# MINIPROYECTO UT2

# USO DE ESTILOS

Tabla de contenido

[MINIPROYECTO UT2 1](#_Toc24970579)

[USO DE ESTILOS 1](#_Toc24970580)

[**1.** **Implantación de GIT.** 2](#_Toc24970581)

[**2.** **Empezamos con los Estilos CSS3** 4](#_Toc24970582)

[**2.1** **Diferentes formas de aplicar estilos a un documento html.** 4](#_Toc24970583)

[**2.2** **Formas de seleccionar un elemento:** 5](#_Toc24970584)

[**2.3** **Estilos de fondo y colores** 7](#_Toc24970585)

[**2.4** **Estilos de bordes** 8](#_Toc24970586)

[**2.5** **Padding** 11](#_Toc24970587)

[**2.6** **Margin** 12](#_Toc24970588)

[**2.7** **Estilos de fuente** 13](#_Toc24970589)

[**2.8** **Pseudoclases** 18](#_Toc24970590)

[**2.9** **Propiedad Display** 20](#_Toc24970591)

[**2.10** **Posicionamiento a través de la propiedad position** 22](#_Toc24970592)

[2.11 **Posicionamiento a través de la propiedad float**. 23](#_Toc24970593)

[**2.12** **Propiedad Z-index** 25](#_Toc24970594)

[**2.13** **Propiedad Overflow** 25](#_Toc24970595)

[**2.14** **Propiedad opacity** 26](#_Toc24970596)

[**2.15** **Medidas Relativas** 27](#_Toc24970597)

# **Implantación de GIT.**

Como en la anterior unidad de trabajo, lo primero sobre lo que voy a tratar es sobre los avances que hemos realizado en GIT.

En esta unidad, lo que hemos aprendido es a instaurar una clave SSH de forma permanente, es decir, que al hacer un push por ssh no sea necesario el añadir de nuevo la clave tanto privada como pública. Para poder hacer esto hay que seguir el siguiente proceso:

1. Hay que ir a la carpeta .ssh en la ruta C:/Usuarios/usuario

Puede que la carpeta no aparezca y es porque esta oculta.

Captura de pantalla de un celular

Descripción generada automáticamente

En esta carpeta, habrá un fichero de known\_hosts. Yo le duplique dos veces para no tener problemas al crear ficheros sin formato

Captura de pantalla de un celular

Descripción generada automáticamente

A estos ficheros los renombro, uno llamado: id\_rsa y el otro id\_rsa.pub

Captura de pantalla de un celular

Descripción generada automáticamente

Una vez hecho esto solo hay que copiar nuestra clave ssh que estará en la carpeta en la que lo creáramos. La clave privada en el fichero id\_rsa y la clave.pub en el fichero id\_rsa.pub

Y ya debería de funcionar, lo único que te pide la contraseña de la clave cada vez que haces el push.

# **Empezamos con los Estilos CSS3**

## **Diferentes formas de aplicar estilos a un documento html.**

Hay diferentes formas de aplicar nuestros estilos a un documento html. Además estas formas, también tienen un orden de aplicación. Vamos ahora a ver cuales existen:

1. Estilos es línea: son los estilos que aplicamos directamente sobre la etiqueta html a través de su propiedad style. Tienen prioridad máxima sobre cualquier otra forma de aplicar los estilos.
2. Estilos internos: se aplican en la cabecera del documento con las etiquetas <style> tienen el problema de que su reutilización en otros documentos es mala.
3. Estilos externos: Son los que declaramos en un archivo externo con formato css. Para que se puedan aplicar a nuestro docuemento solo será necesario añadirlos en la cabecera con un link:

<link rel="stylesheet" href="style.css">

La ventaja de realizar hojas externas es que podremos reutilizarlas en otros documentos de una manera muy sencilla.

Captura de pantalla de un celular

Descripción generada automáticamente

[Salida en el navegador](aplicacionEstilos/fichero.html)

**\*\* Se puede romper la prioridad de aplicación poniendo !important**

## **Formas de seleccionar un elemento:**

**Selector de elemento:** se utiliza para seleccionar un elemento a través de su etiqueta. Seleccionará a todos los elementos del documento con esa etiqueta.

Por ejemplo:

Span{

Color:blue;

}

**Selector por identificador:** se utiliza para seleccionar un elemento por su id que previamente se habrá establecido en el documento html. Para seleccionarlo ha de ir precedido por #

Por ejemplo:

#identificador{

Color: red;

}

**Selector por clase:** al igual que el anterior, podemos seleccionar un elemento por la clase que tiene en el html. Lo bueno aquí es que los elementos pueden compartir clase, por lo que estaríamos afectando a más de un solo elemento. Para seleccionarlo, el nombre de la clase debe de ir precedido por .

Por ejemplo:

.miclase{

Color: violet;

}

**Selector de clase para un tipo de elemento específico:** Podemos seleccionar así el tipo de elemento concreto que tiene esa clase.

Por ejemplo:

p.parrafo{

color:blue;

}

**Agrupar selectores:** podemos seleccionar varios elementos separándolos por comas.

Por ejemplo:

H1, h2, p{

Color: red;

}

**Seleccionar por descendientes:** se puede seleccionar también por descendientes directos mediante el símbolo >

Por ejemplo:

P > span{

Color: blue;

}

Aquí seleccionamos los span hijos directos de un párrafo.

/\* Selector de Elemento\*/

p{

background-color: aqua;

}

/\*selector por identificador \*/

#parafo{

background-color: brown;

}

/\*selector por clase \*/

.clase{

background-color: coral;

}

/\*Selector de clase para un tipo de elemento específico\*/

p.clase2{

background-color: darkgreen;

}

/\* agrupacióln de selectores\*/

h2, h1{

background-color: darkmagenta;

}

/\*seleccion de descendientes \*/

p > span{

background-color: darksalmon;

}

SALIDA EN [NAVEGADOR](selectores/fichero.html)

## **Estilos de fondo y colores**

Vamos ahora a tratar sobre las formas de poner un fondo a un elemento. Hasta el momento conocemos dos:

* Dar fondo con un color.
* Dar fondo con una imagen.

Vamos con la primera, vamos a dar un color de fondo a un elemento. Para ello hay que usar la propiedad: **background-color: ;**

Podemos dar el color de diferentes formatos, los más comunes serían en formato hexadecimal, RGB, RGBA y por el propio nombre del color, ya que ahora con esditores de código es muy sencillo hacerlo.

.fondohexadecimal{

background-color: #feafea;

}

.fondorgb{

background-color: rgb(22, 70, 05);

}

.fondonombre{

background-color: salmon;

}

Vamos ahora a poner una imagen como fondo de un elemento. Lo primero es descargarse la imagen y guardarla dentro de nuestro proyecto.

Para poner una imagen de fondo hay que utilizar la propiedad: **background-image: ;.**

/\* Vamos a poner una imagen como fondo \*/

div{

background-image: url("imagenFondo.jpg");

min-height: 400px;

color: seashell;

}

[SALIDA EN EL NAVEGADOR](fondosycolores/fichero.html)

Ahora igual que hemos visto como poner un color de fondo y una imagen, vamos a ver como poner un color de fuente, se hace a través de la propiedad **color: ;** y el color se puede determinar de la misma forma que el fondo, es decir hexadecimal, RGB, por el nombre…

\* Propiedad color de la fuente \*/

p{

color: #aaaddd;

}

p.rgb{

color: rgb(12, 44, 76) ;

}

p.colores{

color: seagreen;

}

[SALIDA EN EL NAVEGADOR](file:///Users/sergiodepablos/Desktop/segundo%20DAW/Diseño%20de%20interfaces%20web/MINIPROYECTOUT2/fondosycolores/fichero.html)

## **Estilos de bordes**

Vamos ahora a tratar sobre las propiedades que pueden tener los bordes.

Aunque lo normal es ver la versión abreviada de declaración de un borde, en la que se pone el estilo, el grosor y el color, se puede declarar cada una de estas propiedades por separado. Así tenemos:

* Border-style: sirve para declarar el estilo de borde. Si queremos que sea punteado, a rallas, solido…
* Border-width: sirve para declarar la anchura del borde y normalmente se hace mediante píxeles.
* Border-color: sirve para declarar el color del borde.

Además estas propiedades podemos separarlas para cada uno de los bordes, es decir, superior, inferior, izquierdo y derecho.

Pero lo normal, es declararlo todo a la vez mediante la propiedad border.

Otra propiedad importante cuando tratamos sobre border es la propiedad border-radius, la cual hace que los bordes sean redondeados. Necesita también darle una medida.

.estiloborde{

border-style: dotted;

}

.anchoborde{

border-width: 3px;

}

.colorborde{

border-color: red;

}

.dimensionesCapa{

width: 60%;

height: 500px;

margin: auto;

}

.bordeizquierdo{

border-left-color: royalblue;;

border-left-style: solid;

border-left-width: 4px;

}

.bordederecho{

border-right-color: saddlebrown;

border-right-style: dashed;

border-right-width: 1px;

}

.bordesuperior{

border-top-color:violet;

border-top-style: double;

border-top-width: 8px;

}

.bordeinferior{

border-bottom-color: greenyellow;

border-bottom-style: dotted;

border-bottom-width: 2px;

}

.bordeizquierdoresumen{

border-left: 4px solid blue;

}

.bordesuperioresumen{

border-right: 1px dotted yellow;

}

.bordederechoresumen{

border-top: 6px dashed pink;

}

.bordeinferioresumen{

border-bottom: 2px double red;

}

.bordesredondeados{

border-radius: 10px;

border: green solid 4px;

}

IMPORTANTE.

Al poner un borde a un elemento este se suma al tamaño que ya tuviera este elemento, por lo que al posicionar los elementos con borde hay que tener esto en cuenta o usar la propiedad box-sizing: border-box;

Esta propiedad hace que el borde entre dentro del propio tamaño del elemento. Vamos a ver un ejemplo de ello:

/\*Demostración de la propiedad box-sizing\*/

.capaizquierda{

float: left;

width: 20%;

height: 300px;

border: 2px solid green;

}

.capacentro{

float: left;

width: 60%;

height: 300px;

border: 2px solid yellow;

}

.capaderecha{

float: left;

width: 20%;

height: 300px;

border: 2px solid blueviolet;

}

.container{

width: 95%;

margin: auto;

}

.contenedorcapassizing{

width: 100%;

margin-top: 50px;

}

.boxsizing{

box-sizing: border-box;

}

Tenemos tres capas dentro de un contenedor con un ancho y un borde que flotan a la izquierda, si no ponemos el box-sizing, el tamaño de las 3 capas superará el 100% del tamaño del contenedor y entonces una de ellas pasará a la línea de debajo.

[SALIDA EN EL NAVEGADOR](bordes/ejercicio1.html)

## **Padding**

El padding es el espacio interior dentro de un elemento de tipo bloque. Se puede declarar un padding distinto para cada zona (arriba, abajo, derecha o izquierda), declararlo de forma general para todos los lados o se puede declarar mismas medidas para arriba y abajo e izquierda y derecha.

Por ejemplo:

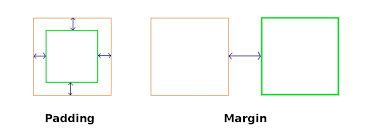
Padding: 5 10 5 10; Aquí lo declaramos primero arriba, derecha, abajo e izquierda.

Padding: 20 40; Aquí lo declaramos primero para arriba y abajo y la segunda medida izquierda y derecha.

Padding: 30; Mismo padding para todos los lados.

Además existen las propiedades padding-top, padding-bottom, padding-left y padding-right.

Cualquiera de estas formas sirve para declarar el padding de un elemento.



Aquí podemos ver una imagen de las diferencias del padding y el margin, que el padding es interno al elemento mientras que el margin es externo.

Ejemplo:

.dimensionescapa{

width: 70%;

height: 400px;

margin: auto;

border: 1px solid black;

margin-top: 30px;

}

.padding1{

padding-top: 30px;

padding-left: 40px;

padding-right: 50px;

padding-bottom: 50px;

}

.paddinggeneral{

padding: 50px;

}

[SALIDA EN NAVEGADOR](paddingyfuentes/ejercicio1.html)

## **Margin**

El margen es el espacio que hay entre dos elementos. Se declara con medidas y puede ser individual para cada uno de los lados o se puede declarar de forma general. También se puede declarar margen negativo y montar elementos dentro de otros (no es recomendable)

h3, p{

border: 1px solid black;

}

.superior{

margin-top: 50px;

}

.inferior{

margin-bottom: 50px;

}

.izquierda{

margin-left: 50px;

}

.derecha{

margin-right: 200px;

}

[SALIDA EN EL NAVEGADOR](margen/fichero.html)

## **Estilos de fuente**

* **Alineación del texto**: se usa la propiedad text-align y existen diferentes opciones:
* Right: para alinearlo a la derecha.
* Left: para alinearlo a la izquierda
* Center: para alinearlo al centro
* Justify: para alinearlo de manera justificada.
* **Decoración del texto**: se usa la propiedad text-decoration y existen diferentes opciones:
* Line-through: para poner texto tachado.
* Underline: texto subrayado.
* Overline: texto sobrerallado.
* None: si no queremos decoración (se usa sobretodo para quitar el subrayado por defecto de los enlaces).
* **Transformación del texto**: se usa la propiedad text-transform y existen diferentes opciones:
* Uppercase: para pasar a mayúsculas.
* Lowecase: para pasar a minúsculas.
* Capitalize: para poner la primera letra mayúscula.
* **Espacios entre texto:** Aquí no hay una sola propiedad, vamos a ir viendo una a una:
* Identación: usamos text-indent y nos sirve para dar un espacio al comienzo del texto.
* Espaciado entre letras: usamos letter-spacing.
* Altura de línea: usamos line-height.
* Espacio entre palabras: usaos Word-spacing.
* Dirección del texto: usamos direction.
* **Sombra del texto**: Usamos para ello la propiedad text-shadow, con 3 valores:
* Primero una cifra que indica la sombra a nivel horizontal.
* Segundo otra cifra que indica la sombra a nivel vertical.
* Tercero un color para la sombra.
* **Fuentes personalizadas**: es habitual, que desarrollando una web haya que poner fuentes personalizadas. Para ello hay dos formas, directamente a través del link que nos proporcione esa fuente (si esta en algún repositorio de fuentes externos como por ejemplo google fonts) o descargándonos la fuente y añadiéndola manualmente. Esta opción será la más conveniente, pues así no dependemos de terceros.

Para poder importar una fuente primero hemos de descargar el archivo .ttf. Una vez que ya tenemos la fuente en nuestro equipo hay que usar la propiedad @font-face para poder utilizarla.

En la propiedad @font-face será necesario establecer dos parámetros:

* Src: que será la ruta relativa de donde se encuentra la fuente.
* Font-family: para establecer un nombre a esa fuente.

Una vez hecho esto ya podemos asignar esta fuente a cualquier elemento de nuestra web a través de la propiedad Font-family y el nombre que le dimos anteriormente en el @font-face.

.alineacionderecha{

text-align: right;

}

.alineacioncentrada{

text-align: center;

}

.alineacionizquierda{

text-align: left;

}

.alineacionjustificada{

text-align: justify;

}

.tachado{

text-decoration: line-through;

}

.subrayado{

text-decoration: underline;

}

.sobrerrayado{

text-decoration: overline;

}

.transformacionmin{

text-transform: lowercase;

}

.transformacionmay{

text-transform: uppercase;

}

.transformacioncap{

text-transform: capitalize;

}

.identacion{

text-indent: 40px;

}

.espaciadoentreletras{

letter-spacing: 4px;

}

.alturadelinea{

line-height: 1.6;

}

.direcciontexto{

direction: rtl;

}

.wordspacing{

word-spacing: 15px;

}

h4{

text-shadow: 4px 6px salmon; /\*primera cifra a nivel horizontal, segunda cifra a nivel vertiacal\*/

}

@font-face{

font-family: familiafuente1;

src: url(../fonts/Modak-Regular.ttf);

}

.fuentepersonalizada{

font-family: familiafuente1;

}

[SALIDA EN EL NAVEGADOR](paddingyfuentes/ejercicio1.html)

**Más propiedades de fuentes:**

Vamos a mencionar otras propiedades que hacen referencia a las fuentes.

* Font-style: para dar un estilo a la fuente. Cabe distintas posibilidades:
* Normal: es el estilo de fuente por defecto.
* Italic: para el estilo de fuente en cursiva
* Oblique.
* Font-size: sirve para dar tamaño a la fuente. Se puede hacer tanto por medidas absolutas como con relativas. Por defecto el navegador tiene un tamaño de fuente de 16px.
* Font-weight: con esta propiedad podemos poner el texto en negrita a través de la propiedad bold.

.familiafuente1{

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

}

.estilofuente1{

font-style: normal;

}

.estilofuenteitalic1{

font-style: italic;

}

.estilofuenteoblicua{

font-style: oblique;

}

.tamaniofuente{

font-size: 25px; /\*medida absoluta\*/

}

.tamaniofuente2{

font-size: 1.5em;/\*medida relativa\*/

}

.anchofuente1{

font-weight: bold;

}

.anchofuente2{

font-weight: bolder;

}

.anchofuente3{

font-weight: normal;

}

.fontvariant1{

font-variant: small-caps;

}

[SALIDA EN NAVEGADOR](fuentesypseudoclases/ejercicio.html)

## **Pseudoclases**

Las pseudoclases se denotan con : tras el elemento sobre el que queremos que áctue. especifica un estado especial del elemento seleccionado. Por ejemplo,  [:hover](https://developer.mozilla.org/es/docs/Web/CSS/:hover) aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.

Vamos a ver las pseudoclases que hemos utilizado en clase:

* :link representa un elemento que aún no se ha visitado.
* :visited representa a un elemento que ya ha sido visitado.
* :active representa a un elemento sobre el que se está haciendo click.
* :hover representa a un elemento sobre el que esta encima el ratón.
* :before representa la parte justo antes de un elemento.
* :after representa la parte justamente posterior de un elemento.
* :first-child representa al primer hijo.
* ::first-line representa a la primera línea.
* ::selection representa a un elemento que ha sido seleccionado

Tanto con after como con before podemos añadir contenido a través de la propiedad content.

a:link{

color: greenyellow;

text-decoration: none;

}

a:visited{

color: palevioletred;

}

a:active{

color: aquamarine;

background: paleturquoise;

}

a:hover{

color: orange;

text-decoration: underline;

}

ol{

background-color: #ff9999;

padding: 20px;

}

ol li{

background-color: #ffe5e5;

padding: 5px;

margin-left: 35px;

}

ul{

list-style-image: url(../img/iconolista.jpg);

margin-left: 20%;

l

}

/\*pseudoclase firschild\*/

div p:first-child{

color: deeppink;

display: block;

}

/\*pseudoelemento firstline\*/

p.primeralinea::first-line{

font-size: 2em;

}

p.sangriafrancesa::first-line{

text-indent: 2em;

}

/\*pseudoelemento firstletter\*/

p.primeraletra::first-letter{

font-size: 3em;

font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;

}

/\*pseudoelemento before\*/

h6::before{

content: "Elemento - ";

}

h5::after{

content: " - videojuego";

}

::selection{

font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;

color: deeppink;

}

[SALIDA EN NAVEGADOR](fuentesypseudoclases/ejercicio2.html)

[SALIDA EN EL NAVEGADOR](pseudoclases/fichero2.html)

## **Propiedad Display**

La propiedad CSS **display** especifica el tipo de cuadro de representación utilizado para un elemento.

Vamos a ver las opciones que tenemos con la propiedad display:

* **Display: inline** . Nos sirve para establecer un elemento como un elemento en línea. Por ejemplo podemos convertir un párrafo que por defecto es de bloque en un elemento en línea.
* **Display block**. Es justo el contrario al anterior, es decir, convierte a un elemento en tipo bloque.
* **Display: inline-block**. Se comporta como un elemento en línea, porque ocupará lo que ocupe su contenido, pero permite aplicarle propiedades de elementos de bloque como un width y un height.
* **Display: none**. Con esto podemos hacer que desaparezca el elemento. Además no ocupará espcio en el documento. Es esa característica lo que le diferencia de otra propiedad que se usa para ocultar elementos que es **visibility: hidden**. Con esta propiedad el elemento desaparece pero mantiene el espacio que ocuparía si estuviera visible.
* .displaynone{
* width: 75%;
* min-height: 500px;
* border: 1px solid black;
* display: none;
* }
* .displayinline{
* display: inline;
* color: red;
* border: 1px solid black;
* width: 70%; /\* Al tratarse del valor inline de la propiedad display. no es posible establecer un ancho. Tampoco alto, margen supeior o inferior.
* \*/
* }
* .pruebasimagen{
* width: 80%;
* border: 3px solid red;
* box-sizing: border-box;
* }
* .displayinlineblock{
* width: 50%;
* border: 2px solid black;
* display: inline-block;
* box-sizing: border-box;

* }
* body{
* margin: 0;
* padding: 0;
* }
* .capaoculta{
* width: 90%;
* min-height: 600px;
* border: 1px solid black;
* visibility: hidden;
* }

[SALIDA EN EL NAVEGADOR](display_position/ejercicio1.html)

## **Posicionamiento a través de la propiedad position**

Existen diferentes formas de posicionar los elementos dentro de una web. Una de esas formas es a través de la propiedad **position.**

Al posicionamiento habrá que darle unas medidas a través de las propiedades: **top, bottom, left y right.**

Las diferentes opciones que permite la propiedad position son los siguientes:

* **Static**: es la posición por defecto natural que tiene un elemento.
* **Absolute**: con esta propiedad podemos posicionar un elemento con respecto alnavegador o a un elemento padre que este posicionado de forma relativa.
* **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).
* **Fixed**: posicionado un elemento de manera fija con respecto a la ventana delnavegador. Siempre estará en esa posición.
* **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.

.positionstatic{

position: static;

left: 30px;

top: 30px;

border: 1px solid black;

}

.positionrelative{

position: relative;

left: 200px;

top: 200px;

border: 1px solid green;

}

.positionrelative1{

position: relative;

left: 200px;

top: 200px;

border: 1px solid yellow;

}

.positionfixed{

position: fixed;

left: 30%;

top: 222px;

border: 1px solid black;

}

.posicionsticky{

position: sticky;

left: 60%;

top: 400px;

border: 1px solid slateblue;

}

[SALIDA EN NAVEGADOR POSITION ABSOLUTE, RELATIVE Y STATIC](display_position/ejercicio2.html)

[SALIDA EN NAVEGADOR RESTO DE POSITION](positionfixedsticky/ejercicio1.html)

## Posicionamiento a través de la propiedad float.

Otra de las formas de posicionar elementos en una página web es a través de la propiedad float, poner esta propiedad hace que un elemento abandone su posición por defecto y flote en la dirección que le hayamos puesto.

Es importante que para posicionar elementos mediante float mantengamos un buen orden de construcción de elementos, metiendo aquellos que queramos que floten dentro de capas para que no afecten al resto de elementos de la página.

Aun así hay dos formas que nos pueden ayudar a solucionar los problemas derivados del float.

* **Clear: both, right o left.** Esta sería la forma clásica para solucionar los problemas.
* **Display: flow-root.** Esta es la nueva forma para arreglar los problemas.

Ejemplo de posicionamiento:

.capaestandar{

width: 25%;

min-height: 200px;

border: 1px solid black;

}

.flotarizquierda{

float: left;

}

.flotarderecha{

float: right;

}

.contenedor{

margin-top: 5%;

width: 90%;

margin-left: auto;

margin-right: auto;

min-height: 400px;

border: 1px solid purple;

box-sizing: border-box;

clear: both;

}

.contenedormenu{

width: 95%;

height: 40px;

margin-bottom: 50px;

margin-left: auto;

margin-right: auto;

}

.elemento{

float: left;

text-align: center;

width: 25%;

padding: 15px;

border: 1px solid red;

box-sizing: border-box;

}

[SALIDA EN EL NAVEGADOR](positionfixedsticky/css/ejercicio4.html)

## **Propiedad Z-index**

Al igual que podemos posicionar elementos en los ejes x e y de la ventana del navegador, también podemos posicionarlos en función del eje Z, esto es la prioridad con la que aparecerán dos elementos posicionados de la misma manera. Quien tenga mayor propiedad Z-index será el elemento que aparezca con prioridad.

Se suelen dar medidas de 100 en 100.

.relativo{

    position: relative;

}

.capa{

    min-height: 300px;

    border: 1px solid black;

}

.z1{

    position: absolute;

    top: 0;

    left: 0;

    z-index: 100;

}

.z2{

    position: absolute;

    top: 0;

    left: 0;

    z-index: 200;

}

[SALIDA EN NAVEGADOR](z-index/fichero.html)

## **Propiedad Overflow**

Esta propiedad se utiliza para que aparezca una barra de scroll cuando el contenido de un elemento es superior al tamaño de este, por lo que se desbordaría. Con overflow el elemento mantiene su tamaño y lo que aparece es una barra de scroll para el contenido que se desborda del elemento.

Por ejemplo si tenemos un párrafo con mucha información y una altura fija, si le ponemos la propiedad overflow el texto no rebasará la altura fijada en el párrafo.

La propiedad overflow, puede tener distintos valores, lo normal es poner “auto” pues es lo más eficiente, ya que solo aparecerá la barra de scroll si es necesario.

Aún así también podemos poner:

* Scroll: poniendo esto, aparecerá barra de scroll siempre y aparecerá barra de scroll lateral.
* Hidden: no aparecerá barra de scroll pero el contenido no se desborda.
* Visible: No aparecerá barra de scroll y el contenido se desborda.
* .over{
* height: 300px;
* border: 1px solid black;
* overflow: auto;
* }
* .over1{
* height: 300px;
* border: 1px solid black;
* overflow: hidden;
* }
* .over2{
* height: 300px;
* border: 1px solid black;
* overflow: scroll;
* }
* .over3{
* height: 300px;
* border: 1px solid black;
* overflow: visible;
* }

[**SALIDA EN NAVEGADOR**](overflow/fichero.html)

## **Propiedad opacity**

La propiedad opacity sobre un elemento hace que este adquiera una transparencia, por lo que es posible ver elementos que estén debajo.

Las medidas van de 0 a 1, siendo 1 el estado normal de un elemento y 0 total transparente.

Ejemplo.

.capafondo{

    background-color: brown;

    opacity: 0.3;

    position: relative;

    height: 300px;

}

p{

    text-align: center;

    font-size: 3rem;

    font-weight: bold;

    position: absolute;

    top: 0;

    left: 30%;

}

[SALIDA EN NAVEGADOR](opacity/fichero.html)

## **Medidas Relativas**

Con el objetivo de hacer nuestras páginas más adaptables a todo tipo de dispositivos es conveniente usar medidas relativas.

Las medidas relativas son aquellas que se adaptan en función del tamaño de la pantalla, por su parte las absolutas siempre mantendrán las mismas dimensiones independientemente de la pantalla.

Dentro de las medidas **relativas** principales encontramos:

* **% :** el porcentaje es una de las medidas relativas más utilizadas.
* **Em:** se usa para el tamaño de fuente y es relativa al tamaño de fuente de su padre.
* **Rem**: También hace referencia al tamaño de fuente, pero en vez de tomar como referencia el padre, siempre tomará como referencia el tamaño de fuente del navegador.

Dentro de las medidas **absolutas** encontramos:

* **Px:** es la más utilizada.
* **Inch:** pulgadas.