.alineacion1 {  
    text-align: center;  
}
```

```
h2.alineacion2 {  
    text-align: left;  
}
```

```
h3.alineacion3 {  
    text-align: right;  
}
```

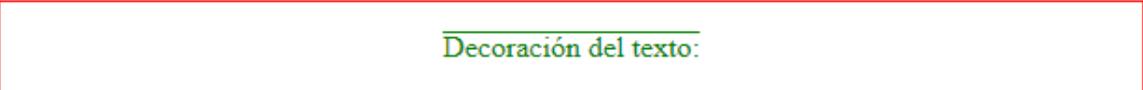
```
p.alineacion4 {  
    text-align: justify;  
}
```

10.3.- Decoración del texto:

sobrerrayado, tachado y subrayado.

```
p.sobre {  
    text-decoration: overline;  
}  
p.tachado {  
    text-decoration: line-through;  
}  
p.subrayado {  
    text-decoration: underline;
```

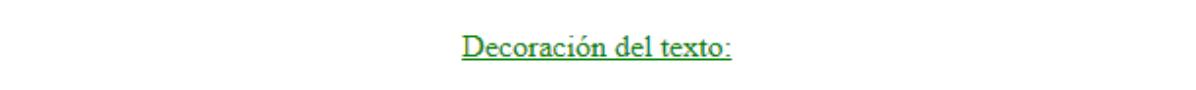
Trabajo con texto, CSS:

Decoración del texto:

Trabajo con texto, CSS:

Decoración del texto:

Trabajo con texto, CSS:

Decoración del texto:

10.4.- Transformación del texto:

mayúsculas, minúsculas y capital.

```
p.may {  
    text-transform: uppercase;  
}
```

Trabajo con texto, CSS:

TRANSFORMACIÓN DEL TEXTO:

```
p.min {  
    text-transform: lowercase;  
}
```

Trabajo con texto, CSS:

transformación del texto:

```
p.cap {  
    text-transform: capitalize;  
}
```

Trabajo con texto, CSS:

Transformación Del Texto:

10.5.- Tabulado del texto:

Lo especificamos con una unidad relativa o absoluta.

```
p.tabulado {  
    text-indent: 30px;  
}
```

```
ejemplosMiniproyecto.html ×  
ejemplosMiniproyecto.html > html > body > div.capa1 > p.parr2  
9     <link rel="stylesheet" text="text/CSS" href="ejemplosMiniproyecto.css">  
10    </head>  
11  
12    <body>  
13        <h1>Trabajo con texto, CSS:</h1>  
14        <div class="capa1">  
15            <h2>Tabulado del texto: </h2>  
16            <p class="parr2">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique  
17                accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium  
18                consectetur, repudiandae ea! Totam unde illo accusamus.*5  
19            </p>  
20        </div>  
21  
# ejemplosMiniproyecto.css ×  
# ejemplosMiniproyecto.css > .parr2  
1 .capa1{  
2     width: 50%;  
3     text-align: center;  
4     border: solid 1px red;  
5 }  
6  
7 .parr2{  
8     color: green;  
9     text-indent: 50px;  
10    text-align: justify;  
11}
```

Tabulado del texto:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique
accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum
laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

10.6.- Espaciado entre letras:

Especifica el espacio que existirá entre los caracteres de un texto. Podrá ser un valor negativo, por ejemplo, aplicado a un h1.

```
h1 {  
    letter-spacing: 10px;  
}
```

Trabajo con texto, CSS:

Espaciado entre letras:

```
  L o r e m   i p s u m ,   d o l o r   s i t   a m e t  
  c o n s e c t e t u r   a d i p i s i c i n g   e l i t .   V e l i t  
  s i t   a s p e r i o r e s   s i m i l i q u e   a c c u s a n t i u m ?  
  O m n i s   c o n s e q u u n t u r   a c c u s a n t i u m ,   e o s  
  a l i a s   f u g i a t   p o r r o   d o l o r i b u s   n o s t r u m  
  l a u d a n t i u m   c o n s e c t e t u r ,   r e p u d i a n d a e  
  e a !   T o t a m   u n d e   i l l o   a c c u s a m u s . * 5
```

10.7.- Altura de línea:

Especifica el espacio entre líneas, por ejemplo, para un párrafo. Si se utiliza un número, multiplica por el actual tamaño de fuente. En el siguiente ejemplo, aumentaría un 80% con respecto al tamaño de fuente.

```
p.muchoespacio {  
    line-height: 1.8;  
}
```

Trabajo con texto, CSS:

Altura de línea:

```
  L o r e m   i p s u m ,   d o l o r   s i t   a m e t  
  c o n s e c t e t u r   a d i p i s i c i n g   e l i t .   V e l i t  
  s i t   a s p e r i o r e s   s i m i l i q u e   a c c u s a n t i u m ?  
  O m n i s   c o n s e q u u n t u r   a c c u s a n t i u m ,   e o s  
  a l i a s   f u g i a t   p o r r o   d o l o r i b u s   n o s t r u m  
  l a u d a n t i u m   c o n s e c t e t u r ,   r e p u d i a n d a e  
  e a !   T o t a m   u n d e   i l l o   a c c u s a m u s . * 5
```

Otros ejemplos (se toma como referencia el tamaño de fuente actual, muchas veces dado por el navegador):

```
div.a {  
    line-height: normal; /*Opción por defecto*/  
}  
  
div.b {  
    line-height: 1.6;  
}  
  
div.c {  
    line-height: 80%;  
}  
  
div.d {  
    line-height: 200%;  
}
```

10.8.- Dirección del texto:

indica la dirección del texto. Por defecto es de izquierda a derecha.

Ejemplo (de derecha a izquierda):

```
p.alreves{  
    direction: rtl;  
}
```

Trabajo con texto, CSS:

Dirección del texto:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similiq**ue** accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

10.9.- Espaciado entre palabras:

especifica el espacio entre palabras. Puede tratarse de un valor negativo.

```
h1 {  
    word-spacing: 10px;  
}
```

```
h2 {  
    word-spacing: -5px;  
}
```

Trabajo con texto, CSS:

Espaciado entre palabras:

 Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

10.10.- Sombreado del texto:

establece un sombreado para el texto, tanto a nivel horizontal como vertical. También puede indicar el color del sombreado.

```
p {  
    text-shadow: 4px 2px blue;  
}
```

Trabajo con texto, CSS:

Sombreado del texto:

 Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

11.- Fuentes

A la hora de trabajar con fuentes, se pueden establecer una serie de propiedades. Familia de fuente: establece una familia de fuente. Si el navegador no soporta la fuente especificada, lo intenta con el siguiente tipo que aparezca en la lista.

Se pueden utilizar dos grandes grupos de familias de fuente: familias de **fuente genérica** y familias de **fuente específica**. Normalmente las familias de fuente específica se indican mediante comillas dobles.

Google tiene un repositorio de fuentes muy interesante y extenso:
https://www.w3schools.com/howto/howto_google_fonts.asp

11.1.- Repositorio de Google Fonts: <https://fonts.google.com/>

Ejemplo:

Tendríamos que enlazar con la api de Google:

```
<link href='https://fonts.googleapis.com/css?family=ABeeZee' rel='stylesheet'>
```

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI',

Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans',

'Helvetica Neue', sans-serif;

```
ejemplosMiniproyecto.html
ejemplosMiniproyecto.html > html > head > link
9   <link rel="stylesheet" type="text/CSS" href="ejemplosMiniproyecto.css">
10  <link href="https://fonts.googleapis.com/css?family=Vibes&display=swap" rel="stylesheet">
11 </head>
12
13 <body>
14   <h1>Trabajo con texto, CSS:</h1>
15   <div class="capa1">
16     <h2>Sombreado del texto: </h2>
17     <p class="parr2">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique
18       accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium
19       consectetur, repudiandae ea! Totam unde illo accusamus.*5
20     </p>
21   </div>
22

# ejemplosMiniproyecto.css
ejemplosMiniproyecto.css > .parr2
6
7 .parr2{
8   color: green;
9   text-indent: 50px;
10  text-align: justify;
11  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI',
12    Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans',
13    'Helvetica Neue', sans-serif;
14 }
```

Trabajo con texto, CSS:

Sombreado del texto:

 Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

Trabajo con @font-face

Uso: permite trabajar con fuentes instaladas por el propio desarrollador web.

Declaración del tipo de fuente:

```
@font-face{  
    font-family: miPrimeraFuente;  
    src: url(estilosDeFuente/Vibes-Regular.ttf);  
}
```

Trabajo con texto, CSS:

Fuentes de letras:

 Lor  m ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5

Uso del tipo de fuente:

```
.parr2{
    color: green;
    text-indent: 50px;
    text-align: justify;
    font-family: miPrimeraFuente;
}
```

The screenshot shows a code editor with two files open:

- ejemplosMiniproyecto.html** (top tab):
 - HTML structure: <html> -> <body> -> <div class="capa1"> -> <h2>
 - CSS link: <link rel="stylesheet" href="ejemplosMiniproyecto.css" />
 - Font link: <link href="https://fonts.googleapis.com/css?family=Vibes&display=swap" rel="stylesheet" />
- ejemplosMiniproyecto.css** (bottom tab):
 - Font declaration: @font-face { font-family: miPrimeraFuente; src: url(estilosDeFuente/Vibes-Regular.ttf); }
 - Class definition: .parr2 { color: green; text-indent: 50px; text-align: justify; font-family: miPrimeraFuente; }

Formatos de fichero de fuentes soportados por los navegadores:

Repositorios de fuentes:

<https://www.dafont.com/es/>

<https://www.1001freefonts.com/es/>

Estilo de fuente: establece un estilo de fuente (normal, cursiva u oblicua).

Ejemplo:

```
p.normal {  
    font-style: normal;  
}
```

Trabajo con texto, CSS:

Fuentes de letras:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

```
p.italic {  
    font-style: italic;  
}
```

Trabajo con texto, CSS:

Fuentes de letras:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

```
p.oblique {
    font-style: oblique;
}
```

Trabajo con texto, CSS:

Fuentes de letras:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similiqe accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

11.2.- Tamaño de fuente:

Esta propiedad determina el tamaño con el que una fuente se va a mostrar en el navegador.

Ejemplo:

Tamaño con píxeles

```
h1 {
    font-size: 40px;
}

h2 {
    font-size: 30px;
}

p {
    font-size: 20px;
}
```

```
.parr2{
    color: green;
    text-indent: 50px;
    text-align: justify;
    font-style: oblique;
    font-size: 20px;
}
```

Trabajo con texto, CSS:

Tamaño de fuente:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similiqe accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

Tamaño con em

(recomendado por la W3C)

1em es igual al tamaño actual de fuente del navegador, por lo general 16px. Permite que los usuarios puedan cambiar el tamaño de fuente en el menú del navegador (ver ejemplo con Firefox). Esta propiedad puede dar problemas en versiones antiguas de Internet Explorer.

```
h1 {  
    font-size: 2.5em; /* 40px/16=2.5em */  
}  
h2 {  
    font-size: 1.875em; /* 30px/16=1.875em */  
}  
p {  
    font-size: 0.875em; /* 14px/16=0.875em */  
}
```

Solución para que se muestre el mismo tamaño de texto en todos los navegadores (debido a que em tiene problemas de compatibilidad con versiones antiguas):

Establecemos la fuente por defecto en el body mediante porcentajes.

```
body {
```

```
    font-size: 100%;
```

```
}
```

```
h1 {
```

```
    font-size: 2.5em;
```

```
}
```

```
h2 {
```

```
    font-size: 1.875em;
```

```
}
```

```
p {
```

```
    font-size: 0.875em;
```

```
}
```

11.3.- Ancho de fuente:

Permite establecer una anchura para la fuente que se utilice.

Ejemplo:

```
p.normal {  
    font-weight: normal;  
}
```

Trabajo con texto, CSS:

Ancho de fuente:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

```
p.negrita {  
    font-weight: bold;  
}
```

Trabajo con texto, CSS:

Ancho de fuente:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velit sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus.*5*

Tamaño de fuente responsivo: este tamaño de fuente va a utilizar como referencia en tamaño de la ventana del navegador. 1vw es el 1% del tamaño de la ventana del navegador. Por ejemplo: si la ventana del navegador tiene 40cm de ancho, 1vw será equivalente a 0.4cm

Ejemplo:

```
<p style="font-size:10vw">Hola a todos</p>
```

Propiedad font-variant:

todas las letras minúsculas son convertidas a letras mayúsculas. Normalmente los tamaños de fuente son variables para las letras mayúsculas.

Ejemplo:

```
p.normal {  
    font-variant: normal;  
}
```

Trabajo con texto, CSS:

Ancho de fuente:

*Lorem ipsum, dolor sit amet consectetur adipisicing elit. Velix sit asperiores similique accusantium? Omnis consequuntur accusantium, eos alias fugiat porro doloribus nostrum laudantium consectetur, repudiandae ea! Totam unde illo accusamus. *5*

```
p.smallcaps {  
    font-variant: small-caps;  
}
```

Trabajo con texto, CSS:

Ancho de fuente:

*LOREM IPSUM, DOLOR SIT AMET CONSECTETUR ADIPISICING ELIT. VELIX SIT ASPERIORES SIMILIQUE ACCUSANTUM? OMNIS CONSEQUUNTUR ACCUSANTUM, EOS ALIAS FUGIAT PORRO DOLORIBUS NOSTRUM LAUDANTUM CONSECTETUR, REPUDIANDAE EA! TOTAM UNDE ILLO ACCUSAMUS. *5*

12.- Iconos

Se pueden añadir iconos a una web mediante una librería de iconos como Font Awesome, por ejemplo.

Lista de iconos: https://www.w3schools.com/icons/icons_reference.asp

Se debe añadir la siguiente línea de código en la sección <head> de la página web:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Estos iconos se deben utilizar con elemento en línea, como puede ser un o un <i> (Se suele utilizar para especificar un texto en lenguaje técnico).

Ejemplo de utilización de iconos:

```
<i class='fa fa-cc-paypal'></i>  
<i class='fa fa-cc-paypal' style='font-size:24px'></i>  
<i class='fa fa-cc-paypal' style='font-size:36px'></i>  
<i class='fa fa-cc-paypal' style='font-size:48px;color:red'></i>  
<br>
```

<p>Uso con un botón:</p>

```
<button style='font-size:24px'>Botón <i class='fa fa-cc-paypal'></i></button>
```

Iconos CSS:

Ancho de fuente:



Uso con un botón:



13.- Favicon

El favicon es un ícono de página. Es una imagen asociada a una web que aparece en la pestaña, barra de direcciones y en los marcadores del navegador.

Su tamaño normalmente es: 16×16, 32×32, 48×48, 64×64 o 128×128 píxeles con una profundidad de color de 8, 24 o 32 bits.

El fichero con la imagen debe de tener una extensión ico, png, jpg o svg.

Utilidad del favicon:

- Asociar la web a un elemento gráfico: imagen corporativa, logotipo...
- Identificar fácilmente la web de entre las pestañas que un usuario tenga abiertas en el navegador.
- Identificar la web fácilmente en los favoritos del navegador, historial de navegación, páginas recientes...

Para ahorrar trabajo, existen herramientas automáticas de generación de favicon:

<http://www.genfavicon.com/es/>

Para establecer un favicon para cualquier navegador (suele ser estándar, aunque existen algunos dispositivos que pueden necesitar adaptaciones extras):

<link rel="icon" href="/imagenfavicon.png" sizes="32x32" type="image/png">



Iconos CSS:

14.- Enlaces

Mediante CSS3 se pueden aplicar diferentes estilos a los enlaces. Para ello se dispone de multitud de propiedades que ya se han visto con antelación y otras que se verán a continuación.

Existen varios estados en los que se pueden encontrar los enlaces:

- **a:link** – un enlace normal, no visitado.
- **a:visited** – un enlace que ha sido visitado por el usuario.
- **a:hover** – cuando un usuario sitúa el ratón encima del enlace.
- **a:active** – un enlace en el momento en el que se hace clic sobre él.

Ejemplo:

```
/* enlace no visitado */  
a:link {  
    color: red;  
}
```

Al hacer clic en el botón se ilumina de color rojo.

Enlaces en CSS:



[Google](#)
[Amazon](#)

```
/* enlace visitado */  
a:visited {  
    color: green;  
}
```

Enlaces en CSS:



[Google](#)
[Amazon](#)

```
/* ratón situado encima del enlace */  
a:hover {  
    color: hotpink;  
}
```

Enlaces en CSS:



[Google](#)
[Amazon](#)
[Fonts Google](#)

```
/* enlace seleccionado, es decir, con clic sobre él*/  
a:active {  
    color: blue;  
}
```

Enlaces en CSS:



[Google](#)
[Amazon](#)
[Fonts Google](#)

14.1.- Decoración del texto:

Permite personalizar el enlace, de tal manera que no aparezca subrayado, por ejemplo.

Ejemplo:

```
a:link {  
    text-decoration: none;  
}  
  
a:visited {  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
a:active {  
    text-decoration: underline;  
}
```

1. Sin estilos css:

Enlaces en CSS:

[Google](#)
[Amazon](#)
[Fonts Google](#)

2. Estilo con CSS

Enlaces en CSS:

[Google](#)
[Amazon](#)
[Fonts Google](#)

14.2.- Color de fondo de los enlaces:

mediante la propiedad background-color se pueden establecer colores de fondo para los enlaces.

Ejemplo:

```
a:link {  
    background-color: yellow;  
}
```

Enlaces en CSS:



Google
Amazon
Fonts Google

```
a:visited {  
    background-color: cyan;  
}
```

Enlaces en CSS:



Google
Amazon
Fonts Google

```
a:hover {  
    background-color: lightgreen;  
}
```

Enlaces en CSS:



Google
Amazon
Fonts Google

```
a:active {  
    background-color: hotpink;  
}
```

Enlaces en CSS:



Google
Amazon
Fonts Google

15.- Listas

Mediante CSS3 se pueden aplicar estilos a los elementos de tipo lista.

- - Lista ordenada
- - Lista desordenada
- - Elemento de una lista

Estilos de lista: mediante esta propiedad se pueden establecer diferentes tipos de lista, tanto para ordenadas como para desordenadas.

Ejemplo:

```
<body>
  <h1>Enlaces en CSS:</h1>
  <div class="capa1">
    <h2>lista1</h2>
    <ol class="lista1">
      <li>Elemento1</li>
      <li>Elemento1</li>
      <li>Elemento1</li>
    </ol>
    <br>
    <h2>lista2</h2>
    <ul class="lista2">
      <li>Elemento1</li>
      <li>Elemento1</li>
      <li>Elemento1</li>
    </ul>
    <br>
    <h2>lista3</h2>
    <ol class="lista3">
      <li>Elemento1</li>
      <li>Elemento1</li>
      <li>Elemento1</li>
    </ol>
    <br>
    <h2>lista4</h2>
```

Estilos con CSS

```
<ul class="lista4">
    <li>Elemento1</li>
    <li>Elemento1</li>
    <li>Elemento1</li>
</ul>
</div>
</body>
```

CSS:

```
.capa1 {
    width: 50%;
    text-align: center;
    border: solid 1px red;
}
```

```
.lista1 li{
    list-style-type: square;
}
```

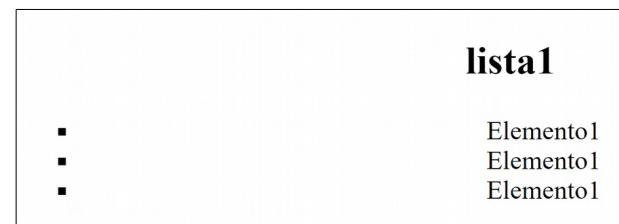
```
.lista2 li {
    list-style-type: circle;
}
```

```
.lista3 li{
    list-style-type: upper-roman;
}
```

```
.lista4{
    list-style-type: lower-alpha;
}
```

Ejemplo:

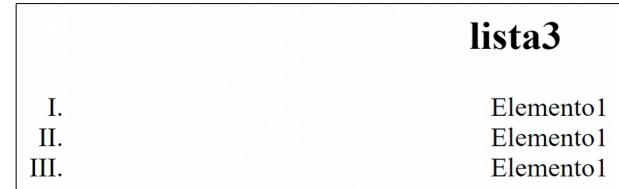
```
ul.a {  
    list-style-type: circle;  
}
```



```
ul.b {  
    list-style-type: square;  
}
```



```
ol.c {  
    list-style-type: upper-roman;  
}
```



```
ol.d {  
    list-style-type: lower-alpha;  
}
```

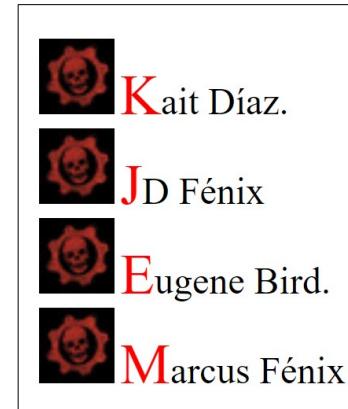


15.1.- Imagen como ícono de lista:

Se puede establecer una imagen como ícono de lista.

Ejemplo:

```
.clasel1KirTsiKos{  
    margin-top: 50px;  
    list-style-image: url("img/icono181119KirTsiKos.jpg");  
}  
  
<ol class="clasel1KirTsiKos">  
    <li>Kait Díaz. </li>  
    <li>JD Fénix</li>  
    <li>Eugene Bird.</li>  
    <li>Marcus Fénix</li>  
</ol>
```



15.2.- Posición de los iconos de lista:

Se puede modificar la posición de los iconos de la lista mediante la propiedad list-style-position

Ejemplo:

```
ul.a {  
    list-style-position: outside;  
}
```



```
ul.b {  
    list-style-position: inside;  
}
```



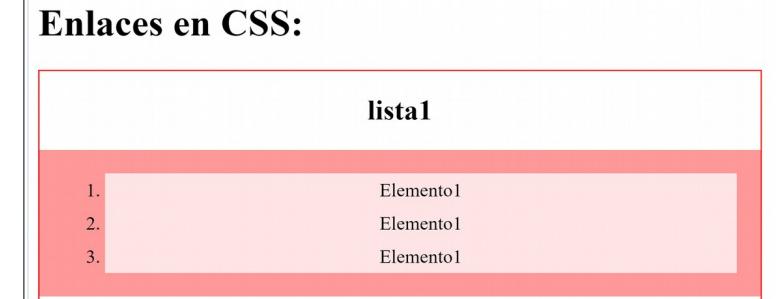
15.3.- Listas con colores:

Se pueden establecer listas con diferentes colores utilizando las propiedades background o background-color.

Ejemplo:

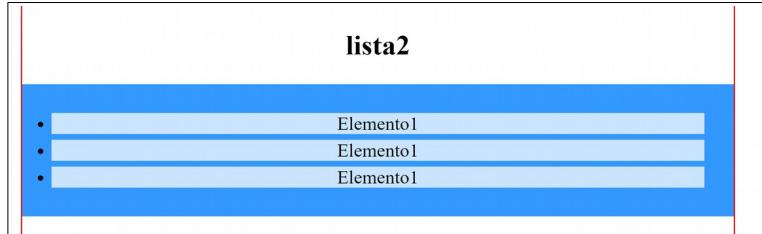
```
ol {
    background: #ff9999;
    padding: 20px;
}
ol li {
    background: #ffe5e5;
    padding: 5px;
    margin-left: 35px;
}
```

Enlaces en CSS:



```
ul {
    background: #3399ff;
    padding: 20px;
}
ul li {
    background: #cce5ff;
    margin: 5px;
}
```

lista2



16.- Combinadores-Selectores

Un combinador establece una relación entre dos selectores. Existen 4 tipos diferentes de combinadores en CSS:

16.1.- Selectores descendentes

Seleccionan todos los elementos que son descendientes de un elemento específico.

Ejemplo: Todos los párrafos que estén dentro de elementos div

1. Body

```
<div class="Cont1KirTsiKos1811">
```

```
    <p>"Este párrafo ha sido creado por extra y cumple un selector descendente"</p>
    
</div>
```

2. CSS:

```
.Cont1KirTsiKos1811 p{
    color: lightsalmon;
    letter-spacing: 2px;
    text-align: center;
}
```



16.2.- Selectores hijos

Selecciona todos los elementos que son hijos inmediatos de un elemento específico.

Ejemplo: todos los párrafos que son hijos inmediatos de un elemento div.

1. Body:

```
<div>
```

```
<p>
```

Todos los párrafos que son hijos inmediatos de un elemento div.

```
</p>
```

```
<snap>
```

`<p>Este otro parrafo no deberia de tener el mismo estilo del parrafo anterior.</p>`

```
</snap>
```

```
</div>
```

2. CSS:

```
div > p {  
    background-color: purple;  
    border: 5px ridge yellowgreen;  
}
```

Enlaces en CSS:

Todos los párrafos que son hijos inmediatos de un elemento div.

Este otro parrafo no deberia de tener el mismo estilo del parrafo anterior.

16.3.- Selectores adyacentes hermanos

Seleccionan todos los elementos que son adyacentes (inmediatamente posterior) a un elemento específico.

Ejemplo:

Body:

```
<body>
```

```
    <h1>Enlaces en CSS:</h1>
```

```
    <div>
```

```
        <div>
```

```
            <p>Este otro parrafo no deberia de tener el mismo estilo del párrafo siguiente.</p>
```

```
        </div>
```

```
    <p>Seleccionan todos los elementos que son adyacentes (inmediatamente posterior) a un elemento específico.
```

```
    </p>
```

```
    </div>
```

```
</body>
```

CSS:

```
div + p {  
    background-color: blue;  
}
```

Enlaces en CSS:

Este otro parrafo no deberia de tener el mismo estilo del parrafo anterior.

Seleccionan todos los elementos que son adyacentes (inmediatamente posterior) a un elemento específico.

16.4.- Selectores generales hermanos

Seleccionan todos los elementos que son hermanos de un elemento específico, es decir, que se encuentren en el mismo nivel.

Ejemplo: (Virgulilla: Alt Gr +4)

```
div ~ p {  
    background-color: green;  
}
```

17.- Pseudoclases

Una pseudoclase es usada para definir un estado de un elemento.

Por ejemplo, para los enlaces:

```
a:link {  
    background-color: yellow;  
}
```

```
a:visited {  
    background-color: cyan;  
}
```

```
a:hover {  
    background-color: lightgreen;  
}
```

```
a:active {  
    background-color: hotpink;  
}
```

17.1.- pseudoclases-combinas con CSS

Las pseudoclases se pueden combinar con clases CSS.

Ejemplo: sólo aplicaría esta propiedad en el estado hover de elementos que cumplan la clase resaltado.

```
a.resaltado:hover {  
    color: #ff0000;  
}
```

Enlaces en CSS:
[Enlace](#)

Estado hover en elementos div.

Posicionando el raton encima del **div** este elemento cambiará de color.

Ejemplo:

```
div {  
    background-color: green;  
    color: white;  
    padding: 25px;  
    text-align: center;  
}  
  
div:hover {  
    background-color: blue;  
}
```

Enlaces en CSS:

Enlace

Estado hover en elementos div.

17.2.- Caso interesante:

Ejemplo:

Mostrar un párrafo oculto descendiente de un div al situar el ratón encima del div.

```
p {  
    display: none;  
    background-color: yellow;  
    padding: 20px;  
}
```

```
div:hover p {  
    display: block;  
}
```

```
<body>
```

```
<div>Sitúa el ratón encima para mostrar el párrafo  
<p>¡Aparece!</p>  
</div>
```

```
</body>
```

Enlaces en CSS:

Enlace

Sitúa el ratón encima para mostrar el párrafo

¡Aparece!

Pseudoclase primer hijo:

selecciona los elementos que son primeros hijos de un elemento.

Ejemplo: seleccionaría únicamente los párrafos que son primeros hijos de algún elemento.

```
p:first-child {  
    color: blue;  
}
```

Ejemplo: seleccionaría el primer span que esté dentro de cada párrafo.

```
p span:first-child{  
    color:yellow;  
}
```

Ejemplo: seleccionaría todos los que se encuentren en los párrafos que son primeros hijos de algún elemento.

```
p:first-child span {  
    color: pink;  
}
```

18.- Pseudoelementos

Se suelen utilizar para dar estilos a unas partes específicas de un elemento.

Pseudoelemento **first-line**: se utiliza para dar estilo a la primera línea de un elemento. **Sólo se puede aplicar a elementos de bloque.**

Ejemplo: la primera línea de los párrafos tendrá el siguiente estilo.

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

18.1.- Pseudoelemento **first-letter**:

Se utilizar para dar estilo a la primera letra de un elemento. Sólo se puede aplicar a elementos de bloque.

Ejemplo: la primera letra de los párrafos tendrá el siguiente estilo.

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

Enlaces en CSS:

[Enlace](#)

Aparece

Los pseudoelementos se pueden combinar con clases CSS

```
p.clase1::first-letter {  
    color: #ff0000;  
    font-size: 200%;  
}
```

Se pueden combinar múltiples pseudoelementos:

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

```
p::first-line {  
    color: #0000ff;  
    font-variant: small-caps;  
}
```

Enlaces en CSS:

[Enlace](#)



18.2.- Pseudoelemento before:

Se utiliza para insertar contenido antes de un elemento.

Ejemplo: inserta una imagen antes de los elementos h1

```
h1::before {  
    content: url(imagen1.png);  
}
```

18.3.- Pseudoelemento after:

se utilizar para insertar contenido después de un elemento.

```
h2::after {  
    content: " - Esto es un encabezado h2";  
}
```

18.4.- Pseudoelemento selection:

se utiliza para dar estilo a un elemento seleccionado por el usuario.

Ejemplo:

```
::selection {  
    color: red;  
    background: yellow;  
}
```



Código para Firefox:

```
::-moz-selection { /* Para Firefox */  
    color: red;  
    background: yellow;  
}
```

19.- Display

Display es una de las propiedades más importantes en CSS a la hora de controlar la distribución de los elementos en una página web.

Posibilidades:

1. p.ejemplo1 {display: none;}

Se utiliza para ocultar elementos.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
    display: none;
}
</style>
</head>
<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the h1 element with display: none; does not take up any space.</p>
</body>
</html>
```

The screenshot shows a browser developer tools interface with a code editor on the left and a preview on the right. In the code editor, there is a red box around the line `display: none;` in the CSS block. In the preview area, there is a red box around the second

element. The text "This is a visible heading" is displayed above the second , and a note below it says: "Notice that the h1 element with display: none; does not take up any space." There are also red arrows pointing from the notes back to their respective elements in the code and preview.

2. p.ejemplo2 {display: inline;}

Los enlaces de la derecha aparecen como elementos de linea y no como un linea.

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
    display: inline;
}
</style>
</head>
<body>
<p>Display a list of links as a horizontal menu:</p>
<ul>
    <li><a href="/html/default.asp" target="_blank">HTML</a></li>
    <li><a href="/css/default.asp" target="_blank">CSS</a></li>
    <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>
</body>
</html>
```

The screenshot shows a browser developer tools interface with a code editor on the left and a preview on the right. In the code editor, there is a red box around the line `display: inline;` in the CSS block. In the preview area, there is a red box around the

 element. A note above the preview says: "Display a list of links as a horizontal menu:" followed by three links: "HTML", "CSS", and "JavaScript".

3. p.ejemplo3 {display: block;}

En este caso los enlaces de la derecha aparecen como elementos de bloque

The screenshot shows a code editor window with the following code:

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  display: block;
}
</style>
</head>
<body>



Display links as block elements:</p>

<a href="/html/default.asp" target="_blank">HTML</a>
<a href="/css/default.asp" target="_blank">CSS</a>
<a href="/js/default.asp" target="_blank">JavaScript</a>

</body>
</html>


```

The output on the right side of the editor shows three links stacked vertically:

Display links as block elements:
[HTML](#)
[CSS](#)
[JavaScript](#)

4. p.ejemplo4 {display: inline-block;}

The screenshot shows a browser developer tools interface with the following code in the left panel:

```
padding: 0;
}

.nav li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}
</style>
</head>
<body>



# Horizontal Navigation Links



By default, list items are displayed vertically. In this example we use display: inline-block to display them horizontally (side by side).



Note: If you resize the browser window, the links will automatically break when it becomes too crowded.



- Home
- About Us
- Our Clients
- Contact Us

```

The right panel shows the rendered HTML with four links: Home, About Us, Our Clients, and Contact Us, displayed horizontally. The developer tools sidebar on the right shows the CSS styles applied to the links.

19.1.- Elementos de bloque:

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

19.2.- Elementos inline:

-
- <a>
-

19.3.- Display: none

Se utiliza para ocultar un elemento y que además no ocupe espacio en la página web.

Ejemplo:

```
div.clase1 {  
    width: 60%;  
    height: 250px;  
    background-color: red;  
    display: none;  
}
```

Modificar la propiedad display por defecto de un elemento: gracias a CSS, esto es posible

Ejemplo: modiflico la propiedad display por defecto de un elemento li (block)

```
li {  
    display: inline;  
}
```

Relativo a los elementos con display: inline

Anotación respecto a los elementos inline-block (se añade un espacio de forma automática entre ellos):

<https://css-tricks.com/fighting-the-space-between-inline-block-elements/>

20.- Visibility

Si utilizamos el valor hidden (oculto) funciona de manera similar a display: none, pero en este caso, el elemento sigue ocupando espacio en la página web. Los elementos suelen tener su propiedad visibility con el valor visible.

Ejemplo:

```
h1.oculto1 {  
    visibility: hidden;  
}  
  
Posibilidades:
```

```
h2.clase1 {  
    visibility: visible;  
}  
  
h2.clase2 {  
    visibility: hidden;  
}
```



The screenshot shows a browser's developer tools with the CSS tab selected. On the left, the CSS code is displayed:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h2.a {  
    visibility: visible;  
}  
  
h2.b {  
    visibility: hidden; [Red arrow]  
}  
</style>  
</head>  
<body>  
  
<h1>The visibility Property</h1>  
<h2 class="a">This heading is visible</h2>  
<h2 class="b">This heading is hidden[Red arrow]</h2>  
  
<p>Notice that the hidden heading still takes up space on the page.</p>  
</body>  
</html>
```

On the right, the preview pane shows the heading "The visibility Property". Below it, the heading "This heading is visible" is displayed, followed by a red arrow pointing to its content. A separate paragraph below states: "Notice that the hidden heading still takes up space on the page.", also with a red arrow pointing to it.

21.- Position

Es una propiedad de CSS que sirve para modificar el tipo de posicionamiento de un elemento.

Posibilidades:

static

relative

fixed

absolute

sticky

21.1.- Position static:

Se trata del valor por defecto en el posicionamiento de un elemento. Los elementos con este posicionamiento siguen el flujo normal de la página y **no se ven afectados por las propiedades top, bottom, left y right.**

Ejemplo:

```
.capa1 {
    width: 50%;
    text-align: center;
    border: solid 1px red;
}

div.estatico {
    position: static;
    border: 3px solid #73AD21;
    left: 50px; /*left no afecta a los elementos de tipo static*/
}
```

Enlaces en CSS:

Enlace

 Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad, quisquam? Optio fuga minima nihil corporis impedit eum id quos ex exercitationem! Dolorem fugiat vero rerum ea molestiae. Ab, mollitia aut.

 Lorem ipsum, dolor sit amet consectetur adipisicing elit. Non quas nulla tenetur quos deserunt cupiditate. Vel, nesciunt facere. Corrupti provident accusantium quibusdam suscipit iste neque quam quis tenetur sapiente ipsam. Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit eaque aliquid vitae architecto, ipsa perspiciatis, ex molestiae unde magnam labore nisi distinctio blanditiis debitis illum ut a illo harum rem? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Molestias, assumenda minima, vero suscipit, labore odit ad nulla fuga quod non itaque tenetur deleniti impedit id rem placeat! Nam, reprehenderit dolore! Doloremque sed quis nostrum. Cum dolore a ullam ipsam laudantium autem quibusdam quod saepe tempora, sunt soluta! Neque deleniti quasi asperiores eveniet optio, unde repellat inventore, et, ut ullam sint! Quis accusamus eaque nihil dolores placeat eum rem! Rerum veniam quaerat nobis vero modi explicabo maxime dolores atque magnam saepe beatae culpa, consequuntur voluptatibus aperiam ipsum assumenda, nesciunt libero fugit! Maiores nihil laudantium repudiandae tempora quod aspernatur doloremque animi eaque! Enim quas non suscipit libero corporis, id aspernatur facilis accusantium blanditiis quos perferendis,

21.2.- Position relative:

Se posiciona de manera normal en el flujo del documento, sin embargo, sí se ve afectado por las propiedades top, bottom, left y right. Se posicionaría respecto a su posición normal, teniendo en cuenta los objetos que se encuentren por encima en el flujo del documento (tiene en cuenta la dimensión de esos elementos).

Ejemplo:

```
div.relativo {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```

Enlaces en CSS:

Enlace

Div posicionado estaticamente "static"

Div posicionado relativamente "relativ"

21.3.- Position fixed:

Se posiciona de manera relativa a la ventana del navegador y se adapta al scroll vertical que se aplique, es decir, el elemento se desplazará si hacemos scroll vertical. El objeto será **posicionado con las propiedades top, bottom, left y right**.

Ejemplo:

```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
```

Enlaces en CSS:

Enlace

Div posicionado estaticamente "static"

Div posicionado relativamente "relativ"

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quam accusamus blanditiis nostrum aliquam est quaerat beatae quidem repudianda officia minus tempora voluptate, explicabo et incidunt eveniet natus sunt enim! Necesitatis veritatis aspernatur recusandae modi sint fuga eaque minima debitis provident. Ratione nostrum aperiam perspiciat. Iste accusamus pariatur, minima voluptates rerum eveniet, mollitia molestias asperiores officia quam quis blanditiis obcaecati. Debitis odit consequatur odio nobis dolorum aspernatur iste asperiores voluptatum beatae inventore repellendus ducimus amet, et deserunt iusto vitae mollitia. Ad animi recusandae ducimus voluptas consequatur sequi laudantium doloremque laborum. Facilis modi rerum possimus blanditiis ad non soluta consequatur laboriosam et quas, nulla aut, enim, labore aperiam asperiores similique? Obcaecati soluta eum, voluptates magni nulla voluptatis illum debitis ut incidunt! Illum magnam eos possimus laborum consequetur, ullam nisi vitae paratur recusandae optio libero quae harum? Dolore consequatur laudantium eos nam. Fugiat aperiam illo praesentium voluptatis explicabo sapienti adipisci ducimus minima? Placet enim iste culpa facilis distinctio perferendis animi, aut, hic ad accutum magnam quisquam, dolor blanditiis laborum veniam laudantium odit aspernatur alias voluptatis omnis natura aliquid commodi. Atque, molestias tempore! Velit, nesciunt doloribus! Esse suscipit error possimus quasi iste, itaque eos nisi optio quia vitae consequetur, ullam repudianda ad odio laborum, sunt hic? Cum quacrat quee odio laborum aut atque! Cumque modis hic vero quis iure ratione accusamus repellat ipsam ab pariatur quasi, iusto delectus laboriosum necessitatibus, atque aspernatur, recusandae dicta voluptatem obcaecati odio ducimus officia corporis sunt! Consequatur, accusamus. Quia optio reprehenderit maiores. Possimus faciliis ducimus molestiae aliquid ut, dolores quasi debitis qui illam placeat voluptates molestias velit architecto quis neque segni officia sit est! Quos ipsam a vel. Corripit assumenda beatae ipsum quo, at a sunt repellendus odit ipsa laborum et consequatur vero consequatur incidunt earum reprehenderit blanditiis consequetur iste maiores amet ad! Expedita maxime ducimus ex. Dicimur. Earmi vero dolorem obcaecati, nemo itaque quisquam esse non! Laborum assumenda possimus, deserunt laudantium architecto quis eum. Vel sit querat qui quidem id praesentium, cum nesciunt, hic temporibus, dolorem asperiores. Atque debitis sit omnis ipsa error veniam, eveniet labore tempora sunt necessitatibus ex illo non minima exercitationem magnam, voluptatum consequetur, quod ratione recusandae aut accusamus commodi libero quae unde! Reiciendis. Tenetur quam animi sed a. Maiores sequi veniam consequetur voluptate aspernatur a sint cumque dolore exercitationem. Velit, ut, quasi esse voluptates totam nesciunt doloribus consequatur distinctio, exercitationem blanditiis sed adipisci! Harum magnam est explicabo rem, accusamus aperiam voluptas consequatur maiores accusantium a aliquid iusto ipsa sequi ad ipsam, quasi, in resellendus officia cundititate sint? Ponendo ea

animi laborum illum sequi esse numquam modi labore sint molestias expedita magni in utere delectus. Aut excepturi cupiditate perferendis consequatur! Quia delectus at nulla. Ut quasi temporibus illum maxime saepe ratione cum sunt ab, at, quae soluta inique quis. Maiores neque accusamus voluptatum nostrum! Error modi sit nobis ad vitae veritatis repellendus dolor illo odit num odio sapiente, reciendis perspiciat amet in assumenda voluptate tempore esse placeat totam possimus? Delectis recusandae est blanditiis voluntatum. Voluptatem, doloremque. Dolore paratur sapiente eos assumenda, voluptatum saepe doloribus quae similique reprehenderit magni dicta quos possimus eam. Minima tenuer iure nihil cum quis repudianda libero praesentium accusamus veniam quo? Debitis voluptas aliquam beatae, necessitatibus tenetur voluptatum sed ipsa non fugit tempora consequetur architecto eos, in quibusdam quam vel ab placeat, quae querat eligendi aperiam itaque voluptatis quis. Laudantium, quibusdam? Maiores fuga quo blanditiis assumenda iste facilis. Nihil quo quis quidem dolore impedit, similique nobis ipsum! Temporibus labrum a perspiciat qui doloribus nesciunt reprehenderit reciendis sepe, nihil tenuer voluptate dignissimos! Distinctio doloremque omnis accusamus fugit, nre nesciunt adipisci culpa aspernatur debitis non praesentium, tempora sed eius natus aut impedit! Architecto veritatis quidem eligend? Sequi, laboriosam sed excepturi iste aspernatur commodi. Molestiae voluptatum maxime asperiores magnam explicabo que, quibusdam iure ex accusantium placeat inventore! Accusamus unde quasi, autem querat, architecto laborum natus consequetur reciendis consequatur voluptas facere est deserunt. Fugit, fugit. Beatae sed esse quod, neque ipsa molestiae asperiores dolor. Officiis perspiciat ducimus dignissimos similique alias iure rerum adipisci quasi ut sapiente, animi id aliquid saepe delectus eos hic tenuer a. Saepē id, vitae sequi, incidunt quos cum voluptate rerum animi ea facere voluptas quam nostrum, ducimus qui molestias quibusdam magnam. Tenetur repellat eum architecto sed suscipit officiis reprehenderit illum natus. Magnam delectus voluptatum dignissimos accusamus animi quam qui dolores erit facilis. Nihil, sliquam veritatis voluptatum dolorum volupat quisquam? Veritatis quis distinctio atque nemo? Molestias commodi temporibus alias voluptate. Commodo, deserunt? Fugiat dolorem aliquid molestiae quod asperiores dicta ullam! Quod iusto officiis sapiente. Non, eaque fugit aperiam nemo asperiores saepe ratione officiis a, magnam reciendis quia ex tempora, accusantium nam delectus. Paratur doloremque exercitationem omnis fugit delectus aliquid, magnam, obcaecati velit quisquam esse est repellendus cumque molestias vel ab possimus ex dolorem quis! In maiores reprehenderit laudantium necessitatibus eaque facilis quasi. Illum vitae eius nemo dignissimos, voluptates quibusdam autem numquam debitis consequatur qui quis sed natus ipsum nihil eurum dolore exercitationem odio facere eveniet, dolores laudantium veniam. Libero sapiente quod vitae. Quo numquam quibusdam, nobis repellat accusamus enim ipsum eveniet! magnam, unde modi nihil omnis labore veniam maiores a? Ut vitae eius voluptatum, iusto totam aliquid rerum cumque dolor maiores odio? Minus harum ad ratione? Aperiam quas, paratur numquam aspernatur consequetur id facilis quis delectus nema excepturi rem totam repellat inventore odit atque, officia neque debitis dolorum eam omnis que quasi. Iste veritatis facilis fugiat facere explicabo repellat minima in hic consequetur, consequatur cumque culpa delectus, iure quidem delectus distinctio nobis voluptates dolorum. Officiis ratione possimus quos autem anima dolor eum. Odio, querat? Ducimus modi in, neque deserunt ea animi beatae repellendus ad dolore totam dignissimos enim consequetur id consequatur itaque libero a? Autem, non ipsa provident in sint praesentium. Dignissimos. Illo, excepturi rem possimus a natus obcaecati impedit nam tenetur culpa repellat tempora ut incident aut voluptatum nesciunt nemo numquam sapiente sequi, querat consequatur ducimus dicta dolor! Ratione, ab asperiores. Quidem aperiam veritatis quibusdam quod distinctio corporis possimus pariatur, quisquam aliquid minus expedita, dolores tempora facilis eius labore laudantium harum inventore eos qua accusamus. Unde explicabo labore id facilis quibusdam? At asperiores delectus beatas suscipit, reprehenderit iusto distinctio explicabo quasi ea molestiae excepturi voluptatum magnam totam culpa amet voluptibus tempore! Ex est consequetur quidem eum alias veniam repellat facilis qui? Facilis expedita delectus officiis facere? Illum sunt omnis optio ullam, accusamus, alias iste ea culpa harum aperiam officia, magnam vero ipsam nam. Fugit iste recusandae corripi repellat corporis natus quis. Tenetur in eaque nemo accusantium magnam, ut quam consequatur cum aperiam veritatis sit, facilis harum blanditiis! Officiis distinctio laboriosam, error quae qui, esse labore expedita asperiores assumenda voluptas quo. Div' posiconado "fixed"

21.4.- Position absolute:

El elemento se posicionará en relación a la ventana del navegador o su parent más cercano (al contrario que fixed, que siempre lo hará en relación a la ventana del navegador).

Ejemplo:

```
div.absoluta {  
    position: absolute;  
    top: 80px;  
    right: 0;  
    width: 200px;  
    height: 100px;  
    border: 3px solid #73AD21;  
}
```

Posicionamiento en CSS:

Div posicionado estaticamente "static"

Div posicionado relativamente "relativ"

Div posicion absolut
"absolut"

Div posicionado "fixed"

21.5.- Position sticky:

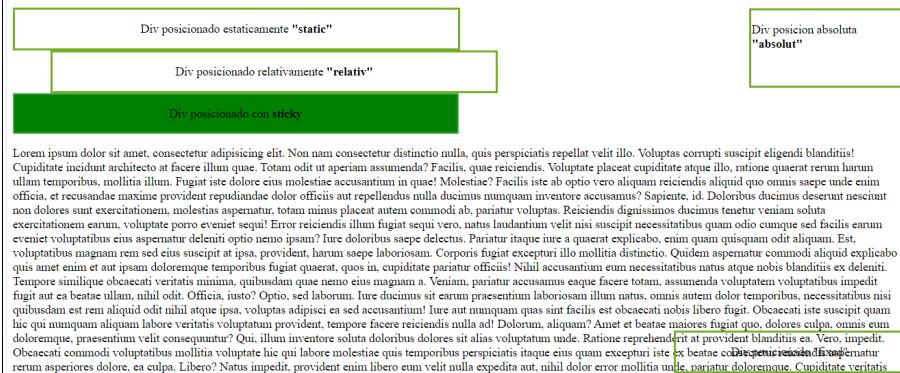
Se trata de una mezcla entre position relative y position fixed. Se posiciona de manera relativa hasta que su posición es alcanzada por la parte superior del navegador y es “arrastrada” por éste.

Versiones antiguas de Edge no soportan este tipo de posicionado. Safari requiere un código personalizado mediante webkit, tal como se muestra en el siguiente ejemplo.

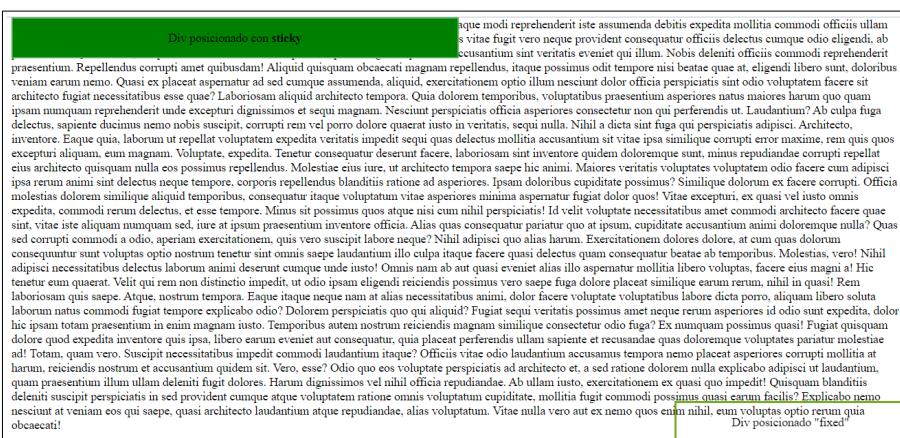
Ejemplo:

```
div.sticky {  
    position: -webkit-sticky; /* Safari */  
    position: sticky;  
    top: 0;  
    background-color: green;  
    border: 2px solid #4CAF50;  
}
```

Posicionamiento en CSS:



Al hacer scroll en la pagina este elemento se mantiene en la parte superior del navegador.



21.6.- z-index

Posicionamiento de elementos por niveles: z-index

Mediante la propiedad z-index, se puede variar el orden en el que aparecerán unos elementos que se encuentran encima de otros. Cuanto mayor sea el valor de z-index, más prioridad le dará al elemento sobre el que se ha aplicado para aparecer sobre los demás.

Lo aplicamos de 100 en 100 para facilitar la corrección de posibles modificaciones.

Ejemplo:

```
div.capa1 {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: -1;  
}
```

```
div.capa1 {  
    position: absolute;  
    left:30px;  
    top: 50px;  
    z-index: 1;  
}
```

Ver ejemplo con imágenes y texto:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.contenedor {  
    position: relative;  
}  
  
.centrado {
```

```
position: absolute;  
left: 0;  
top: 50%;  
width: 100%;  
text-align: center;  
font-size: 18px;  
}  
  
img {  
width: 100%;  
height: auto;  
opacity: 0.3;  
}  
</style>  
</head>  
<body>  
  
<h2>Texto sobre imagen</h2>  
<p>Vamos a añadir texto sobre la imagen:</p>  
  
<div class="contenedor">  
    
  <div class="centrado">Centrado</div>  
</div>  
  
</body>  
</html>
```



22.- Overflow

Mediante la propiedad **overflow** se puede especificar que el contenido del elemento aparezca cortado o se creen barras de desplazamiento para que se pueda visualizar en su totalidad.

Existen diferentes valores que puede adquirir esta propiedad:

- **visible** – Opción por defecto. El contenido no será cortado y sobrepasará el tamaño de su contenedor.
- **hidden** – El contenido será cortado y no se verá en su totalidad.
- **scroll** – El contenido será cortado, pero se añadirán barras de desplazamiento para poder visualizar el resto.
- **auto** - El contenido será cortado, pero se añadirán barras de desplazamiento (si es necesario) para poder visualizar el resto.

Ejemplo de visible:

```
div {  
    width: 200px;  
    height: 50px;  
    background-color: #eee;  
    overflow: visible;  
}
```

Overflow

Div opcion overflow **visible**

Ejemplo de hidden:

```
div {  
    overflow: hidden;  
}
```

Ejemplo de scroll:

```
div {  
    overflow: scroll;  
}
```

Ejemplo de auto:

```
div {  
    overflow: auto;  
}
```

Overflow

clearfix

Lore ipsum dolor, sit amet consectetur adipisicing elit. Totam maxime officiis delectus quis, laboriosam quia, illo sapiente quo, laborum blanditiis autem earum ratione aliquam aspernatur provident vero hic. Debitis, cupiditate. Mollitia, quidem nostrum dolores quam dolor in, eligendi voluptas molestias odio cum deserunt reprehenderit porro. Fugit quis officia ad aliquid reiciendis est nemo. Animi reiciendis fugit eos voluptatum repellendus non? Dolorum dolore eveniet cum iure sed accusamus modi in totam ratione, placeat nulla temporibus nesciunt hic accusantium et delectus.



Se pueden ajustar las barras de desplazamiento a nivel horizontal y a nivel vertical:

```
div {  
    overflow-x: hidden; /* A nivel horizontal, se ocultan las barras */  
    overflow-y: scroll; /* A nivel vertical, se muestran las barras */  
}
```

Overflow

clearfix

Lore ipsum dolor, sit amet consectetur adipisicing elit. Totam maxime officiis delectus quis, laboriosam quia, illo sapiente quo, laborum blanditiis autem earum ratione aliquam aspernatur provident vero hic. Debitis, cupiditate. Mollitia, quidem nostrum dolores quam dolor in, eligendi voluptas molestias odio cum deserunt reprehenderit porro. Fugit quis officia ad aliquid reiciendis est nemo. Animi reiciendis fugit eos voluptatum repellendus non? Dolorum dolore eveniet cum iure sed accusamus modi in totam ratione, placeat nulla temporibus nesciunt hic accusantium et delectus.



22.1.- *Overflow:auto*

Es un muy buen recurso cuando tenemos un elemento flotante dentro de un contenedor y el elemento flotante le supera en altura. Mediante esta propiedad el contenedor se adaptará a la altura del contenido.

```
div {
```

```
    border: 3px solid #4CAF50;
```

```
    padding: 5px;
```

```
}
```

```
.clearfix {
```

```
    overflow: auto;
```

```
}
```

```
.img2 {
```

```
    float: right;
```

```
}
```

```
<div class="clearfix">
```

```
    
```

 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Phasellus imperdiet, nulla et dictum interdum...

```
</div>
```

Overflow

clearfix

 Lorem ipsum dolor, sit amet consectetur adipisicing elit. Totam maxime officiis delectus quis, laboriosam quia, illo sapiente quo, laborum blanditiis autem earum ratione aliquam aspernatur provident vero hic. Debitis, cupiditate. Mollitia, quidem nostrum dolores quam dolor in, eligendi voluptas molestias odio cum deserunt reprehenderit porro. Fugit quis officia ad aliquid reiciendis est nemo. Animis reiciendis fugit eos voluptatum repellendus non? Dolorum dolore eveniet cum iure sed accusamus modi in totam ratione, placeat nulla temporibus nesciunt hic accusantium et delectus voluptates aspernatur magni porro. Mollitia quae ipsum veritatis natus impedit quisquam! Consectetur culpa pariatur omnis, ratione debitis deleniti corporis. Accusamus repellat provident dicta blanditiis, inventore laboriosam perspiciatis ut dolores, cupiditate placeat quia accusantium animi fuga dolore veniam modi harum, sapiente soluta. Molestias tempora hic voluptas rerum a temporibus quisquam necessitatibus reprehenderit modi



23.- Float

La propiedad **float** especifica como un elemento debe flotar en una página web. Esta propiedad es muy importante a la hora de trabajar en la distribución de un documento web. Se suele utilizar para maquetar textos e imágenes de manera simple.

Puede tomar los siguientes valores:

- **left** – El elemento flotará a la izquierda, dentro de su contenedor.
- **right** - El elemento flotará a la derecha, dentro de su contenedor.
- **none** – El elemento no flotará a ningún lado. Este es el comportamiento por defecto de los elementos
- **inherit** – El elemento heredará el valor float de su parente.
- Ejemplo de float right (ver ejemplo con texto e imagen):

```
img {  
    float: right;  
}
```

En el body:

```
<p>  
Texto párrafo</p>
```

Ejemplo de float left:

```
img {  
    float: left;  
}
```

Ejemplo de float none (la imagen aparece donde esté situada en el texto):

```
img {  
    float: none;  
}
```

24.- Clear

La propiedad **clear** dictamina si los elementos pueden flotar o no.

Puede tomar varios valores:

- **none** – Permite que los elementos floten a ambos lados. Esta es la propiedad por defecto.
- **left** – No permite que los elementos floten a la izquierda.
- **right** – No permite que los elementos floten a la derecha.
- **both** – No permite que los elementos floten a ambos lados.
- **inherit** – Hereda el valor clear de su padre.

Ejemplo:

```
<head>
<style>
.div1 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}
.div2 {
    border: 1px solid red;
}
.div3 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}
.div4 {
    border: 1px solid red;
    clear: left;
```

```
</style>
</head>
<body>
<h2>Sin clear</h2>
<div class="div1">div1</div>
<div class="div2">Este ejemplo no utiliza clear</div>
<br><br>

<h2>Con clear</h2>
<div class="div3">div3</div>
<div class="div4">Este ejemplo utiliza clear left</div>
</body>
```

24.1.- Clearfix

Es un truco que permite que, si un elemento flotante es más grande que el elemento que lo contiene obligue, al elemento contenedor a adaptarse a esas dimensiones.

La clase **clearfix** la aplicamos sobre el contenedor.

Ejemplo:

```
<head>
<style>
div {
    border: 3px solid #4CAF50;
    padding: 5px;
}
```

```
.img1 {
    float: right;
}
```

```
.clearfix {
    overflow: auto;
}
.img2 {
    float: right;
}
```

Estilos con CSS

```
</style>  
</head>  
<body>
```

<p>En este ejemplo, la imagen es más alta que el elemento que la contiene y además flota a la derecha, por lo que desborda sobre su contenedor</p>

```
<div>  
  
Texto aleatorio</div>
```

<p style="clear:right">Para solucionar este problema, añadimos una clase clearfix con overflow auto:</p>

```
<div class="clearfix">  
  
Otro texto aleatorio</div>
```

```
</body>
```

Existe una clase clearfix aún más avanzada y que suele evitar cualquier posible error (se utiliza mucho en la actualidad aunque no es necesaria al utilizar Flex y Grid):

```
.clearfix::after {  
    content: "";  
    clear: both;  
    display: table;  
}
```

Otra versión moderna de clearfix:

```
.clearfix {  
    display:flow-root;  
}
```

Actualmente este problema se intenta solventar usando las tendencias Flexbox y Grid para la maquetación de páginas web.

25.- Alinear elementos

Alineación centrada de elementos a nivel horizontal: para centrar horizontalmente elementos de tipo bloque, se utiliza la propiedad margin:auto

Ejemplo:

```
.center {  
    margin: auto;  
    width: 50%;  
    border: 3px solid green;  
    padding: 10px;  
}
```

Alineación centrada de texto a nivel horizontal: se utiliza la propiedad text-align:center

Ejemplo:

```
.center {  
    text-align: center;  
    border: 3px solid green;  
}
```

Centrado de imágenes a nivel horizontal: lo ideal es poner las propiedades margin-left y margin-right a auto y además convertir en elemento en bloque.

Ejemplo:

```
img {  
    display: block;  
    margin-left: auto;  
    margin-right: auto;  
    width: 40%;  
}
```

Alinear elementos

```
margin-left: auto;  
margin-right: auto;
```



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Totam maxime officiis delectus quis, laboriosam quia, illo sapiente quo, laborum blanditiis autem earum ratione aliquam aspernatur provident vero hic. Debitis, cupiditate. Mollitia, quidem nostrum

Alineación izquierda y derecha utilizando **position:absolute** (Cuidado – Si posicionamos de esta manera dentro de un contenedor, hay que recordar que el contenedor debe estar posicionado a su vez, por ejemplo, **con position:relative**)

Hay que recordar que los elementos con position absolute puede situarse independientemente del flujo del documento web y se pueden superponer sobre otros.

```
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}
```

Alinear elementos

position: absolute



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Totam maxime officiis delectus quis, laboriosam quia, illo sapiente quo, laborum blanditiis autem earum ratione aliquam aspernatur provident vero hic. Debitis, cupiditate. Mollitia, quidem nostrum

Alineación izquierda y derecha utilizando float

```
.right {  
    float: right;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}
```

25.1.- Centrado vertical utilizando padding:

una solución muy rápida y sencilla es utilizar padding top y bottom:

Ejemplo con un texto dentro de un contenedor:

```
.center {  
    padding: 70px 0;  
    border: 3px solid green;  
}
```

```
<div class="center">  
    <p>Este texto está centrado verticalmente mediante padding</p>  
</div>
```

Se le puede añadir text-align:center para también centrarlo a nivel horizontal.

Centrado vertical utilizando line-height (sólo funciona con 1 línea)

```
<head>  
<style>  
.center {  
    line-height: 200px;  
    height: 200px;  
    border: 3px solid green;  
    text-align: center;
```

```
}

.center p {
    line-height: 1.5;
    display: inline-block;
    vertical-align: middle;
}
</style>
</head>
<body>

<h2>Centrado a nivel vertical y horizontal</h2>
<p>En este ejemplo se va a realizar el centrado vertical y horizontal de un elemento:</p>

<div class="center">
    <p>Este párrafo está centrado a nivel vertical y horizontal.</p>
</div>

</body>
```

Centrado vertical utilizando position y transform

```
<style>
.center {
    height: 200px;
    position: relative;
    border: 3px solid green;
}

.center p {
    margin: 0;
    position: absolute;
    top: 50%;
```

Estilos con CSS

```
left: 50%;  
-ms-transform: translate(-50%, -50%);  
transform: translate(-50%, -50%);  
}  
</style>
```

<h2>Centrado a nivel vertical y horizontal:</h2>

<p>Centrado de un elemento a nivel vertical y horizontal mediante position y transform:</p>

```
<div class="center">  
  <p>Párrafo centrado a nivel vertical y horizontal</p>  
</div>
```

<p>Cuidado. En versiones anteriores a IE8 puede haber problemas</p>