



CSS

Introducción CSS

- En HTML 4 hay dos categorías de elementos
 - Elementos en bloque
 - Elementos en línea
- Otras categorías HTML5 (son muchas y las veremos a parte)

Elementos en bloque

- Un elemento en bloque ocupa todo el espacio de su elemento padre (contenedor), creando así un "bloque"
- Tienen un a nueva línea antes y después del elemento
- Los elementos de bloque sólo deben aparecer dentro del elemento `<body>`

Ejemplo elemento bloque

<p style="background-color: #FF0000;">

Este párrafo es un elemento en bloque; este fondo se ha coloreado para mostrar elemento principal (o padre) del párrafo.

</p>

Elementos en bloque mas comunes

- `<address>` html5
- `<article>`html5
- `<aside>`html5
- `<audio>`html5
- `<blockquote>`
- `<canvas>`html5
- `<dd>`
- `<div>`

Elementos en bloque mas comunes

- `<dl>`
- `<fieldset>`
- `<figcaption>html5`
- `<figure>html5`
- `<footer>html5`
- `<form>`
- Todos los `<h1>< h6>`
- `<header>html5`
- `<hgroup>html5`
- `<hr/>`

Elementos en bloque mas comunes

- ``
- `<main>`
- `<nav>`
- `<noscript>`
- ``
- `<output>html5`
- `<p>`
- `<pre>`
- `<section>html5`
- `<table>`

Elementos en bloque mas comunes

- `<tfoot>`
- ``
- `<video>`html5
- Todos los elementos constan de etiqueta de fin `</...>`

Elemento en Línea

- Un elemento en línea ocupa sólo el espacio delimitado por las etiquetas que definen el elemento en línea
- Los elementos en línea no comienzan con la nueva línea
- Los elementos en línea pueden contener únicamente los datos y otros elementos en línea

Ejemplo elemento en línea

<p>Este

span es un elemento en línea; este fondo se ha coloreado para mostrar el principio y fin de la influencia del elemento en línea</p>

Elementos en Línea mas comunes

- ``
- `<big>`
- `<i>`
- `<small>`
- `<tt>`
- `<abbr>`
- `<acronym>`
- `<cite>`

Elementos en Línea mas comunes

- `<code>`
- `<dfn>`
- ``
- `<kbd>`
- ``
- `<samp>`
- `<time>`
- `<var>`

Elementos en Línea mas comunes

- `<a>`
- `<bdo>`
- `
`
- ``
- `<map>`
- `<object>`
- `<q>`
- `<script>`

Elementos en Línea mas comunes

- ``
- `<sub>`
- `<sup>`
- `<button>`
- `<input>`
- `<label>`
- `<select>`
- `<textarea>`

Modelo de cajas

- Es el que hace que todos los elementos se representen como cajas rectangulares
- Cada vez que se inserta una etiqueta se crea una nueva caja
- Las cajas pueden ser de bloque o de línea

`<p>Párrafo de texto con algunas palabras resaltadas</p>`

`<p>Otro párrafo</p>`

Párrafo de texto con algunas palabras resaltadas

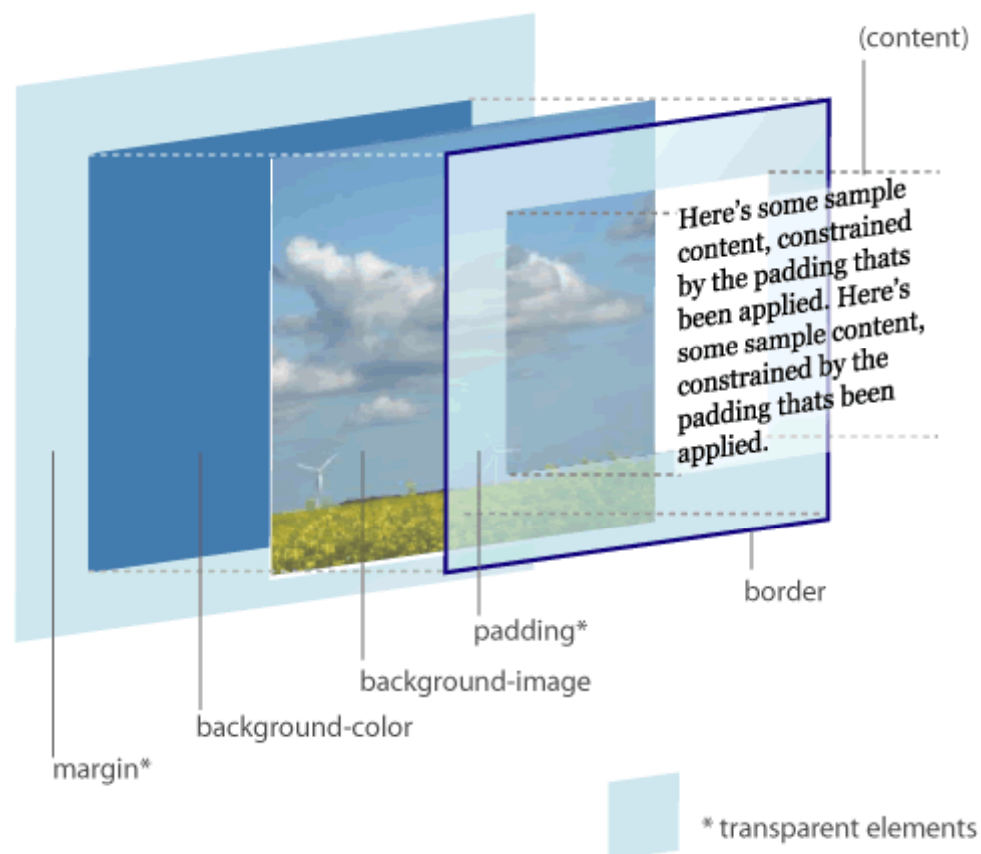
Otro párrafo

Modelo de cajas

- Las cajas no son visibles a simple vista
- Los navegadores colocan las cajas automáticamente
- Nosotros podemos modificar todas sus características con CSS

Modelo de cajas

THE CSS BOX MODEL HIERARCHY



<http://www.hicksdesign.co.uk/boxmodel/>

Agrupación de elementos

- Tenemos dos etiquetas para agrupar elementos y usarlas en css
- `` es un elemento neutro que sirve para añadir capacidades distintivas a los elementos de texto a los que se aplica. (elemento de linea)
- `<div></div>` se usa para crear bloques con uno o mas elementos de HTML

Etiqueta de línea

- No añade nada en si al documento
- Con CSS se puede usar para añadir características visuales distintas a partes específicas de texto en los documentos

Etiqueta de bloque <div>

- No añade nada nuevo al documento
- Sirve para crear secciones o agrupar contenidos
- Crea una caja de bloque

Etiqueta de bloque <div>

- Atributos
 - title misma función que alt
 - id asigna un identificador tiene que ser único
 - class asigna nombre de clases como una lista de nombres separadas por espacio
 - style permite especificar información de estilo
 - lang informa sobre el idioma del contenido y del valor de sus atributos
 - dir indica la dirección del texto y tablas
 - align "left", "center", "right", o "justify"

Etiqueta de bloque <div>

- Eventos
 - onclick,
 - ondblclick,
 - Onmousedown,
 - Onmouseup,
 - onmouseover,
 - Onmousemove,
 - Onmouseout,
 - Onkeypress,
 - Onkeydown,
 - onkeyup.

Funcionamiento de CSS básico

- A toda etiqueta HTML se le puede asociar estilos
- El esquema es:
 - selector {property: value;}
 - Selector indica a que etiquetas se le aplica el estilo
 - Propiedad de la etiqueta
 - Un valor de la propiedad

Formas de escribir CSS

- En línea
- En cabecera
- En fichero independiente

CSS en línea

- Mediante el atributo style="propiedad: valor;" dentro de la etiqueta en cuestión

```
<html>
<head> <title>Ejemplo</title> </head>
  <body style="background-color: #FF0000;">
    <p>Esta es una página con fondo rojo</p>
  </body>
</html>
```

CSS interno

- Usamos la etiqueta `<style></style>` en el header y ponemos todos los selectores que queramos

```
<html>
<head>
  <title>Ejemplo</title>
  <style type="text/css">
    body {background-color: #FF0000;}
  </style>
</head>
<body>
  <p>Esta es una página con fondo rojo</p>
</body>
</html>
```

CCS en fichero independiente

- Tenemos que generar un fichero con los contenidos CSS debe de tener extensión .css
- Tenemos que asociar el fichero css mediante in link del head

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

fichero.html

```
<html>
<head>
  <title>Mi documento</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
  <h1>Mi primera hoja de estilo</h1>
</body>
</html>
```

style.css

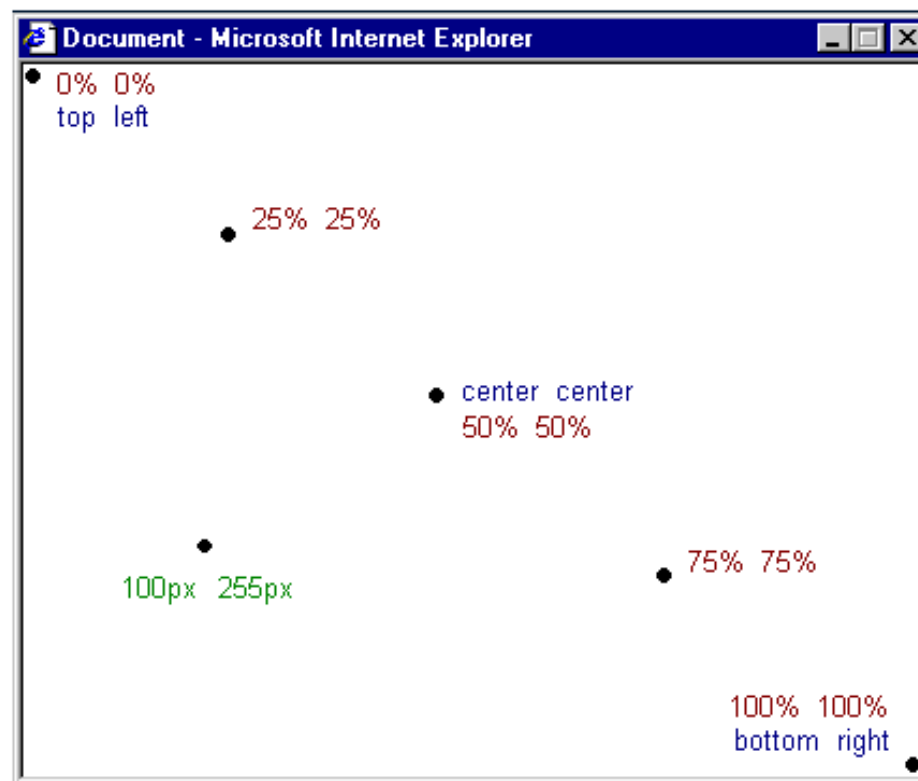
- `body { background-color: #FF0000; }`

Colores y fondos

- color
- background-color
- background-image
 - url("URL")
- background-repeat
 - repeat-x
 - repeat-y
 - repeat
 - no-repeat
- background-attachment
 - scroll
 - fixed

Colores y fondos

- background-position



Colores y fondos

- Background permite escribir todas las propiedades anteriores de forma resumida.

De forma extendida sería

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}
```


Colores y fondos

- Lo mismo se puede conseguir con solo una línea
- Se debe de respetar el orden siguiente

[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]

La forma resumida sería:

```
Body {  
    background: #FFCC66 url("butterfly.gif") no-repeat fixed right  
    bottom;  
}
```

CSS fuentes

- font-family lista las posibles fuentes favoritas la última debería ser una familia generica sans-serif, monospace, serif...
- font-style
 - Normal/italic/oblique
- font-variant
 - Normal
 - Small-caps primera mayúscula mas grande el resto mas pequeñas
- font-weight
 - normal o bold
- font-size
 - Usar % o em para que sea accesible

CSS fuentes

- Font es la etiqueta que permite escribir de forma resumida todas las características anteriores según el orden:

font-style|font-variant | font-weight | font-size | font-family

p { font: italic normal bold +30px arial, sans-serif; }

CSS texto

- text-indent sangría del texto en:
 - Longitud
 - px,pt,cm
 - em(relativa a la altura del tamaño de fuente)
 - ex relativo a la altura de la letra(x-height)
 - rem relat a la fuent del ele raíz
 - % del tamaño del padre
 - Pc picas (1pica=12puntos)
 - Pt puntos (1/72 pulgadas)
 - Initial por defecto del sistema
 - Inherit hereda la propiedad del padre
- text-align right | center | justify
- text-decoration none|underline|overline|line-through|initial|inherit;
- letter-spacing
- text-transform capitalize | uppercase | lowercase | none

CSS enlaces

- Podemos definir las propiedades dependiendo del estado del enlace
- Lo aplicamos usando las subclases:
 - `a:link {...}` cuando no se han visitado
 - `a:visited {...}` ya visitados
 - `a:hover {...}` pasa por encima
 - `a:active {...}` que están activos
- El orden importa debe de ser el anterior

Identificación y agrupación de elementos

- Por un lado podemos aplicar estilos a las etiquetas de HTML directamente
- Por otro, para poder elegir a quien queremos aplicar estilo tenemos que identificar elementos
 - ◉ El atributo id identifica elementos de forma única
 - ◉ `<elementoHTML id="nombreId" ...>`
- Para aplicar un estilo a varios elementos tenemos que agruparlos
 - ◉ El atributo class agrupa elementos
 - ◉ `<elementoHTML class="nombreClase" ...>`

Identificación y agrupación de elementos

- Para aplicar estilo a etiquetas usamos su nombre directamente en el selector, de dos maneras:
 1. `nomEtq, nomEtq, ... {` ← se aplica a todas la etiquetas el mismo estilo
`/*estilo*/`
`}`
 2. `nomEtq1 nomEtq2... {` ← El estilo se aplica al elemento `nomEtqN` si es descendiente de `nomEtq(n-1)`
`/*estilo*/`
`}`
 3. `* {` ← Todos los elementos
`/* estilos */`
`}`

Identificación y agrupación de elementos

- Para las clases en la hoja de estilo tenemos que usar el nombre de la clase con punto delante
 - `a.nombreClase1 {` ← Los elementos con una clase de valor `nombreClase1`
`/* estilos */`
`}`
- Para los identificadores usamos la almohadilla antes del nombre del identificador
 - `#nombreId {` ← Solo la etiqueta con el id `nombreid`
`/* estilos */`
`}`

Identificación y agrupación de elementos

- Para mezclar clases y pseudoclases usamos

```
a.nombreClase1: hover {  
    /* estilos */  
}
```

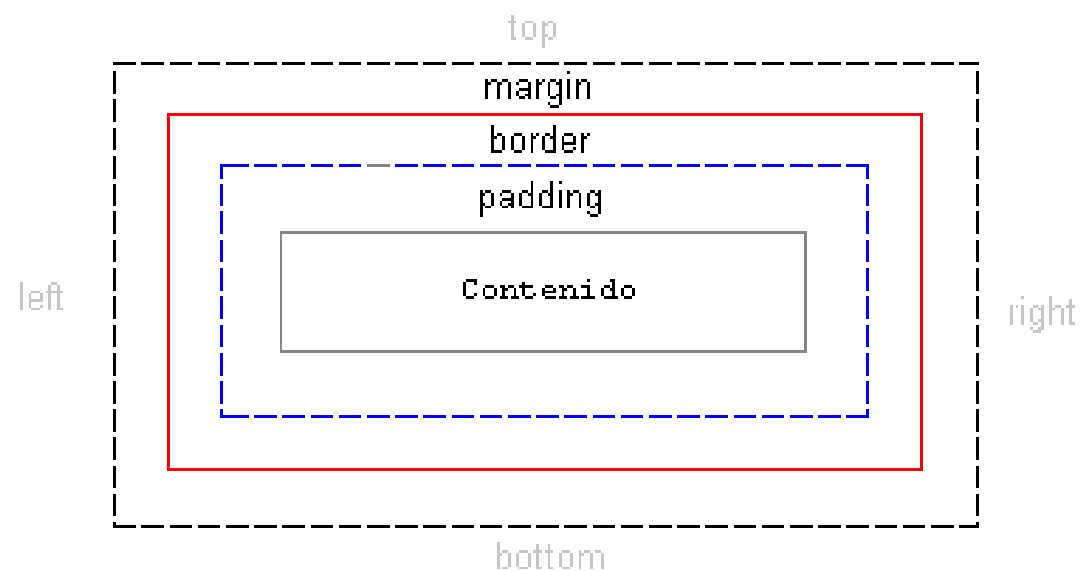
- Para definir una clase no asociada a un elemento usamos

```
.nombreclase {  
    /* estilos */  
}
```



```
*.nombreclase {  
    /* estilos */  
}
```

Repaso modelo caja



- límite del margen
- límite del borde
- - - límite del relleno (padding)
- límite del contenido

Código html

`<h1>Artículo 1:</h1>`

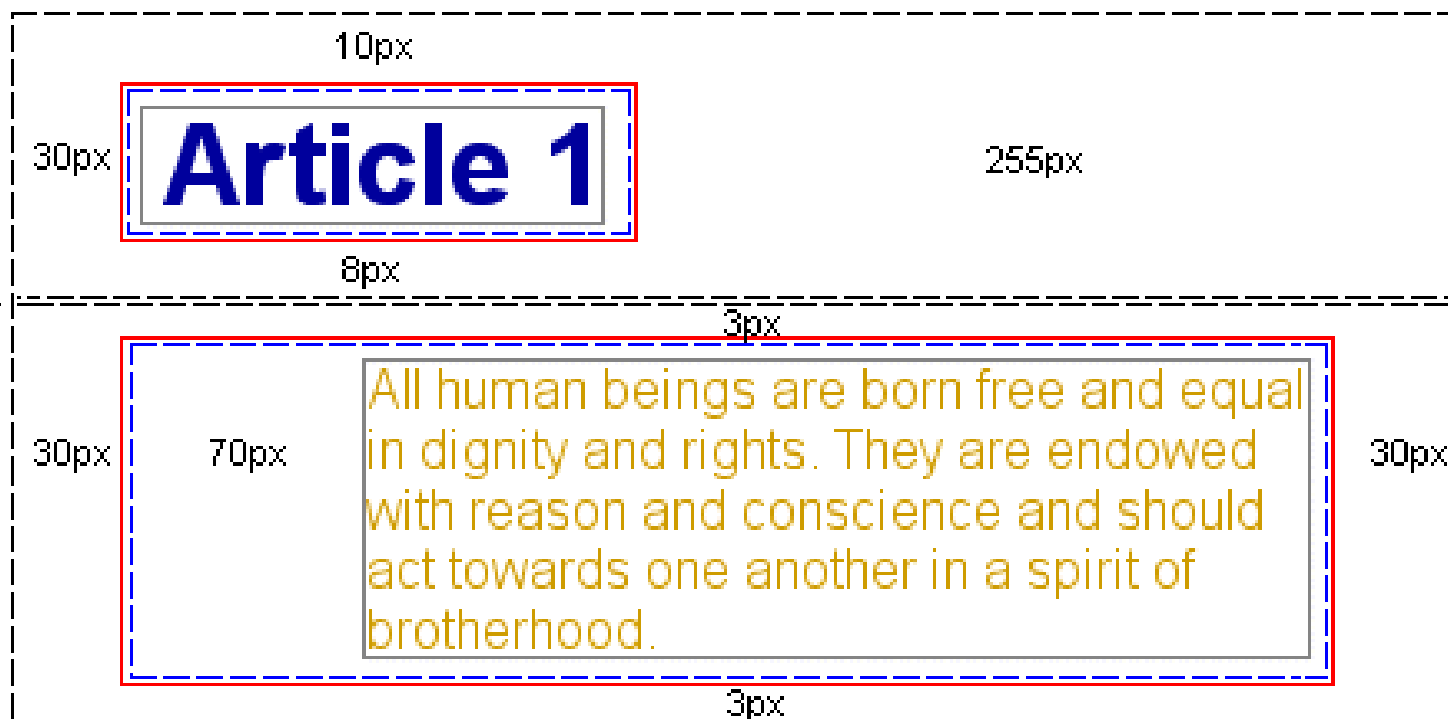
`<p>Todos los hombres nacen libres e iguales en
dignidad y derechos. Están dotados de razonamiento
y consciencia y deberían de comportarse entre sí
con espíritu de hermandad.</p>`

Vista html

Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

Modelo caja html



Repaso modelo caja

- Los elementos del código HTML esta rodeado por cajas cuando lo visualizamos usando el navegador.
- Las cajas son invisibles a primera vista pero existen
- Estas cajas las podemos ajustar a nuestro antojo usando CSS

Controlar el modelo caja

- Margen:
 - Todo elemento tiene cuatro lados: derecho, izquierdo, superior e inferior
- Ej:

```
body {  
    margin-top: 100px;  
    margin-right: 40px;  
    margin-bottom: 10px;  
    margin-left: 70px;  
}
```
- Otra forma:

```
body { margin: 100px 40px 10px 70px; }
```

Controlar el modelo caja

- Relleno:
 - Padding el relleno o distancia entre el contenido y el borde

Ej:

```
body {  
    margin: 100px 40px 10px 70px;  
}  
  
p {  
    margin: 5px 50px 5px 50px;  
}
```


Controlar el modelo caja

- Bordes:
 - border-width anchura del borde
 - Thin,medium, thick o en px
 - border-color color del borde
 - Exadecimal #123456, rgb(123,123,123), nombre del color "yellow"
 - border-style
 - none=hidden,dotted, dashed, solid, double, groove, ridge,inset,outset

Ejemplo de bordes

```
h1 {  
    border-top-width: thick;  
    border-top-style: solid;  
    border-top-color: red;  
  
    border-bottom-width: thick;  
    border-bottom-style: solid;  
    border-bottom-color: blue;  
  
    border-right-width: thick;  
    border-right-style: solid;  
    border-right-color: green;  
  
    border-left-width: thick;  
    border-left-style: solid;  
    border-left-color: orange;  
}
```

El borde se puede resumir poniendo: `border: 12px solid blue;`

Controlar el modelo caja

- Altura, con height podemos definir la altura de un elemento
- Anchura, con width podemos definir la anchura de un elemento

Ej:

```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Controlar el modelo caja

- Muy interesante es la propiedad que permite flotar las cajas a la izquierda o a la derecha
 - float: left|right|none
- Se hace flotar a uno de los lados el elemento que esta definido primero en el HTML de modo que el que viene detrás ocupara el espacio que deja vacio

Ejemplo float 1/1

- Imagen rodeada de texto:

```
<div id="imagen">
```

```
  
```

```
</div>
```

```
<p>El veloz murciélago hindú comía feliz cardillo y  
kiwi. La cigüeña toca el saxofón detrás del palenque  
de paja. 1234567890...</p>
```

Ejemplo float 1/2

- Aplicar esta css

```
#imagen{  
    float:left;  
    width: 100px;  
}
```

Ejemplo 2/1

```
<div id="column1">
```

```
  <p>Haec disserens qua de re agatur et in quo causa consistat non  
  videt...</p>
```

```
</div>
```

```
<div id="column2">
```

```
  <p>causas naturales et antecedentes, idcirco etiam nostrarum  
  voluntatum...</p>
```

```
</div>
```

```
<div id="column3">
```

```
  <p>nam nihil esset in nostra potestate si res ita se haberet...</p>
```

```
</div>
```

Ejemplo 2/2

```
#column1 {  
    float:left;  
    width: 33%;  
}
```

```
#column2 {  
    float:left;  
    width: 33%;  
}
```

```
#column3 {  
    float:left;  
    width: 33%;  
}
```


Controlar el modelo caja

- Para evitar que un elemento flote se usa clear
 - Clear: left, right, both y none

Ejemplo NO Float HTML

- Imagen NO rodeada de texto:

```
<div id="imagen">
```

```
  
```

```
</div>
```

```
<p class="nofloat">El veloz murciélago hindú comía  
feliz cardillo y kiwi. La cigüeña toca el saxofón detrás  
del palenque de paja. 1234567890...</p>
```

Ejemplo NO Float CSS

```
#imagen{  
    float:left;  
    width: 100px;  
}
```

```
.nofloat{  
    clear:both;  
}
```

Posicionamiento CSS

- Absoluto desde los bordes de la ventana de navegador, el hueco lo ocupa el siguiente
 - Position: absolute;
 - Podemos usar después las propiedades: top, left, right y bottom con pixel
- Relativo se posiciona partiendo de su posición original, el hueco se respeta como si estuviera todavía
 - Position: relative;
 - Podemos usar después las propiedades: top, left, right y bottom con pixel

Posicionamiento CSS

- Static no hace caso de las coordenadas
 - Position: static;
 - Podemos usar después las propiedades: top, left, right y bottom con pixel
- Fixed se desplaza donde tiene que desplazarse como absolute pero se queda fijo cuando hacemos scrolling
 - Position: fixed;
 - Podemos usar después las propiedades: top, left, right y bottom con pixel

Orden de capas

- CSS funciona sobre tres dimensiones alto, ancho y profundidad
- Para convertir una caja en una capa hay que darle profundidad con z-index
- La capa con z-index menor es la mas profunda y por lo tanto se ve solapada por la de menor profundidad z-index mayores

Trabajando con el selector

- El selector es la parte mas importante del modelo CSS ya que es quien selecciona los elementos de HTML a los que les tenemos que aplicar los estilos.
- Todos los elementos de HTML forman parte del selector.
- Cuando seleccionamos un elemento CSS pasa a ser una caja y es a esta y a su contenido a la que damos formato en CSS

Trabajando con el selector

- Hemos visto como usamos el selector de forma básica.
- Ahora vamos a ver selectores mas avanzados.
- Podemos ver que selectores podemos usar con nuestro navegador favorito usando el enlace:
 - <http://tools.css3.info/selectors-test/test.html>

Selectores avanzados

- * {...} comodín todos los elementos
- Descendientes selecciona todos los elementos que están 'dentro de:' independientemente del nivel.
 - Elemento elemento { ... }
- Pseudoclases generalmente son selectores relacionadas con eventos vimos algunas y añadimos otras:
 - :focus campos de formularios
 - :lang(es) selecciona elementos en función de su idioma

Selectores avanzados

- PseudoElementos Permite seleccionar cierto tipo de elementos:
 - :first-line { ... } selecciona la primera línea de texto
 - p:first-line { text-transform: uppercase; }
 - :first-letter { ... } selecciona la primera palabra de l texto.
 - p:first-letter { text-transform: uppercase; }
 - :before :after { ... } para poner contenido nuevo antes o después de un elemento. Se usa junto con la propiedad *'content'*

Selectores avanzados

- Ejemplo de `:before` y `:after`
 - `h1:before { content: "Capítulo - "; }`
 - `p:after { content: "."; }`
 - `p:first-letter { text-transform: uppercase; }`
- Con CS3 el selector de elementos es con doble dos puntos.
- `::after`
- `::before`
- `::first-letter`
- `::first-line`

Selectores avanzados

- Hijos directos: parecido al selector descendente, con la salvedad de que es solo para hijos directos.
 - ejemplo1

```
p > span { color: blue; }  
<p><span>Texto1</span></p>  
<p><a href="#"><span>Texto2</span></a></p>
```
 - ejemplo2

```
p a { color: red; }  
p > a { color: yellow; }  
<p><a href="#">Enlace1</a></p>  
<p><span><a href="#">Enlace2</a></span></p>
```

Selectores avanzados

- Selector adyacente para seleccionar elementos que están justo a continuación de otros elementos.

elemento1 + elemento2 { ... }

Selectores avanzados

- Selector de atributos selecciona elementos en función de sus atributos.
 - [nombre_atributo], selecciona los elementos que tienen establecido el atributo llamado nombre_atributo, independientemente de su valor.
 - [nombre_atributo=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo con un valor igual a valor.
 - [nombre_atributo~=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y cuyo valor es una lista de palabras separadas por espacios en blanco en la que al menos una de ellas es exactamente igual a valor.
 - [nombre_atributo|=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con valor. Este tipo de selector sólo es útil para los atributos de tipo lang que indican el idioma del contenido del elemento.

Selectores avanzados

- `a[class] { color: blue; }`
- `a[class="externo"] { color: blue; }`
- `a[href="http://www.ejemplo.com"] { color: blue; }`
`/*que apunten a*/`
- `a[class~="externo"] { color: blue; }` `/*al menos uno sea..*/`
- `*[lang]="es"] { color : red }` `/*todos los elementos de la pagina cuyo atributo 'lang' empiece por 'es'*/`

Selectores nuevos CSS3

- `elemento[atributo^="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor comienza exactamente por la cadena de texto indicada.
- `elemento[atributo$="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor termina exactamente por la cadena de texto indicada.
- `elemento[atributo*="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor contiene la cadena de texto indicada.

Altura/anchura max/min

- Para controlar los tamaños máximos y mínimos de los contenedores.
- max/min-width
- Max/min-height
- Ej:
- `#contenedor { min-width: 500px; max-width: 900px; }`
-

anchura max/min

```
div {  
  max-width: 800px;  
  width: expression(document.body.clientWidth > 801?  
    "800px": "auto");  
  /* ----- */  
  min-width: 800px;  
  width: expression(document.body.clientWidth < 801?  
    "800px": "auto" );  
}
```

Propiedades shorthand

- son propiedades de CSS que permiten establecer de forma simultánea el valor de varias propiedades diferentes pero relacionadas.
- **Background** (background-color|| background-image || background-repeat || background-attachment || background-position) | inherit

Propiedades shorthand

- font ((font-style || font-variant || font-weight)? font-size (/ line-height)? font-family) | caption | icon | menu | message-box | small-caption | status-bar | inherit
- , margin, padding, border,
- List-style (list-style-type || list-style-position || list-style-image) | inherit
- * Inherit fuerza la herencia.

Personalizar el cursor

- `cursor ((url ,)* (auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress)) | inherit`
- Ej:
`:link, :visited { cursor: url(puntero.svg), url(puntero.cur), pointer }`
- <http://www.echoecho.com/csscursors.htm>

Hacks y filtros

- A pesar de que utilizar filtros y *hacks* es una solución poco ortodoxa, en ocasiones es la única forma de conseguir que una página web muestre un aspecto idéntico en cualquier navegador.
- los filtros permiten definir u ocultar ciertas reglas CSS para algunos navegadores específicos.
- Los filtros se definen aprovechando los errores de algunos navegadores (sobre todo los antiguos) a la hora de procesar las hojas de estilos.
- <http://css-discuss.incutio.com/?page=CssHack>

Ejemplo hacks/filtro

- solamente carga la hoja de estilos basico_ie6.css si el navegador es la versión 6 o superior de Internet Explorer

```
<!--[if gte IE 6]>
```

```
<style type="text/css"> @import ("basico_ie6.css");
```

```
</style>
```

```
<![endif]-->
```

Ayudas en diseño

- <http://alistapart.com/articles/fauxcolumns/>
- <http://www.positioniseverything.net/articles/onetruelayout/>
- <http://www.fu2k.org/alex/css/onetruelayout/example/interactive>

Altura max/min

```
Div {  
  max-height: 300px;  
  overflow: hidden;  
  height: expression(this.scrollHeight > 301? "300px" : "auto" );  
  /* ----- */  
  min-height:300px;  
  overflow: hidden;  
  height: expression(this.scrollHeight < 301? "300px" : "auto" );  
}
```

Ejercicio CSS

- Centrar una pagina Horizontalmente:

```
#contenedor { width: 300px; margin: 0 auto; } <body>  
  <div id="contenedor"> <h1>Lorem ipsum dolor sit  
    amet</h1> ... </div> </body>
```

- Modelo fluido:

```
#contenedor { width: 70%; margin: 0 auto; }
```

Ejercicio CSS

- Centrar pagina Verticalmente:

Ejercicio Css

- Estructuras y layout

Resolución de colisiones

- Como regla general podemos decir que se aplican estilos según:
 - Cuanto más específico sea un selector, más importancia tiene su regla asociada.
 - A igual *especificidad*, se considera la última regla indicada.
- Mas adelante veremos la regla global teniendo en cuenta las reglas de navegador, usuario y diseñador.

Resolución de colisiones

- Los navegadores aplican al HTML dos hojas de estilos diferentes:
 - CSS de navegador
 - CSS de usuario
- Estas permiten adaptar como se ‘muestra|interpreta’ el HTML
- Donde están:
 - En las siguientes diapositivas veremos donde están en cada navegador

Hojas de estilo de los navegadores

- IE
 - Herramientas→Opciones de Internet→General
→Accesibilidad→Apariencia
 - Opción →Formatear los documentos con mi hoja de estilos→y
buscamos con examinar
- Firefox
 - Crea un archivo userContent.css
 - Busca el perfil de firefox actual en la configuración del S.O.
 - `$(ConfiguracionDelSistema)\Mozilla\Firefox\Profiles\[cadena_aleatoria_de_letras_y_numeros].default`
 - Copia la hoja de estilos userContent.css en el directorio chrome de tu perfil.
 - Hay que reiniciar.

Orden de aplicación de CSS



- CSS define la palabra reservada `!important` para controlar la prioridad de las declaraciones de las diferentes hojas de estilos.

```
p { color: red !important; color: blue; }
```


Orden de aplicación de CSS



- Cuando usamos la clausula !important cambia el orden.

CSS por medios

<head>

...

<link rel="stylesheet" type="text/css" href="/css/basico.css" media="screen" />

<link rel="stylesheet" type="text/css" href="/css/seccion.css" media="screen" />

<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />

<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />

...

</head>

CSS por medios

- El archivo estilos.css contiene las llamadas a las otras hojas de estilo de la forma:

```
@import url("basico.css") screen;  
@import url("seccion.css") screen;  
@import url("impresora.css") print;  
@import url("movil.css") handheld;
```

- En el HTML llamo a estilos.css como siempre

<head>

...

```
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="all" />
```

...

</head>

Validar nuestra hoja de estilos

- Al igual que validamos en su momento nuestro HTML podemos validar nuestras hojas de estilos en:

<http://jigsaw.w3.org/css-validator/>

Tenemos generadores de CSS para probar

<http://codepen.io/edgardorito/pen/JdqpXo/>