

# CSS

# ¿Qué es CSS?

CSS es un lenguaje que permite asociar estilos a documentos de marcación como html y xml, además de poder ser utilizados por actionscript.

Son conjuntos de **reglas**, que se **asocian un estilo a una parte o todo el documento**.

Uno de los principales **problemas de accesibilidad** radica en la **utilización de etiquetas y atributos HTML para determinar el formato visual de una página Web.**

La utilización de estilos se convierte entonces en la **mejor herramienta para separar exitosamente la información del formato visual.**

# Hojas de Estilo en Cascada (Cascading Style Sheet - CSS)

**Son "bibliotecas de estilos"**

Su uso contribuye a tener **documentos bien definidos** y de significado completo, llamados **"DOCUMENTOS SEMÁNTICOS"**.

La forma en que CSS se asocian con documentos nos permite **reutilizarlos** para determinar el **formato de sitios enteros**, facilitando el mantenimiento del formato visual de los mismos.

Si no usamos CSS...

Si no separamos el contenido del diseño, o sea, incluimos los atributos en cada etiqueta html: debemos **abrir y modificar todas y cada una de las etiquetas para realizar los cambios.**

# Beneficios del uso de CSS

- **Aumento de accesibilidad**, ya que el contenido se mantiene "**limpio**" con etiquetas semánticas
- Trabajo **ordenado**, disminución del trabajo de producción y **mantenimiento**
- Utilidades que no se pueden lograr con HTML
- **Anidamiento**

# Beneficios del uso de CSS

- Podemos ver el mismo documento en una **infinidad de navegadores y dispositivos diferentes.**
- Pueden existir **varias hojas de estilo para un mismo documento**, discriminando por soporte o hasta se usa para **ordenar** los estilos en sitios **muy complejos**



# Sintaxis en CSS

## Selector o patrón de selección

Establece el **tipo de vínculo** de la regla con el documento html asociado

## Una o más declaraciones

Una propiedad y un valor

\*Múltiples declaraciones se separan por punto y coma.

# Sintaxis en CSS

```
selector {  
    propiedad: valor ;  
    propiedad: valor ;  
}
```

# Sintaxis en CSS

**¿A qué elemento?** **{ selector {**  
**propiedad: valor ;**  
**}**

**¿Qué propiedad  
deseo cambiar?** **¿Qué valor le doy?**

# CSS interno

Se puede aplicar en la misma página (html), en el HEAD.

```
<style type="text/css">  
  p {  
    color: red;  
  }  
</style>
```

# CSS externo - Virtudes

- **Una hoja sola controla todo**

Se pueden hacer todos los cambios juntos, optimiza tiempos

- **Optimizar para nuevas tecnologías**

Se tiene por separada el contenido de la apariencia

- **No tenemos que repetir** los mismos estilos

- Tenemos la **misma base para todas las páginas**, es reutilizable

# CSS externo

Se detalla en un archivo externo (archivo .css)  
y se llama desde el HEAD del html.

---

```
<link href="/css/archivo.css"  
rel="stylesheet" type="text/css" >
```

---

1

```
<style type="text/css">  
@import url("/css/archivo.css"); </style>
```

2

# CSS externo

1

```
<link href="/css/archivo.css"  
rel="stylesheet" type="text/css" />
```

**href** - Url del vínculo, puede ser absoluto o relativo.

# CSS externo

1

```
<link href="/css/archivo.css"  
rel="stylesheet" type="text/css" />
```

**rel** - Tipo de vínculo que se hace con el archivo externo.



# CSS externo

1

```
<link href="/css/archivo.css"  
rel="stylesheet" type="text/css" />
```

**type** - tipo de datos externos, text/css, textos y reglas css.

# **Tipos de selectores**

# Tipos de selectores simples

**selector universal** ("para todo")

```
* {  
  color: red;  
}
```

# Tipos de selectores simples

**selector de tipo** (elemento predefinido -etiquetas-)

```
p {
```

```
  color: red;
```

```
}
```

```
h1 {
```

```
  color: blue;
```

```
}
```

# Tipos de selectores simples

## selectores de clase

(cualquier elemento particularmente denominado, pueden ser varios)

```
.nombreClase {  
color: red;  
}
```

# Tipos de selectores simples

## selectores de ID

(un elemento particularmente denominado no se puede usar más de una vez por página)

```
#nombreID {  
  color: blue;  
}
```

# Tipos de selectores simples

## selectores de descendencia

(Cadenas de dos o más selectores separados por espacio blanco, que funcionan en cascada, no tienen porque ser "hijos directos")

```
p a{  
  color: blue;  
}
```

# Tipos de selectores simples

Los diferentes selectores pueden combinarse y agruparse.

- **combinados:** anidación, descendencia

(separados por espacio en blanco)

≠

- **agrupados:** p, a, span, em

(separados por coma)



# Tipos de selectores simples

## selectores de atributo

(permite seleccionar elementos HTML en función de sus atributos y/o valores de dichos atributos)

```
a [class]{  
color: red;  
}
```

(todos los enlaces que tengan asociados una clase, cualquiera sea, van a tomar color rojo y/o valores de dichos atributos)

# Tipos de selectores simples

## **selectores de atributo**

(permite seleccionar elementos HTML en función de sus atributos y/o valores de dichos atributos)

```
a [href="http://www.google.com"]{  
color: green;  
}
```

# Tipos de selectores compuestos

Algunos son selectores dinámicos ya que exigen una acción del usuario para su cumplimiento

## Pseudoclases - refiere a estados

\* es importante conservar el orden

**:link** links que aun no han sido visitados

**:visited** links previamente visitados

**:hover** cuando se pasa por encima sin accionarlo

**:active** mientras está siendo accionado

**:focus** cuando el enlace recibe el foco (tabulador)

# Tipos de selectores compuestos

## Pseudoclases

### **:first-child**

Primer elemento especificado dentro de otro.

"Primer hijo de su padre"

**Ej. `p:first-child { font-size:14px; }`**

Todos aquellos casos que "p" sea el primer hijo del elemento que lo contiene.

# Tipos de selectores compuestos

Otros selectores refieren a elementos generados por la estructura HTML pero no están etiquetados particularmente, sino que son abstracciones que nos permiten referenciar contenido del documento HTML...**Pseudoelementos**

# Tipos de selectores compuestos

## Pseudoelementos

### ::first-line

Primera línea de texto del elemento de bloque para el cual es especificado.

**Ej. `p::first-line { font-size:14px; }`**

Primer línea del párrafo.

# Tipos de selectores compuestos

## Pseudoelementos

### **::first-letter**

Primer letra de un elemento de bloque.

**Ej. `p::first-letter { font-size:14px; }`**

Primer letra del párrafo.

# Tipos de selectores compuestos

## Pseudoelementos

### ::before

Inserta dinámicamente contenido de un elemento.

**Ej. `a::before{ content:"Nuevo! " }`**

### ::after

Inserta dinámicamente contenido después de un elemento.

**Ej. `a::after{ content:"*" }`**



# Herencia - "estilos en Cascada"

**Cuanto más específica sea la regla, más poder tiene. Si hay dos normas iguales, prevalece la última declarada.**

!important - Algunos navegadores hacen caso a la importancia e interpretan dicha regla como definitiva.

# Prototipo de Hoja CSS

/\*

versión: 1.0

autor: Guzmán

email: info@miSitio.com

sitio: <http://www.miSitio.com/>

\*/

# Prototipo de Hoja CSS

*/\*GENERAL\*/*

*/\* Quita rellenos y márgenes \*/*

**\* { margin: 0; padding: 0; }**

*/\* Clase para eliminar flotaciones \*/*

**.clear { clear: both; }**

*/\* Sacamos borde alrededor de imágenes enlace \*/*

**img { border: 0; }**

# Prototipo de Hoja CSS

/\*\* GENERAL \*/

/\* TIPOGRAFÍA \*/

/\* ENCABEZADOS \*/

/\* ENLACES \*/

/\* PUBLICIDAD \*/

/\* NAVEGACIÓN PRINCIPAL \*/

/\* NAVEGACIÓN SECUNDARIA \*/

/\* CONTENIDO PRINCIPAL \*/

/\* CONTENIDO SECUNDARIO \*/

/\* PIE \*/

/\* FORMULARIOS \*/

/\* TABLAS \*/

/\* PARTICULARES \*/

# **Unidades de medida**

Las medidas **absolutas** y **relativas**, se indican con un **valor numérico entero o decimal** seguido de una **unidad de medida** (sin ningún espacio en blanco entre el número y la unidad de medida).

Si el valor es 0, la unidad de medida es opcional.

# Unidades relativas

**Em** es relativa respecto del tamaño de letra empleado

(el valor de la unidad se actualiza por herencia, tomando como referencia el padre de cada caja para establecer la nueva medida real)

*Para tener una idea de tamaños: 1.2em-es grande, 1em-es medio, 0.8em es chico*

**Rem** es relativa respecto del tamaño de letra empleado (root)

(el valor de la unidad se actualiza por herencia a partir del valor declarado en el html)

**Px** relativa respecto de la pantalla del usuario

# Unidades absolutas

Las unidades absolutas definen las medidas de forma completa, ya que **sus valores reales no se calculan a partir de otro valor de referencia, sino que son directamente los valores indicados.**

in, cm, mm, milímetros, picas, pt

# Porcentaje

Valor numérico seguido del símbolo % y siempre **está referenciado a otra medida**



**Propiedades  
de la fuente**

**font-style:** normal / italic / oblique

**font-variant:** small-caps / normal

**font-weight:** lighter / normal / bold / bolder / 100-900

**font-size/line-height:**

Tamaño absoluto (pt, in, cm, px) / Tamaño relativo (em, ex) / Porcentaje del tamaño

Palabras clave: xx-large, x-large, large, medium, small, x-small, xx-small, larger, smaller

**font-family:**

Verdana / Geneva, sans-serif

Georgia / Times New Roman / Times, serif

Courier New / Courier / monospace

Arial / Helvetica / sans-serif

## font-family:

El nombre genérico de una familia tipográfica no se refiere a ninguna fuente en concreto, sino hacen referencia al estilo de letra.

serif / sans-serif / monospace / cursive / fantasy

serif            Times New Roman

sans-serif    Arial

monospace   Courier New

cursive        **Comic Sans**

fantasy        **Impact**

Los navegadores muestran las páginas con las letras instaladas en la pc del usuario. Si se propone un tipo de letra rebuscado, casi ningún usuario dispondrá de la misma.

Para evitar esto, se propone una lista de fuentes y el navegador mostrará la página con la primer fuente disponible.

# font:

- Abreviatura de las propiedades de fuente
- El orden de los valores es importante:

font-**style** font-**variant** font-**weight** font-**size/line-height** font-**family**

- Cualquiera de estos valores puede ser omitido, pero se debe respetar el orden

*font: oblique small-caps bold 18px/24px sans-serif;*

*font: italic lighter 12px/14px serif;*

*font: 14px/20px fantasy;*

# **Propiedades de texto**

**word-spacing:** Espacios en blanco entre palabras

**letter-spacing:** Espacios en blanco entre letras

**text-decoration:** none / underline / overline / line-through

**vertical-align:** baseline / middle / sub / super / text-top / text-bottom

Alineación vertical de un elemento con respecto a su elemento padre. Sólo se puede aplicar a elementos de línea.

**text-transform:** capitalize / uppercase / lowercase

**text-align:** left / center / justify / right

**text-indent:**

Indentación de la primera línea en un elemento. Usado para crear el efecto de tabulación en los párrafos.

# **Propiedades de listas**

## list-style-type:

disc / circle / none / square

decimal / decimal-leading-zero / lower-latin / lower-roman / none

upper-latin / upper-roman

**list-style-position** inside / outside

**list-style-image** url (...)

## list-style:

- Abreviatura de las propiedades de lista
- El orden de los valores es importante:

**list-style-type** **list-style-position** **list-style-image**

- Cualquiera de estos valores puede ser omitido, pero se debe respetar el orden



# **Color y Propiedades de fondo**

**color:** #000 / red / #00ff00 / rgb(0,0,255) *\*para textos, hr, etc*

**background-color:** transparent / yellow / #00ff00 / rgb(255,0,255)

**background-position:**

left top/center/bottom

center top/center/bottom

right top/center/bottom

**background-repeat:** repeat / repeat-x / repeat-y / no-repeat

**background-attachment:** scroll / fixed

**background-image:** url

*\*Siempre que se use una imagen de fondo se recomienda agragar un color de fondo similar a la imagen para que aquellas personas que deshabilitan las img, puedan ver el sitio sin problemas*

# background:

- Abreviatura de las propiedades de fondo
- El orden de los valores es importante:

background-**color** background-**position** background-**repeat**  
background-**attachment** background-**image**

- Cualquiera de estos valores puede ser omitido, pero se debe respetar el orden

*background: #00ff00 center top no-repeat fixed url(img.jpg);*

*background: red repeat-x url(img.jpg);*

*background: right repeat-y url(img.jpg);*

# background:

- Ocupa el contenido y relleno, o sea, toda la caja. No el margin.
- Desde el punto de vista de la **accesibilidad**, nos conviene usar **imágenes de fondo** cuando se trata de un **gráfico decorativo y sin significado para quien navega en el documento**.
- Cuando se muestra una **imagen de fondo**, es habitual indicar además un **color de fondo**, ya que en caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

# **Propiedades de las cajas**

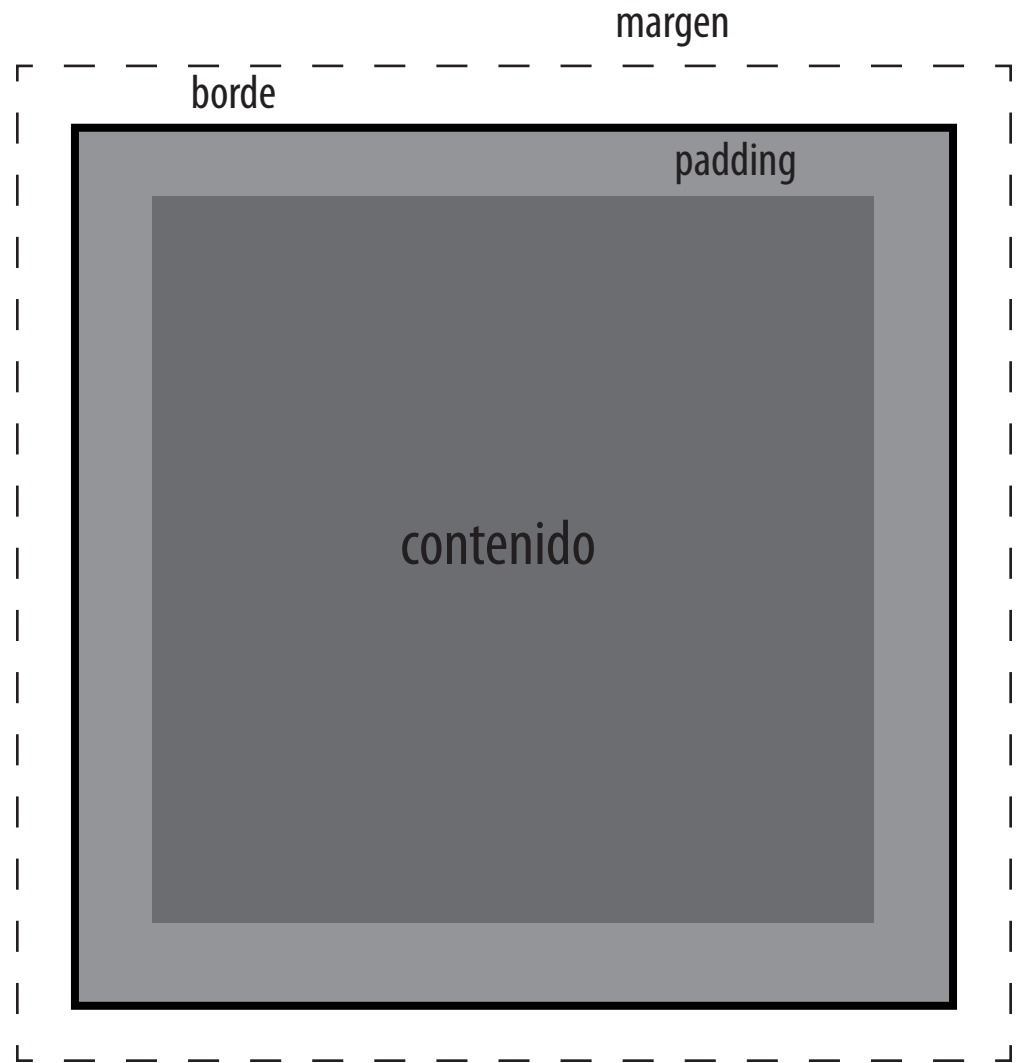
# El modelo de caja CSS

Este comportamiento provoca que todos los elementos incluidos en una página HTML se representen **mediante cajas rectangulares** donde el **CSS controla las propiedades y apariencia** de las mismas.

Las cajas se crean a medida que se insertan nuevas etiquetas o elementos en la página, estas **encierran cada contenido del sitio.**

# El modelo de caja CSS

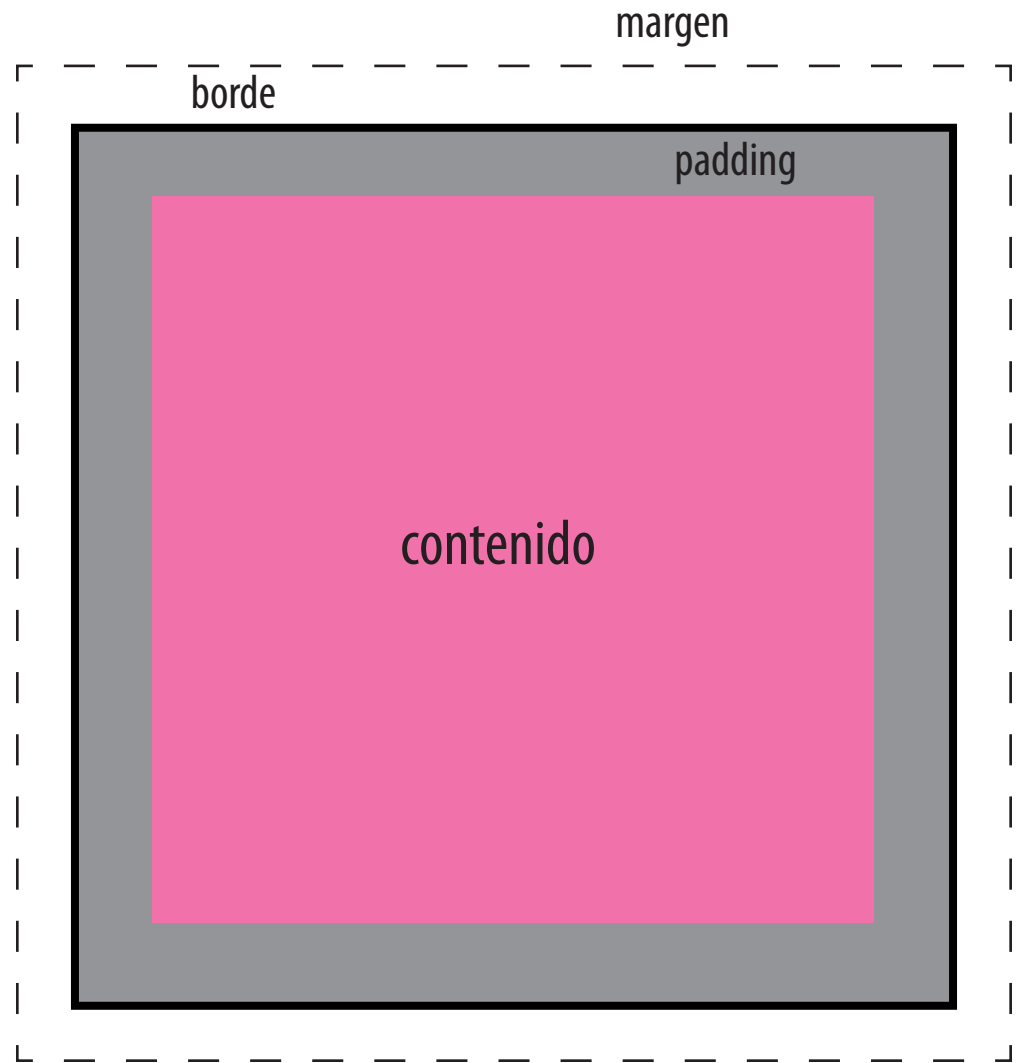
Las cajas se componen de un área de **contenido**, y áreas opcionales como **márgenes**, **rellenos** y **bordes**.



# El modelo de caja CSS

**Dichas áreas se separan por límites:**

- Límite del contenido:**  
Creado por el espacio ocupado por el contenido del elemento





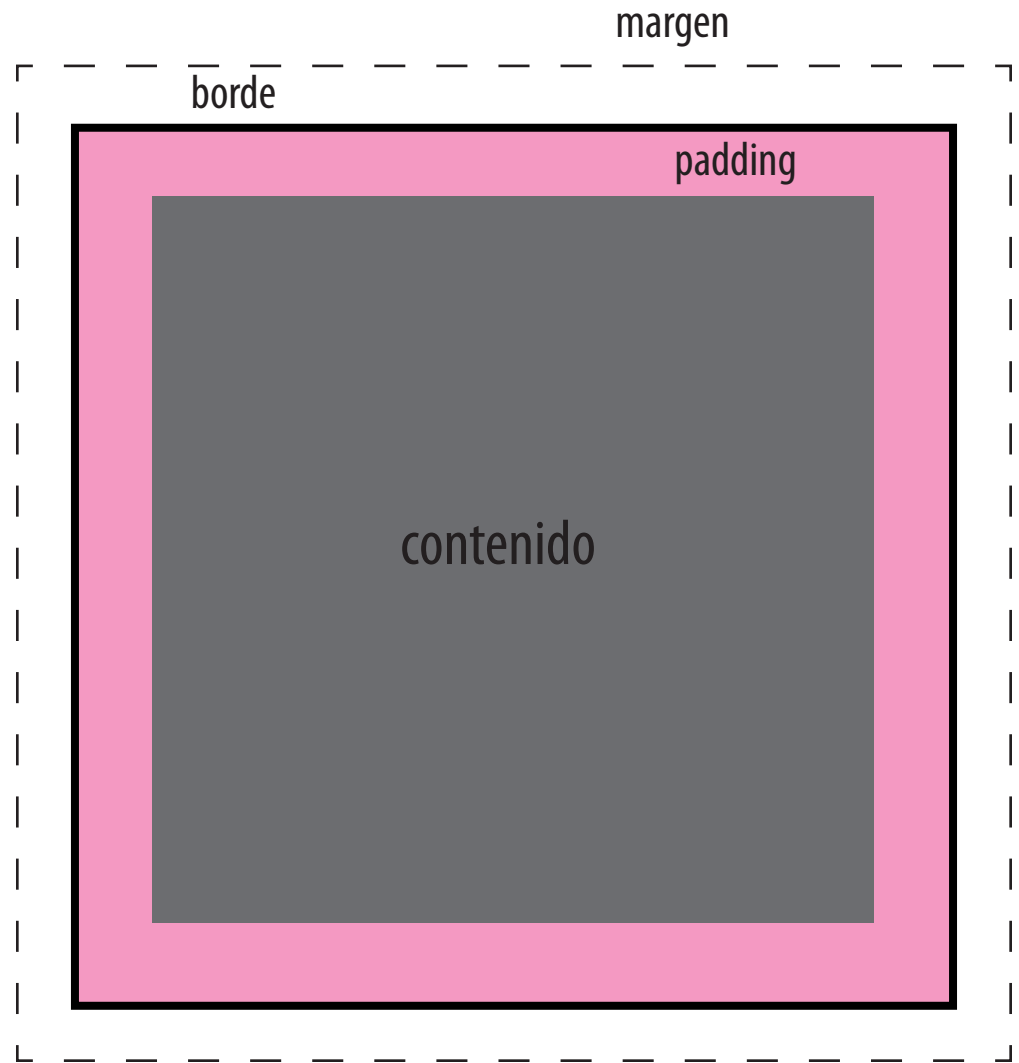
# El modelo de caja CSS

**Dichas áreas se separan por límites:**

**- Límite de padding:**

Se encuentra entre el límite del contenido y el borde

Es el espacio libre entre el contenido y el borde que lo encierra



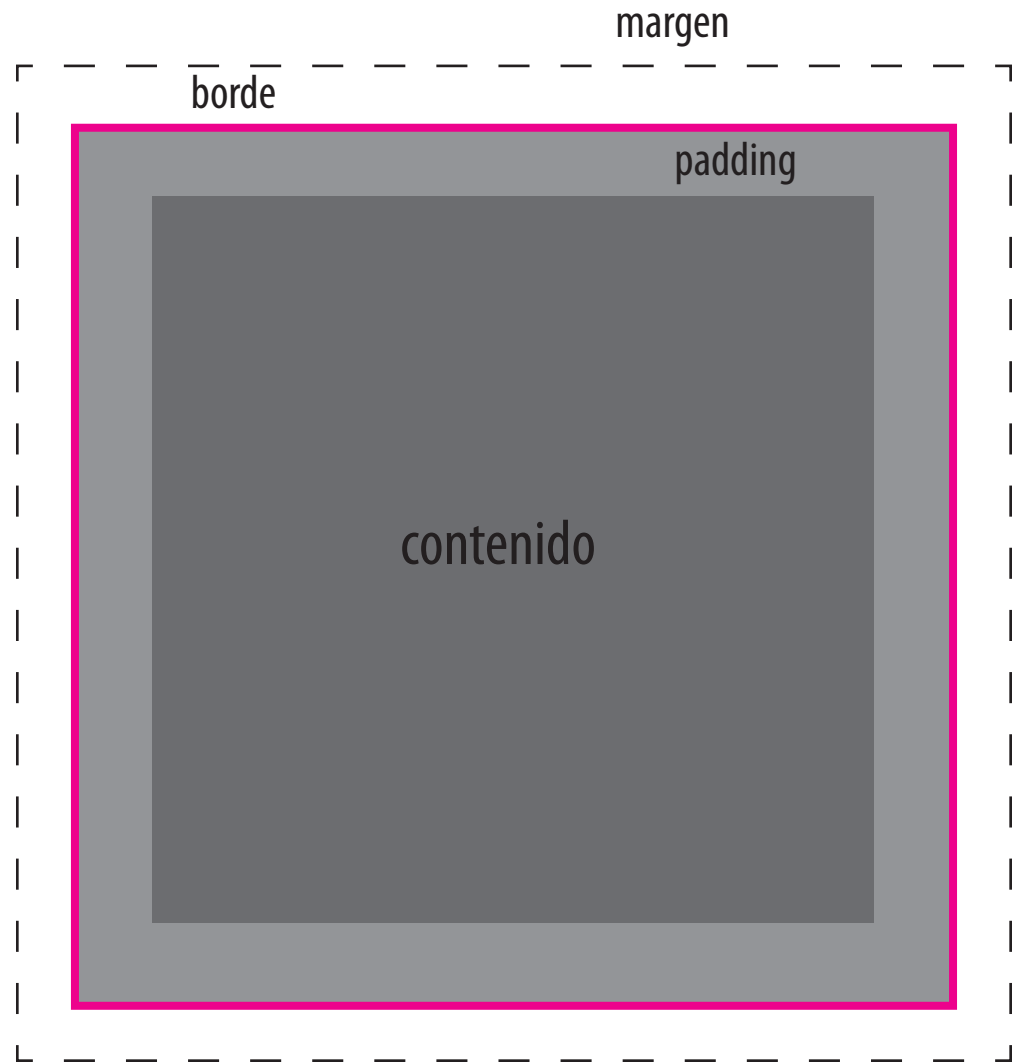
# El modelo de caja CSS

**Dichas áreas se separan por límites:**

**- Límite del borde:**

Establecido por el grosor del borde

Línea que encierra completamente el contenido y su relleno



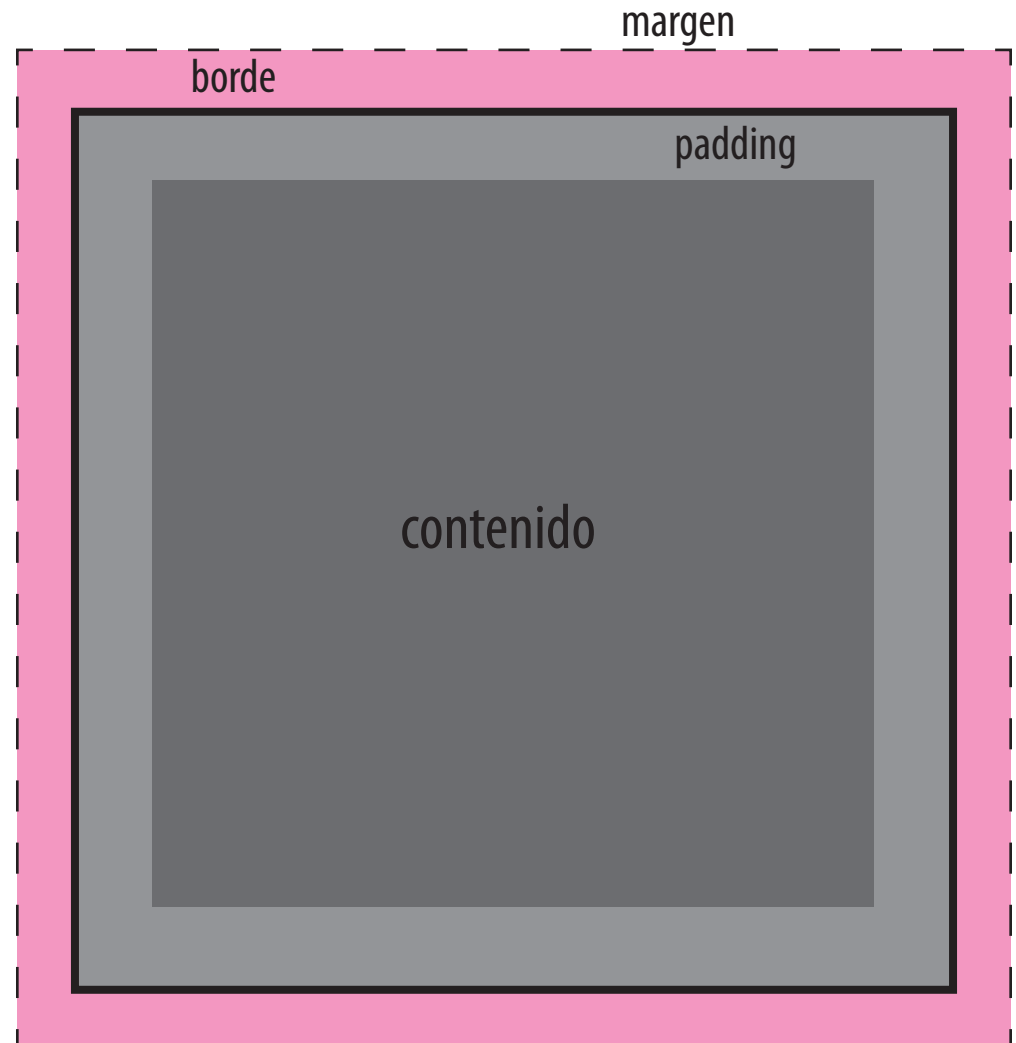
# El modelo de caja CSS

**Dichas áreas se separan por límites:**

**- Límite de margen:**

Establecido por el margen del elemento

Espacio libre entre la caja y las posibles cajas adyacentes.



# El modelo de caja CSS

El margen, el relleno y los bordes establecidos a un elemento determinan **la anchura y altura final del elemento**.

En el css la anchura es la del **contenido**, pero se debe tener presente la existencia del resto de los elementos del modelo de la caja.

# El modelo de caja CSS

Ej. Si el ancho del contenido es de 300px, el ancho final (a la vista) que presentará el elemento resulta de la suma de:

**ancho contenido (300px) + padding + borde + margen**

La altura total que ocupa el elemento se calcula de la misma forma: tomando el eje vertical (arriba y abajo) para las mismas propiedades (contenido, bordes, paddings y márgenes).

# Formato visual

Los elementos HTML se pueden clasificar de tres formas según su representación visual en el documento:

## Elementos de bloque:

Se muestran como bloques independientes, causando un salto de línea antes y después de su declaración. Por defecto ocupan el 100% del ancho del elemento que los contiene.

Ej. <div> <p> <blockquote> <h1-h6> <fieldset> <form> <ol> <ul>

# Formato visual

## Elementos de línea:

Se muestran siguiendo la línea en la que se encuentran, y no empiezan en una nueva línea, ocupan solamente el espacio que está ocupado por su contenido.

*\* Un elemento de tipo bloque, nunca puede estar contenido en un elemento de línea. Viceversa, si.*

Ej. <span> <em> <a> <img> <label> <input> <strong>

# Formato visual

## Elementos de lista:

Generan una caja principal, y otra adicional para viñetas o caracteres alfanuméricos.

Ej. <li>



# Formato visual

## Display

Controla la visualización de los elementos.

**inline** (que se visualice como elemento de línea )

**block** (que se visualice como elemento de bloque)

**none** (oculta el elemento, como si no existiera)

\* Tener cuidado cuando se especifica el valor "none" por defecto para elementos de contenido es importante, **ya que la información que contienen no estará disponible para lectores de pantalla al cargar.**

# Formato visual

## Display

Controla la visualización de los elementos.

inline-block

table

table-row

table-cell

list-item

# margin:

*margin-top / margin-right / margin-bottom / margin-left*

*margin: margin-top / margin-right / margin-bottom / margin-left*

***margin: 10px 20px 30px 10px;***

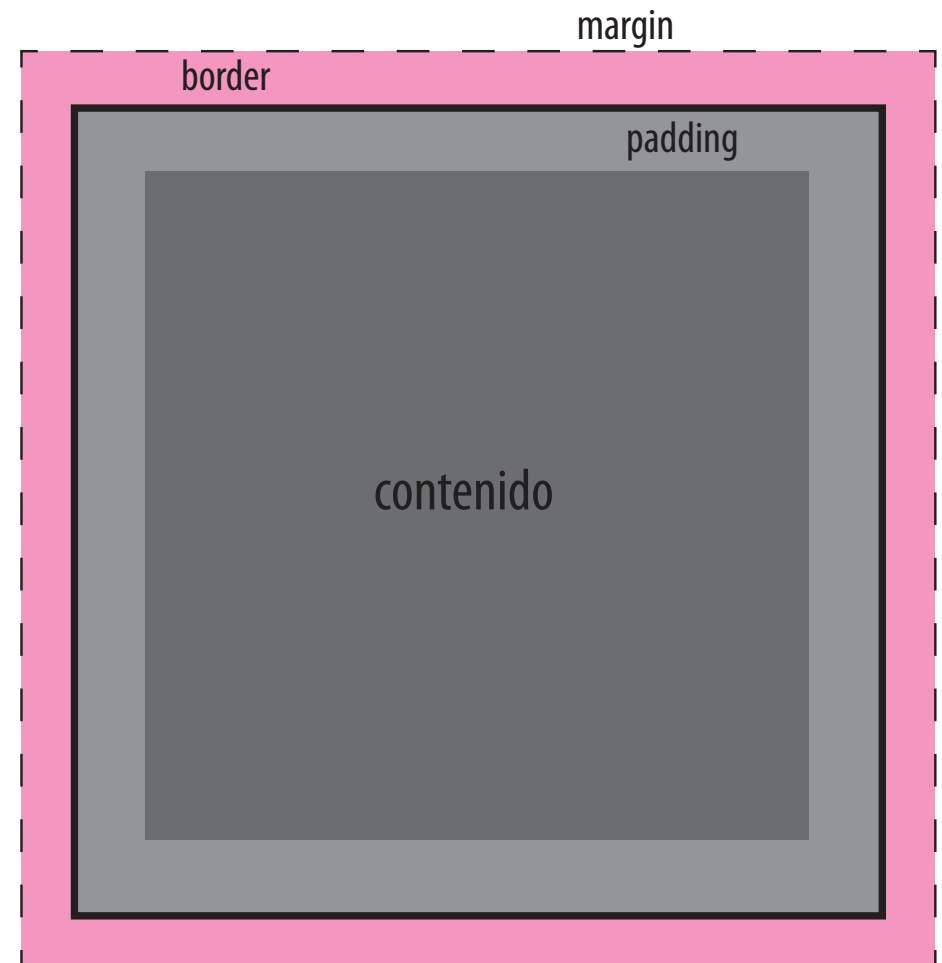
*(10px superior, 20px derecha,  
30px inferior, 10px izquierda)*

***margin: 10px 20px;***

*(10px superior e inferior, 20px derecha e izquierda)*

***margin: 10px;***

*(10px superior, inferior, derecha, izquierda)*



**border-width:** medium / thin / thick / length

**border-style:**

dotted / dashed / solid / double / groove / ridge / inset / outset

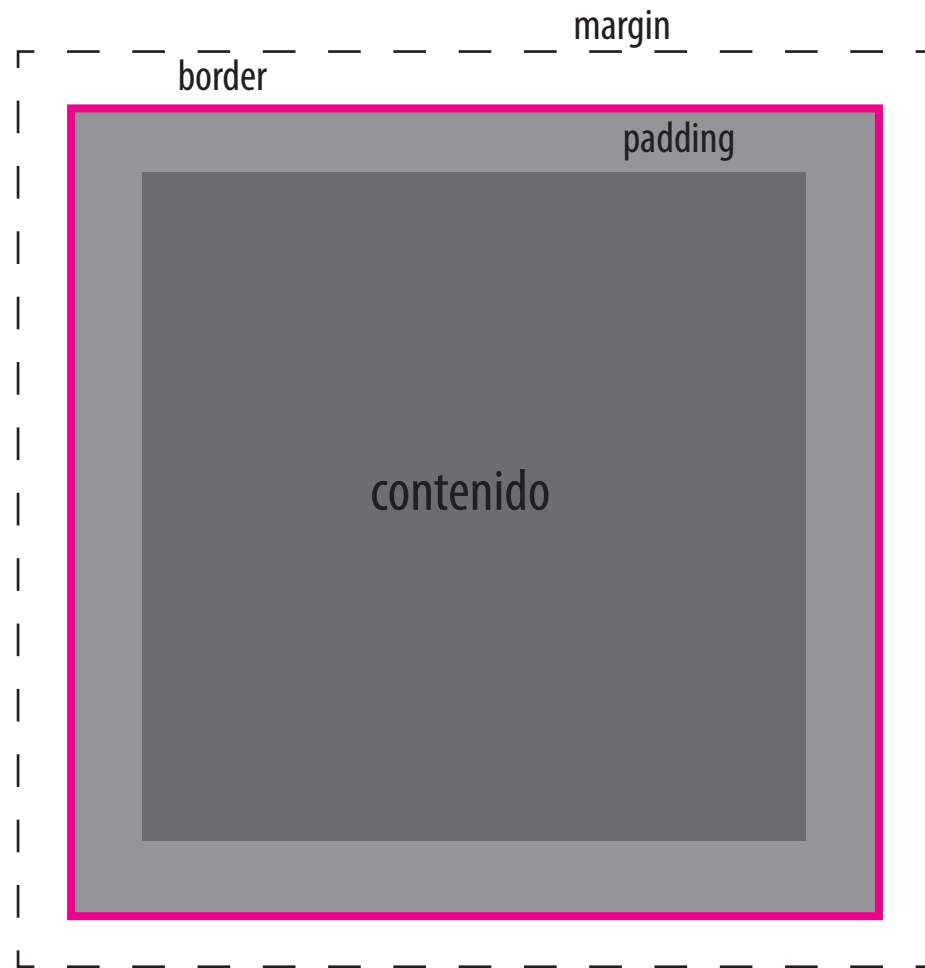
**border-color:** red / #00ff00 / rgb(0,0,255)

**border:**

- Abreviatura de las propiedades de borde
- El orden de los valores es importante: border-**width** border-**style** border-**color**
- Cualquiera de estos valores puede ser omitido, pero se debe respetar el orden

*border: 5px solid red;*

*border: 1px double #000;*



## **\*esquinas redondeadas**

*border-radius:20px;*

*border-radius: 20px 10px 30px 50px;*

*border-radius: 20px 50px;*

*width:200px;*

*height:200px;*

*border-radius: 100px;*

## \*cajas con sombra

*box-shadow: #999 5px 5px;*

*box-shadow: #999 5px 5px 7px;*

*box-shadow: #999 -5px -5px 7px;*

*box-shadow: #999 0 0 15px;*

*box-shadow: #999 5px 5px 7px inset;*

# padding:

*padding-top / padding-right / padding-bottom / padding-left*

*padding: padding-top / padding-right / padding-bottom / padding-left*

***padding: 10px 20px 30px 10px;***

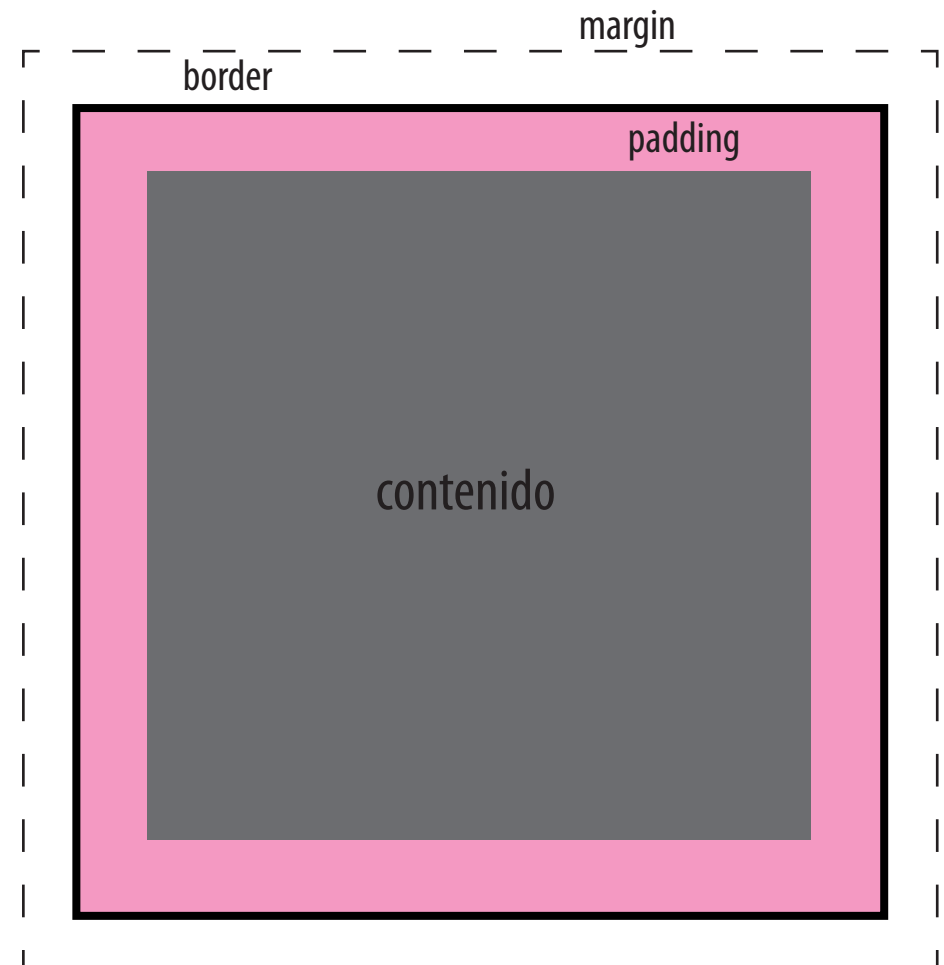
*(10px superior, 20px derecha,  
30px inferior, 10px izquierda)*

***padding: 10px 20px;***

*(10px superior e inferior, 20px derecha e izquierda)*

***padding: 10px;***

*(10px superior, inferior, derecha, izquierda)*





# width/height

## float:

Causa que elemento sea flotante del lado marcado.

Valores posibles: left; right; none.

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

**Una vez colocado el elemento, lo desplaza todo lo posible hacia la izquierda o derecha, y el resto de los elementos de bloque se distribuyen alrededor adaptando sus contenidos.**

# clear:

Especifica si el elemento puede tener elementos flotantes alrededor.

Valores posibles: both; left; right; none.

```
h1 {clear: both;}
```

\* **clear:left;** (se fuerza a que un elemento no fluya desde su lado izquierdo alrededor de otro elemento posicionado mediante float, el elemento se desplaza de forma hasta que pueda colocarse en una línea en la que no haya ningún elemento en el lado izquierdo).

\* **clear:both;** (desplaza el elemento de forma descendente hasta asegurar que el borde superior se encuentre por debajo de cualquier borde inferior de los elementos desplazados mediante float).

# position:

Esta propiedad controla la posición de los elementos en la página

**static** (normal)

**relative** (relativo)

**absolute** (absoluto)

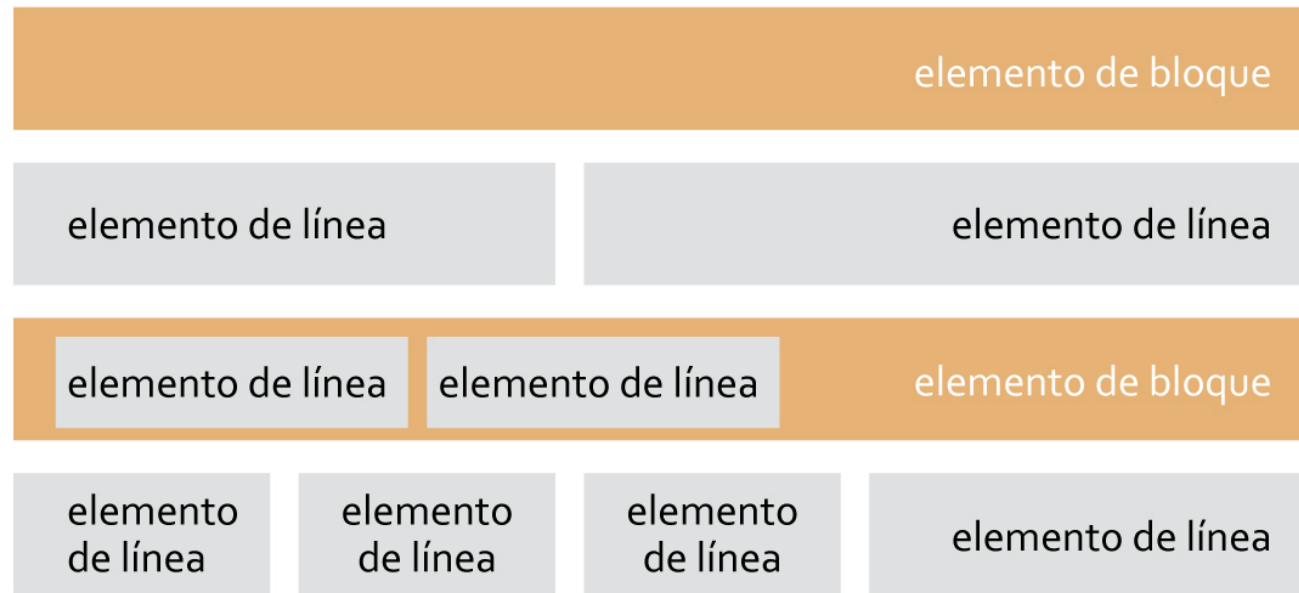
**fixed** (fijo)

# position:

## **Position static (normal)**

Por defecto, dependiendo si son elementos de línea o de bloque, los posiciona como corresponde.

\*No se tienen en cuenta los valores de las propiedades top, right, bottom y left.



Elementos de **bloque**: las cajas se muestran una debajo de otra comenzando desde el principio del elemento padre y la distancia entre cajas se controla mediante los márgenes verticales.

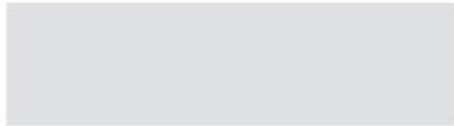
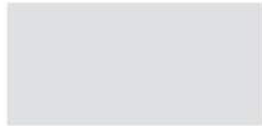
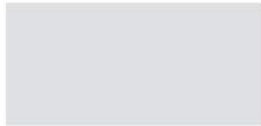
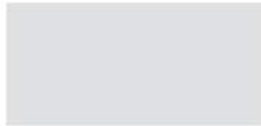
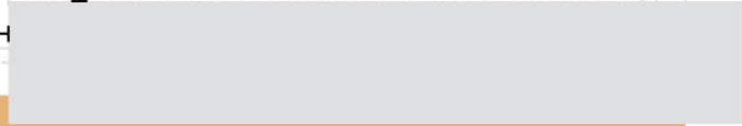
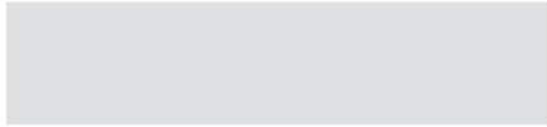
Elementos en **línea**: las cajas se muestran una detrás de otra de forma horizontal comenzando desde la posición más a la izquierda del elemento padre y la distancia entre las cajas se controla mediante los márgenes laterales.

# position:

## Position relative

Desplaza un elemento respecto de su posición original, esto no afecta al resto que se muestran en la misma posición como si la caja desplazada no se hubiera movido.

\*Se indica el desplazamiento horizontal y vertical del elemento desde el punto de referencia, desde donde se encontraba en position: static



# position:

## Position absolute

Extrae el elemento de su posicionamiento normal y lo coloca en la posición indicada respecto de su elemento padre\*\*\*.

\*La posición del resto de los elementos se determina como si no existiera el elemento desplazado. El elemento se desentiende por completo del posicionamiento normal.





# position:

## **Position fixed**

Los elementos se muestran de forma estática en la ventana del navegador y no varían su posición aunque se desplace la ventana del navegador o se haga scroll sobre ella.

\*Impresión

**Otras propiedades...**

## overflow:

Especifica como el contenido que desborda de la caja debe ser manejado.

*visible (ver contenido); hidden (ocultar contenido desbordado); scroll (proveer mecanismo de desplazamiento); auto*

## visibility:

Especifica si el elemento es visible. El espacio del elemento es reservado en cualquier caso.

*visible; hidden*

## z-index:

Especifica si el elemento es visto por encima de elementos en capas superiores.

*auto (se apilan en el orden en que aparecen en el código) o un entero.*

*Los elementos con mayor número ocultan a los de más bajo.*