

Índice

lunes, 19 de febrero de 2018 11:38

Introducción al CSS

1. Formato. Estilos CSS.
 - a. Elementos básicos.
 - b. DOM. Selectores. Herencia
 - c. Cascada de estilos. Precedencia. Especificidad
2. Elementos de diseño avanzados
 - a. Selectores: pseudoclases y pseudoelementos
Nuevos selectores en CSS3
 - b. Variables en CSS3. Funciones: calc()
 - c. Modelo de caja. Concepto. Modelos disponibles
 - d. Inicialización. Reseteo y normalización..
 - e. Soporte y prefijos



Formato. Estilos CSS.

Lunes, 19 de febrero de 2018 11:37

- a. Elementos básicos.
 - 1. Concepto de estilos CSS
 - 2. Sintaxis: declaraciones de estilos y comentarios
 - 3. Ubicación
 - 4. Directivas en CSS
- b. DOM. Selectores
 - 1. DOM
 - 2. Selectores
 - 3. Herencia
- c. Múltiples CSS
 - 1. Cascada de estilos. Precedencia.
 - 2. Especificidad



Formato en los documentos HTML



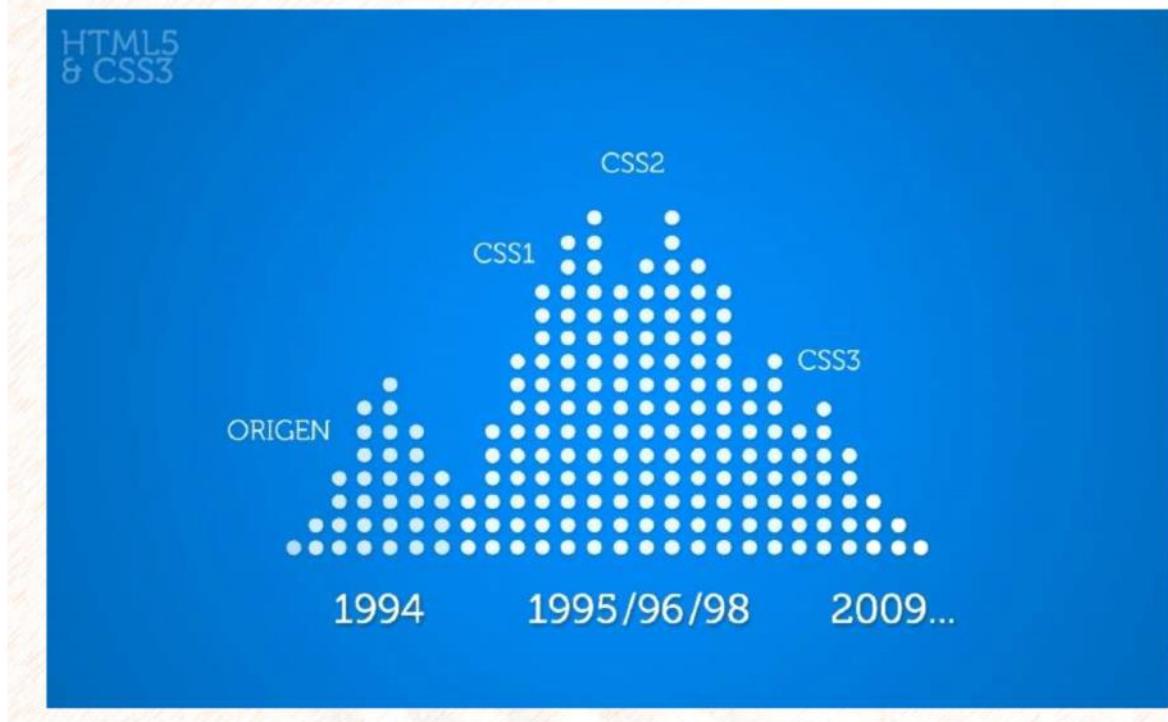
CSS Zen Garden

<http://csszengarden.com/>

Una web para mostrar como cambia radicalmente el aspecto de una Web cambiando únicamente los estilos CSS que se aplican



Estilos



Estilos

Inicialmente

HTML es un lenguaje descriptivo de la estructura del documento, en el que las etiquetas no dan formato al contenido

Más adelante,

HTML 3.2 incorpora etiquetas como `` y atributos de color que contradicen el planteamiento inicial

Como reacción

En HTML 4.0, el W3C incorpora las Hojas de Estilos en Cascada, (*Cascading Style Sheets, CSS*)

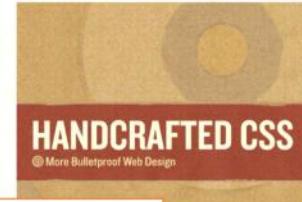
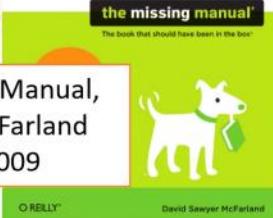
Regresamos
así al inicio

Todos los elementos relativos al formato se retiran del cuerpo del documento HTML y se incorporan a los estilos, que en el mejor de los casos pasan a un fichero independiente; CSS

Más información

"The Missing Manual series is simply the most intelligent and usable series of guidebooks."
—KEVIN KELLY, CO-FOUNDER OF WIRED

CSS



Handcrafted CSS: More
Bulletproof Web Design,
Dan Cederholm
New Riders, 2009

DAN CEDERHOLM
with Ethan Marcotte

Sintaxis de los estilos

Selector

h1

Declaración

{color:blue; font-size:12px}

Declaración

Propiedad Valor

Propiedad Valor

Propiedad color

Valor:

red, purple, fuchsia, blue, navy, teal, aqua
green, lime, olive, yellow, maroon,
silver, gray, white, black,

Propiedad font-size

Valor: ... 12px ... 16px (valor por defecto) ...

Nombre	Color
White	
Black	
Red	
Lime	
Blue	
Yellow	
Silver	
Gray	
Maroon	
Purple	
Fuchsia	
Green	
Olive	
Navy	
Teal	
Aqua	

Comentarios

texto del comentario

/* Comentario */

Barra *

* Barra

/* Esto es un comentario en CSS */

(No confundir con HTML)

<!--Esto es un comentario en HTML-->

Ubicación de los estilos

Se pueden incorporar los estilos en tres posiciones

Estilos en línea

Junto al elemento al que afectan, con el atributo **style="declaración"**

Estilos embebidos o internos

En la cabecera del documento
`<style> lista de estilos </style>`

```
<style>
h1 {color:blue; font-size:12px}
</style>
```

Estilos externos

En un fichero de extensión .css incluye la **lista de estilos**

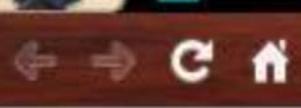
```
h1 {color:blue; font-size:12px}
```

La cabecera incluirá `<link rel="stylesheet" href="mystyle.css">`

Ejemplos de estilos

Estilo como atributo

```
<body>
<h1 style=
  "color:red;
  font-size:14px;">
Título
</h1>
</body>
```



Título

Estilo embebido

```
<head>
<style>
h1 {
  color: red;
  font-size: 14px;
}
</style>
</head>

<body>
<h1> Título </h1>
</body>
```



El resultado es el mismo en los tres casos

Fichero de estilo

```
h1 {
  color: red;
  font-size: 14px;
}
```

Fichero HTML

```
<body>
<h1> Título </h1>
</body>
```

AC3

AC5

Importación de estilos: @import

@import

directiva CSS que permite incorporar reglas desde otras hojas de estilo (como link a nivel HTML)

@import url [list-of-media-queries]

```
@import url("tables.css");
```

se pueden añadir *media queries* que filtraran la importación

```
@import url("shetland.css") screen and  
(max-device-width: 480px);
```

@import debe preceder a todas las demás reglas de estilo, excepto @charset.

Directivas CSS: @ condicionales

@media → permite definir un conjunto de reglas anidadas que sólo se aplicaran en función la *media query* indicada

```
@media screen and (max-device-width: 480px)  
{.column {float: none;} ... }
```

@page → permite modificar algunas propiedades CSS cuando se imprime el documento

@supports → permite comprobar si una condición es válida en un navegador, en cuyo caso la directiva devuelve true y ejecuta la serie de reglas anidadas indicada

@document → restringe el uso de las reglas en función de la url del documento (experimental)

Otras directivas CSS @

@charset → especifica el conjunto de caracteres utilizado en la hoja de estilos:
debe ser siempre el primer elemento que aparece en ella)

```
@charset "UTF-8";
```

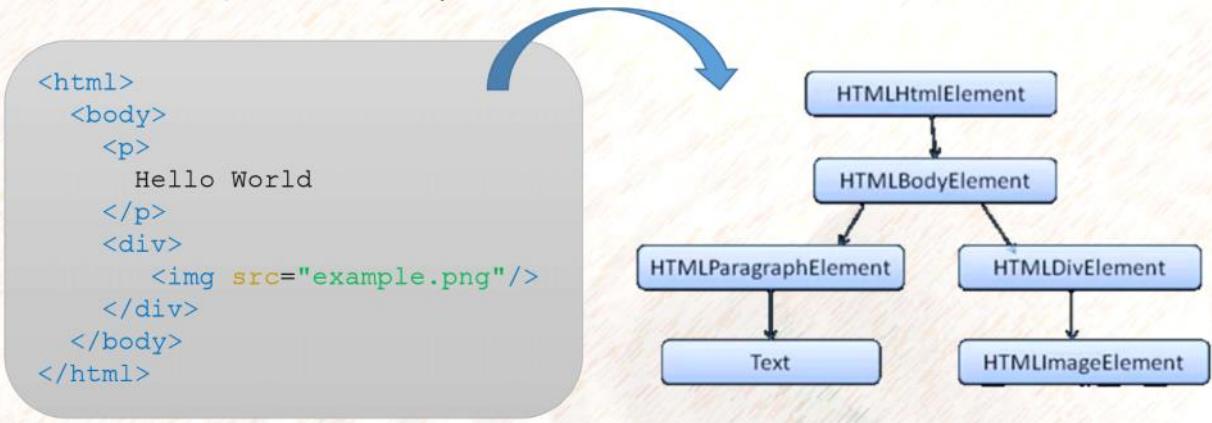
Relativas a elementos concretos (fuentes, animaciones)

@font-face → permite especificar fuentes online para
visualizar en las páginas web.

@keyframes → permite controlar los pasos intermedios en una
secuencia de animación CSS (experimental)

DOM (*Document Object Model*)

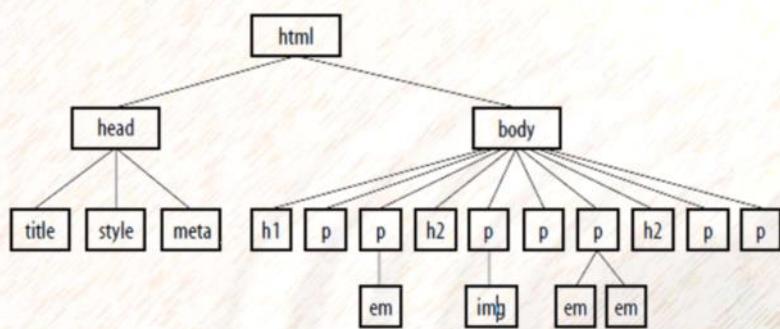
DOM (**modelo de objetos del documento**) es la representación estándar de los objetos del documento HTML y la interfaz (API) de los elementos HTML para el mundo exterior, como JavaScript



El árbol de salida ("árbol de análisis") de un motor de renderización está formado por nodos: elementos y atributos del DOM.

Relaciones entre elementos del DOM

Desde el punto de vista de CSS, la estructura del documento es una versión simplificada del DOM en la que se definen las relaciones entre los elementos como si fueran **relaciones de parentesco**



- padres (*parents*)
- hijos (*child*)
- hermanos (*siblings*)
- ascendentes (*ancestors*)
- descendientes (*descendants*)

Múltiples estilos: herencia

Existen propiedades que automáticamente son heredadas en los elementos hijos. Tal es el caso de las propiedades que afectan al texto (**font-family, color...**): si se definen en <body> se aplicaran a todos los elementos contenidos en él.

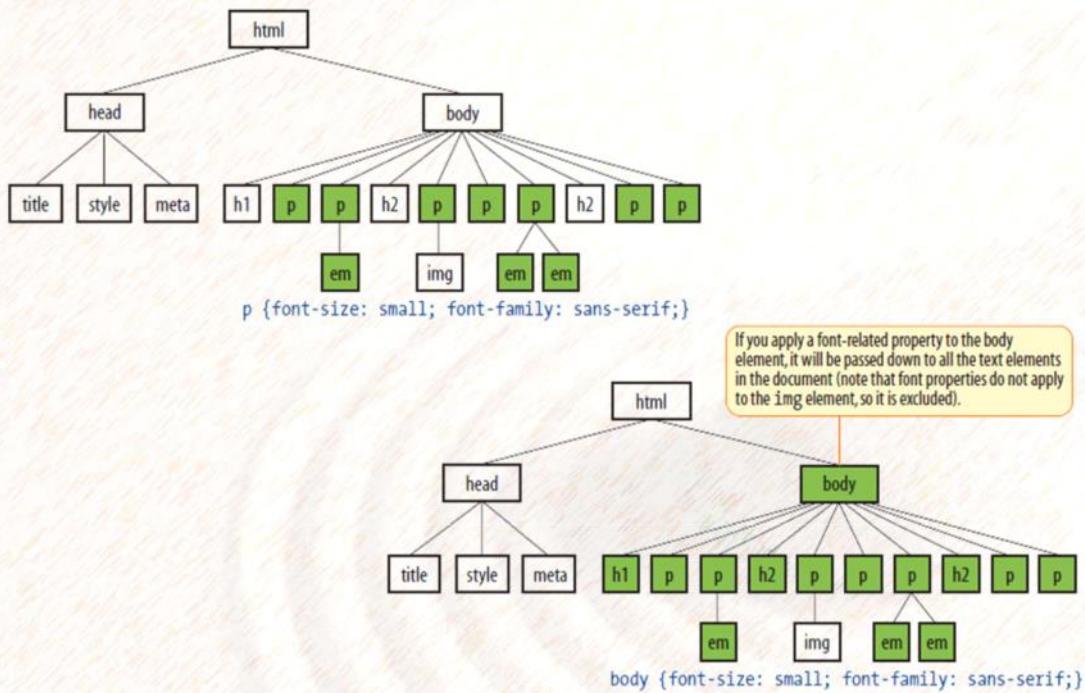
Existen otras propiedades que no son heredadas automáticamente. Tal es el caso de las que afectan a los contenedores del texto, i.e. las "cajas" (**background-color, margin, padding, border...**)

Para la mayoría de estos casos, se puede forzar la herencia de los valores del elemento padre indicando como valor el término *inherit*

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;  
    padding: 10px;}  
.page {  
    border: 1px solid #665544;  
    background-color: #efefef;  
    padding: inherit;}
```

La clase page hereda automáticamente **font-family** y **color** y además forzamos que herede **padding**

Ejemplo de herencia



Referencia: Selectores

Selector

Declaración

Declaración

h1

{color:blue; font-size:12px}

Hace referencia a la parte del documento a la que se aplicara el correspondiente estilo

- Referencia por etiqueta:
 - el selector es el nombre de una etiquete HTLM:
h1, p, span, div...
 - el estilo se aplica siempre que aparezca dicha etiqueta
- Referencia individual (por el atributo id)
- Referencia por clase (por el atributo class)

Referencias individuales

Selector

#estilo1

{color:blue; font-size:12px}

- Se define con el selector `#nombre_estilo`, para especificar un estilo exclusivo para un único elemento
- El estilo se aplica la única vez que aparece el atributo id correspondiente:
`<id="nombre_estilo">`

Las etiquetas `<div>` y ``, que define un grupo de elementos en bloque o en línea sin modificarlos en nada, son muchas veces el paso previo para aplicarles un estilo:

```
<span id="nombre_estilo">...</span>
```

Referencias de clase

`.estilo1`

`{color:blue; font-size:12px}`

Se define con el selector `.nombre_estilo`

El estilo se aplica cada vez que aparece el atributo **class**:

```
<class="nombre_estilo">
```

Nuevamente es muy habitual su uso con las etiquetas `<div>` y ``,

```
.termino_info {color:blue; font-size:12px}
```

```
<span class="termino_info">OOP</span>
```

El selector puede ser una clase asociada a una etiqueta concreta

```
p.párrafo_resumen {color:blue; font-size:12px}
```

Ejemplos del uso de id y class

Estilos embebidos

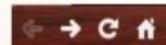
```
<head>
<style>
#titulo{
    color: red;
    font-size: 28px;
}
.autores{
    color: blue;
    font-size: 14px;
}
</style>
</head>
```

Fichero de estilo

```
#titulo{
    color: red;
    font-size: 28px;
}
.autores{
    color: blue;
    font-size: 14px;
}
```

Cuerpo del Fichero HTML

```
<body>
<h1 id="titulo"> Titulo </h1>
<p class="autores">Primer Autor</p>
<p>Biografia, Obras ...</p>
<p class="autores">Segundo Autor</p>
<p>Biografia, Obras ...</p>
</body>
```



Título

Primer Autor
Biografia, Obras ...
Segundo Autor
Biografia, Obras ...

Selectores agrupados

lunes, 19 de febrero de 2018 16:55

Selectores agrupados

Selectores

h1, h2, p

Declaraciones

**{color:blue;
font-size:12px}**

Para aplicar una o varias declaraciones a un conjunto de selectores, es suficiente indicarlos uno tras otro, separados mediante comas

Con frecuencia, un mismo selector se utiliza agrupado o por separado

```
h1, h2, p, em, img {border: 1px solid blue; }  
h1 {font-size: 1.5em}  
h2 {font-size: 1.3em}
```

Múltiples estilos. Cascadas

Lectura

1. Estilos “por defecto” del agente de usuario del navegador
2. Estilos definidos por el usuario a nivel del navegador. E.g. en Ópera, Ver → Estilo → Administrar modos... → Pantalla → Mi hoja de estilo.
3. Estilos aplicados por un autor mediante CSS externa
4. Estilos embebidos en la página web
5. Estilos “en-línea”, aplicados a una etiqueta.

Precedencia

- En caso de selectores idénticos, el último leído tiene precedencia
- Los estilos más específicos tienen precedencia sobre los más generales
- la propiedad **!important**: se impone a las reglas anteriores, haciendo que el estilo al que se aplica tenga precedencia

Ejemplo de múltiples estilos

```
<body>
  <h1>Tipos de elfos</h1>
  <p id="intro">Existen <i>docenas</i> de diferentes variedades de
  <b>elfos</b>.</p>
  <p>Entre otros pueden distinguirse por su ubicación: de los bosques, de
  los ríos, de las montañas...</p>
</body>

* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```

Tipos de elfos

Existen *docenas* de diferentes variedades de **elfos**.

Entre otros pueden distinguirse por su ubicación: de los bosques, de los ríos, de las montañas...

Especificidad

lunes, 19 de febrero de 2018 19:36

a | b | c | d

- a** es igual a 1 si la declaración está definida como estilos en línea
- b** es igual al número de id's
- c** es igual al número de otros atributos y pseudo-clases
- d** es igual al número de elementos y pseudo-elementos

0 | 0 | 0 | 1
elemento {}
::pseudo-elemento {}

0 | 0 | 1 | 0
.clase {}
[atributo] {}
:pseudo-clase {}

0 | 1 | 0 | 0
#identificador {}

1 | 0 | 0 | 0
style="color: lightcoral"

0 | 0 | 0 | 0
>, +, *, ~, not()
@media

Se mejora el rendimiento utilizando estilos con la menor especificidad posible y cargando los estilos según el mismo criterio

0010 < 0012
0 | 0 | 1 | 0 > 0 | 0 | 0 | 13

<https://specificity.keegan.st/>

Specificity Calculator

`li:first-child h2 .title`

0

Inline styles

0

IDs

2

Classes, attributes
and pseudo-classes

2

Elements and
pseudo-elements

+ Duplicate

`#nav .selected > a:hover`

0

Inline styles

1

IDs

2

Classes, attributes
and pseudo-classes

1

Elements and
pseudo-elements

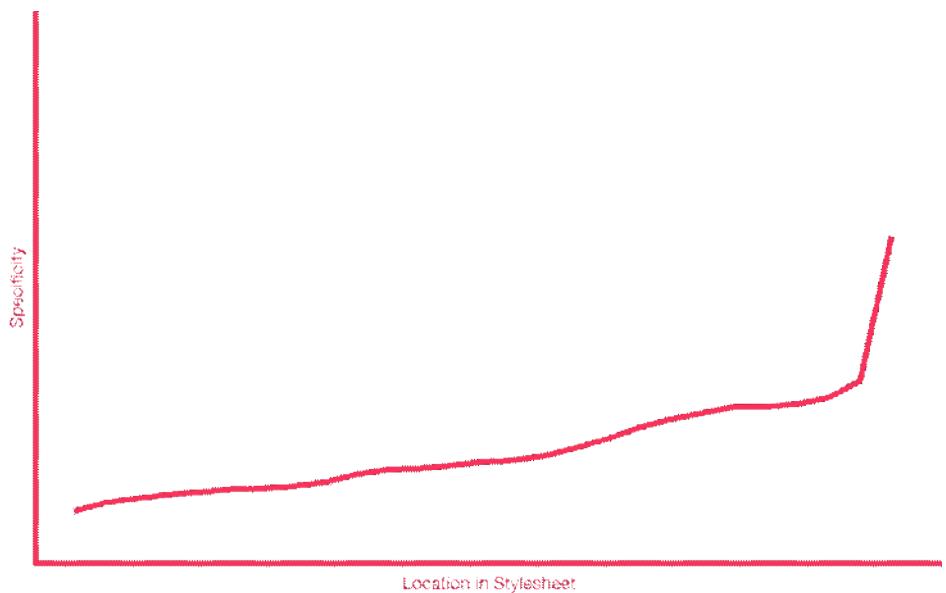
+ Duplicate

Gráficos de especificidad. Arquitecturas

lunes, 19 de febrero de 2018 19:46



Distribución de los niveles de especificidad en distintos archivos



Reordenación de los estilos en diversos archivos
de acuerdo con una arquitectura CSS

Elementos de diseño avanzados

Lunes, 19 de febrero de 2018 11:37

1. Selectores, pseudoclases y pseudoelementos
 - a. Selectores contextuales
 - b. Selector de atributo
2. Nuevos selectores en CSS3
 - Pseudoclases relativas a la posición en el DOM
 - Selector universal
 - Pseudoclase :not()
 - Selectores contextuales: Selector ~
3. Variables en CSS3. Funciones: calc()
4. Modelo de caja. Concepto.
 - a. Modelos disponibles: content-box / border-box
5. Inicialización. Reseteo y normalización.
6. Soporte y prefijos

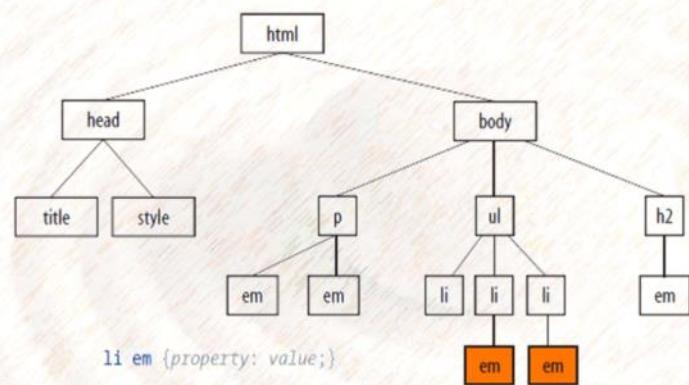


Selector de "descendientes"

Selector “espacio” (Descendant Selector) referencia elementos que descienden de otro, sin que necesariamente tengan que ser hijos

Por ejemplo, `div p {}` referenciará cada elemento `<p>` que está en un elemento `<div>`. Sólo se ve afectado el segundo selector, mientras que el primero le da el contexto

Anteriormente denominados **“selectores contextuales”**, HTML5 se refiere a este tipo como **“combinadores”**



Selectores contextuales / combinadores

- **Selector “espacio” (*Descendant Selector*)**
- **Selector > (*Child Selector*)** Este selector referencia al elemento de la derecha cuando tiene el elemento de la izquierda como parente.
Por ejemplo, `div > p {}` referenciará cada elemento `<p>` que es hijo de un elemento `<div>`.
- **Selector + (*Adjacent Sibling Selector*)** Este selector referencia elementos que son hermanos. La referencia apuntará al elemento de la derecha cuando es inmediatamente precedido por el de la izquierda.
Por ejemplo, `span + p {}` afectará a los elementos `<p>` que son hermanos y están ubicados luego de un elemento ``.
- **Selector ~ (*General Sibling Selector*)** Este selector es similar al anterior, pero en este caso el elemento de la derecha no tiene que estar ubicado inmediatamente después del de la izquierda.



Selector de Atributo



palabraclave[atributo=valor]

podemos utilizar otros atributos además de id y class para asignar estilos a elementos que tiene un atributo particular, incluso con un valor específico.

```
p[name="texto"] {color:blue; font-size:12px}
```

referenciará cada elemento <p> con un atributo llamado name y el valor "texto".

Esta referencia puede ser aún más específica incorporando símbolos

- ^= elementos que comienzan con el valor provisto
- \$= elementos que terminan con ese valor
- *= elementos que contienen el texto provisto

```
p[name^="texto"] {color:blue; font-size:12px}
```

se aplica a elementos <p> con un atributo name cuyo valor comienza por "texto".

[atributo = valor] elementos con ese valor

[atributo ^= valor] elementos que comienzan con el valor provisto

[atributo \$= valor] elementos que terminan con ese valor

[atributo *= valor] elementos que contienen el texto provisto

Selectores contextuales / combinadores

Selector "espacio" (Descendant Selector)

Selector > (Child Selector)

Este selector referencia al elemento de la derecha cuando tiene el elemento de la izquierda como parente.
Por ejemplo, div > p {} referenciará cada elemento <p> que es hijo de un elemento <div>.

Selector + (Adjacent Sibling Selector)

Este selector referencia elementos que son hermanos.

La referencia apuntará al elemento de la derecha cuando es inmediatamente precedido por el de la izquierda.
Por ejemplo, span + p {}

afectará a los elementos <p> que son hermanos y están ubicados luego de un elemento .

Selector ~ (General Sibling Selector)

Este selector es similar al

anterior, pero en este caso el elemento de la derecha no tiene que estar ubicado inmediatamente después del de la izquierda.

Pseudoclasess

Son clases definidas dinámicamente, sin un atributo específico que las identifique

- en función de acciones del usuario, como la posición del cursor (:link, :visited, :active, :hover, :focus)
- en función de la posición de los elementos y sus relaciones con otros (i.e su posición en el DOM)

[selector]:pseudoclase

Este aspecto dinámico les da un valor condicional

- Si se dan determinadas circunstancias se aplica un estilo
- Si no se dan no se aplica

Hiperenlaces y Pseudoclases

Las **pseudo-classes** permiten responder a **eventos** ligados a una etiqueta con la aplicación de estilos particulares



El estilo varia en función de elemento: se introduce dinamismo sin necesidad de scripts

:link

Para el identificador **a** se han definido dos de estas **pseudo-clases**, que permiten modificar el comportamiento que presentan los hiperenlaces por defecto: su cambio de color una vez que han sido visitados

:visited

a:link {propiedad: valor}

Enlace no visitado

a:visited {propiedad: valor}

Enlace visitado

Otras Pseudoclases

:focus

Elemento seleccionado
(con el teclado)

:hover

Elemento sobre el que se sitúa el ratón
u otro dispositivo apuntador

:active

Enlace clicado.
Si es un hiperenlace <a>,
a continuación cambiara a visitado

Muy habituales con hiperenlaces

a:hover {propiedad: valor}

:lang(<l>)

El estilo se aplicara solo si el elemento
afectado esta definido con el idioma <l>

:target

Se activa cuando un enlace del tipo `...` lleva a un elemento concreto de una página, identificado por su id

Pseudoelementos primera letra y línea

Los **pseudo-elementos** permiten aplicar un estilo a parte de un elemento

::first-letter

Afectará a la primera letra de un elemento

::first-line

Afectará a la primera línea de un elemento

Los orígenes de Internet se remontan a la década de 1960, dentro de ARPA (hoy DARPA), como respuesta a la necesidad de esta organización de buscar mejores maneras de usar los computadores de ese entonces, pero enfrentados al problema de que los principales investigadores y laboratorios deseaban tener sus propios computadores, lo que no sólo era más costoso, sino que provocaba una duplicación de esfuerzos y recursos

```
p.intro:first-line {font-weight: bold;}  
p.intro:first-letter {font-size: 200%;}
```

Pseudelementos *before* y *after*

Son estilos que sirven para definir una parte adicional del elemento al que se aplican, al principio o al final de este.

[selector]::pseudelemento

::before

Permiten añadir contenido antes o después del elemento al que se aplican

::after

Para ello es necesario utilizar la propiedad **content** (incluso si su valor es "")

```
h1::before {  
    content: url(smiley.gif);  
}
```

Content

content

Permite añadir contenido desde la hoja de estilo

content: **normal** | **none** | counter | attr | <**string**> | open-quote | close-quote | no-open-quote | no-close-quote | url | initial | inherit;

counter → añade un contador

attr (<atributo>) → Recoge el valor del atributo indicado del selector

url (url) → Incorpora un recurso (img, video audio)

open-quote (no-open-quote)
close-quote (no-close-quote) → Abrimos cerramos comillas en el selector

Content: las comillas

content open-quote (no-open-quote)
 close-quote (no-close-quote)

Es frecuente su uso ligado a los pseudoelementos ::after y ::before

Existe además la posibilidad de definir el tipo de comillas

quotes	"	double quote	" ;
	'	single quote	' ;
	<	single, left angle quote	‹ ;
	>	single, right angle quote	› ;
	«	double, left angle quote	« ;
	»	double, right angle quote	» ;
	‘	left quote (single high-6)	‘ ;
	’	right quote (single high-9)	’ ;
	“	left quote (double high-6)	“ ;
	”	right quote (double high-9)	” ;
	„	double quote (double low-9)	„ ;

modificar el
aspecto de las
comillas, e.g. en
<q> o al añadirlas
con content

quotes

Novedades en CSS3

- Pseudoclases relativas a la posición en el DOM
 - pseudoclase :nth-child()
 - Pseudoclase :first-child
 - Pseudoclase :last-child
 - Pseudoclase :only-child
- Selector universal y pseudoclase :not()
- Selectores contextuales (con operadores)
 - Selector ~
(amplia los ya existentes
Selector > / Selector +)



Pseudoclases jerárquicas (1)



- **:nth-child(n)** – referencia a un hijo de índice (n), o los hijos pares o impares (odd/even)

```
p:nth-child(2) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo que sea segundo entre los párrafos hijos

N puede ser un número, una palabra clave (como odd/even) o una fórmula tipo $a n + b$ donde a representa un ciclo, n es un contador a partir de 0 y b es un valor de desplazamiento

```
p:nth-child(3n+0) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo cuyo índice como hijo sea múltiplo de 3

Pseudoclases jerárquicas (2)



- **:nth-of-type(n)** -- referencia a un hijo específico (n), del mismo tipo que el selector

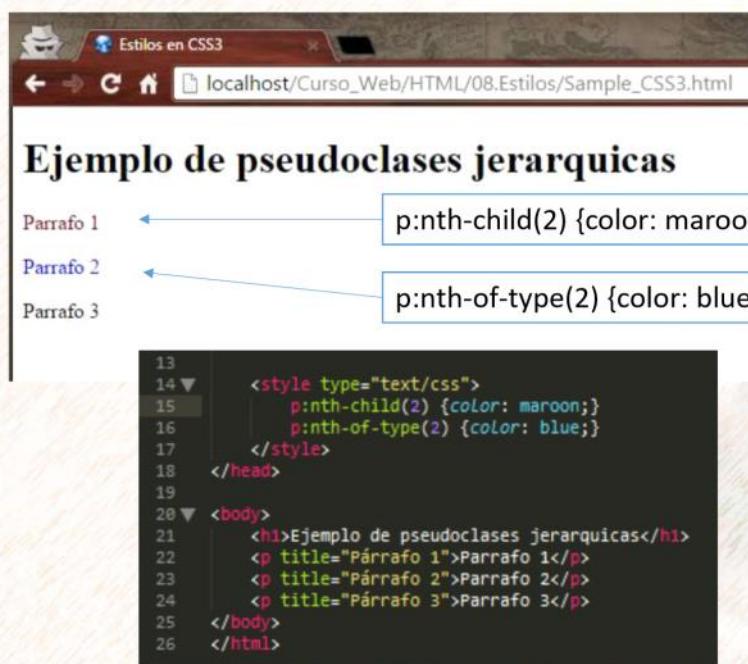
```
p:nth-child(2) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo que sea segundo hijo, sea el primero del tipo que sea

N puede ser un número, una palabra clave (como odd/even) o una fórmula

- **:first-child** – referencia al primer hijo, similar a :nth-child(1)
- **:last-child** – referencia al último hijo
- **:only-child** – referencia a un elemento que es el único hijo disponible de un mismo elemento padre

Ejemplo



The screenshot shows a web browser window titled "Estilos en CSS3" with the URL "localhost/Curso_Web/HTML/08.Estilos/Sample_CSS3.html". The page content is an

element followed by three elements: "Parrafo 1", "Parrafo 2", and "Parrafo 3". A callout box points from the text "Parrafo 1" to the CSS rule "p:nth-child(2) {color: maroon;}" in the code editor. Another callout box points from the text "Parrafo 2" to the CSS rule "p:nth-of-type(2) {color: blue;}" in the code editor. ``` 13 14 <style type="text/css"> 15 p:nth-child(2) {color: maroon;} 16 p:nth-of-type(2) {color: blue;} 17 </style> 18 </head> 19 20 <body> 21 <h1>Ejemplo de pseudoclases jerarquicas</h1> 22 <p title="Párrafo 1">Parrafo 1</p> 23 <p title="Párrafo 2">Parrafo 2</p> 24 <p title="Párrafo 3">Parrafo 3</p> 25 </body> 26 </html> ```

Selector universal y pseudoclase :not()



*

{color:black; font-size:16px}

el selector universal * asigna el mismo estilo a todos los elementos del documento.

:not (p)

{color:black; font-size:16px}

La pseudoclase :not() hace referencia a todos los elementos excepto el declarado entre paréntesis.

Ambos selectores representan el conjunto de todos los elementos en el cuerpo del documento o todos menos las excepciones indicadas, y son útiles cuando necesitamos establecer ciertas reglas básicas.

Variables y funciones en CSS

Lunes, 19 de febrero de 2018 21:07

Variables

Las variables permiten asignar cualquier valor a una propiedad cuyo nombre podemos elegir libremente.

```
:root {  
    --color-principal: #06c;  
}
```

se definen dentro de la etiqueta :root

Esos valores se reutilizan en cualquier otra propiedad gracias a la función var()

```
#foo h1 {  
    color: var(--color-principal);  
}
```

<http://librosweb.es/tutorial/como-usar-las-nuevas-variables-css/>

Pueden utilizarse expresiones complejas como

```
--foo: if(x > 5) this.width = 10;
```

Función calc()

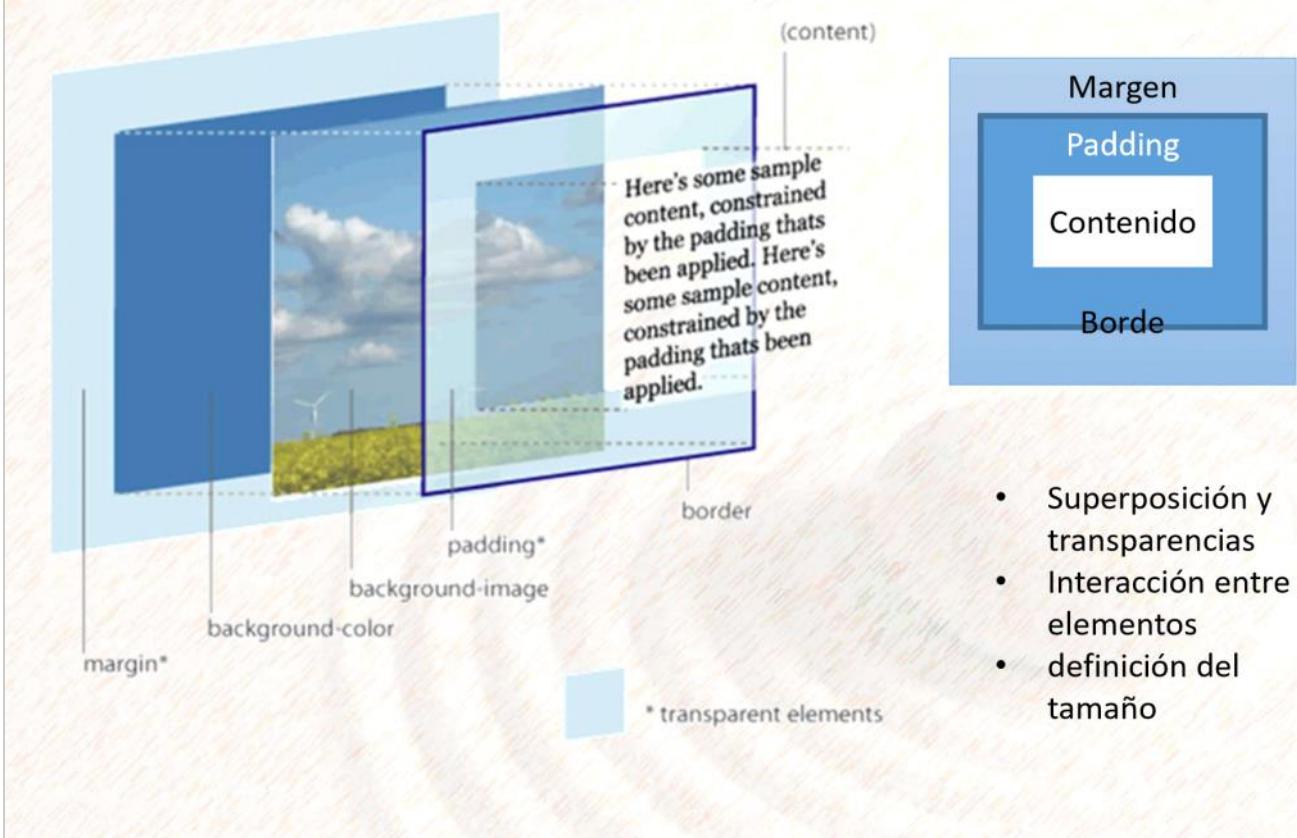
La función calc() es muy útil para realizar cálculos al definir los valores de las propiedades CSS. Todos los navegadores modernos la soportan sin problemas y se puede combinar con las variables CSS. Ejemplo:

```
.foo {  
    --separacion: 20;  
    margin-top: calc(var(--separacion) * 1px);  
}
```

Modelo de cajas

lunes, 19 de febrero de 2018 12:07

Modelo de cajas (Box Model)



Box-sizing: content-box

domingo, 14 de mayo de 2017 10:10

Por defecto cualquier elemento en HTML adapta su tamaño a su contenido.

En los elementos “en bloque” el ancho es el del contenedor y el alto se ajusta al número de líneas escritas

En los div, por defecto la altura es 0 si no contienen nada

También podemos definir el tamaño de forma explícita

box-sizing

content-box

Define como se contabilizan la altura y la anchura de los elementos de tipo bloque, de forma predeterminada. Era la única forma disponible antes de CSS3

height

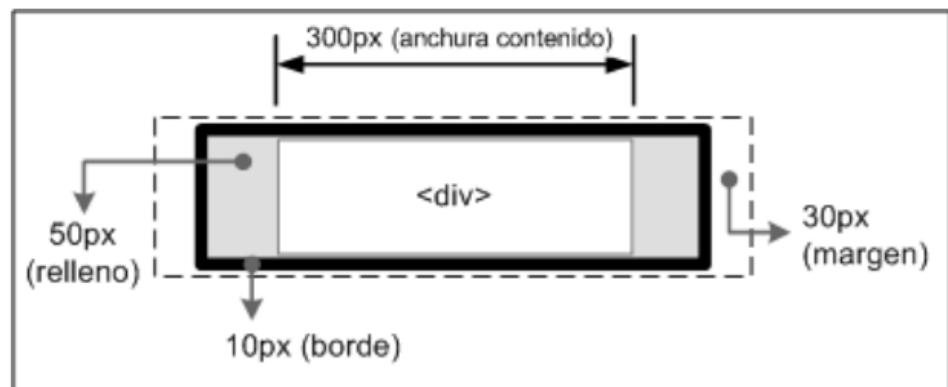
width

- Auto (valor por defecto)
- valor absoluto (pixeles, cm,...)
- valor relativo (%)

son solo las medidas en el contenido,

El tamaño que ocupa un elemento corresponde a la suma de margen + borde + relleno (padding) + tamaño indicado

En la figura vemos un div que realmente ocupa 480px



Box-sizing: border-box

lunes, 19 de febrero de 2018 18:46

box-sizing

border-box

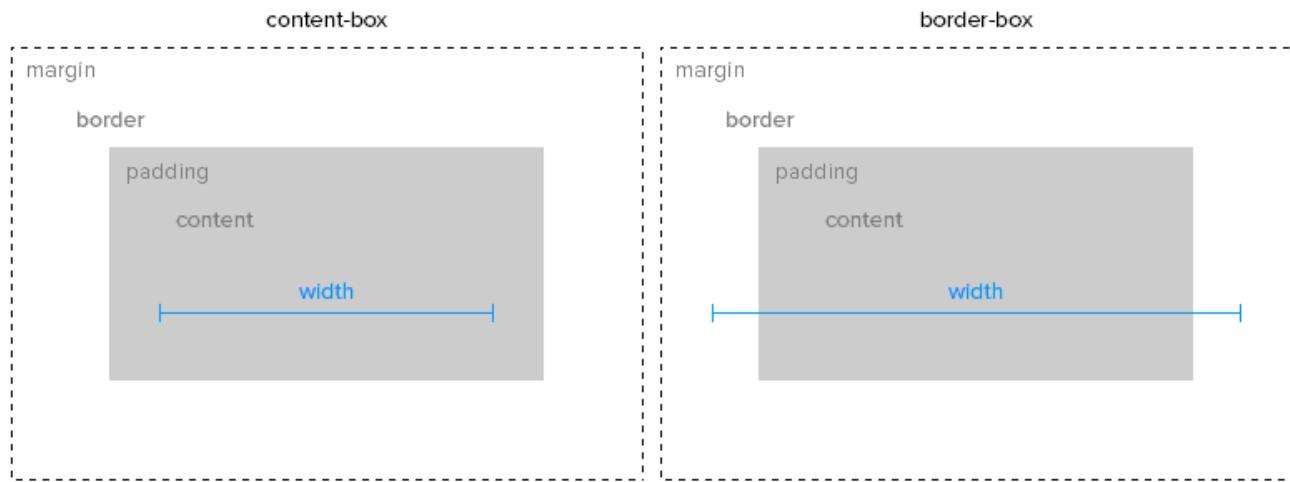
height

width



incluyen el borde y el relleno (padding), que por tanto se descuenten del tamaño indicado, en lugar de sumarse,

El tamaño final sería margen + tamaño indicado.



El nuevo modelo supone un gran cambio:

De leer el ancho y alto en función del contenido (modelo antiguo) pasamos a hacerlo en función de borde + padding + content

Cajas: límites de tamaño

min-height,
max-height

min-width,
max-width

Valores mínimos y máximos del ancho y alto de un elemento bloque

longitud (px, in, cm ...) porcentaje (%)

Definimos el comportamiento del contenido cuando las dimensiones de la caja no están en auto, y por tanto no se adaptan automáticamente a él

overflow

hidden
scroll

clip

rect(*top, right, bottom, left*)

RESULT

Fender Stratocaster

The Fender Stratocaster or "Strat" is one of the most popular electric guitars of all time, and its design has been copied by many guitar makers.

Gibson Les Paul

The Gibson Les Paul is a solid body electric guitar that was first sold in 1952. The Les Paul was designed by

para un elemento de posición definida de forma absoluta, fija (**clips**) el espacio que ocupa, sin que se pueda salir de el

Reseteo y normalización

Junes, 19 de febrero de 2018 13:45

Suele incluirse un CSS base, que consiste en hojas de normalización o de *reset*,

Una de sus funciones principales es aplicar de forma genérica, el Box-sizing: border-box
Con frecuencia en asociado al selector genérico (*) y añadiendo las pseudo clases ::after y ::before.

```
*,  
*::before, *::after,  
*:before, *:after {  
  box-sizing : border-box;  
}
```

Otros elementos reseteados con frecuencia

```
html, body, div, span, applet, object, iframe,  
h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
a, abbr, acronym, address, big, cite, code,  
del, dfn, em, img, ins, kbd, q, s, samp,  
small, strike, strong, sub, sup, tt, var,  
b, u, i, center,  
dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed,  
figure, figcaption, footer, header, hgroup,  
menu, nav, output, ruby, section, summary,  
time, mark, audio, video {  
  margin: 0;  
  padding: 0;  
  border: 0;  
  font-size: 100%;  
  font: inherit;  
  vertical-align: baseline;  
}
```

Prácticamente todas las etiquetas quedan "a cero"

```
body {  
  line-height: 1;  
}  
ol, ul {  
  list-style: none;  
}  
table {  
  border-collapse: collapse;  
  border-spacing: 0;  
}
```

Tamaño de línea 1

Listas sin esilos

Tablas sin bordes

```
blockquote, q {  
  quotes: none;  
}  
blockquote:before, blockquote:after,  
q:before, q:after {  
  content: '';  
  content: none;  
}
```

Citas sin comillas

article, aside, details, figcaption, figure,

Solo para navegadores antiguos,

```
article, aside, details, figcaption, figure,  
footer, header, hgroup, menu, nav, section {  
    display: block;  
}
```

Solo para navegadores antiguos,
se establece el carácter de
bloque para los componentes de
este tipo definidos en HTML5

Ejemplo más completo: <http://html5doctor.com/html-5-resetstylesheet/>

Una alternativa más potente es el uso
de herramientas de normalización

<https://necolas.github.io/normalize.css/>



Normalize.css

**A modern, HTML5-ready alternative
to CSS resets**

Soporte en los navegadores

Existen estilos que están aún en proceso de incorporación al estándar pero ya son soportados por los navegadores.

En ese caso es necesario indicarlo en los estilos, con los prefijos de los navegadores afectados

-webkit → Safari, Chrome y Opera 15 o >
-moz → Firefox
-o → Opera < 15
-ms → Internet Explorer

Para asegurarnos un valor flex = 1 en todos los navegadores que lo soporten

Chorme 29.0 (21.0 -webkit-)
IE 11.0 (10.0 -ms-)
Firefox 28.0 (18.0 -moz-)
Safari 6.1 -webkit-
Opera 12.10

```
#main div {  
    -webkit-flex: 1;  
    -moz-flex: 1;  
    -ms-flex: 1;  
    flex: 1;  
}
```

Texto, espacio y color en CSS

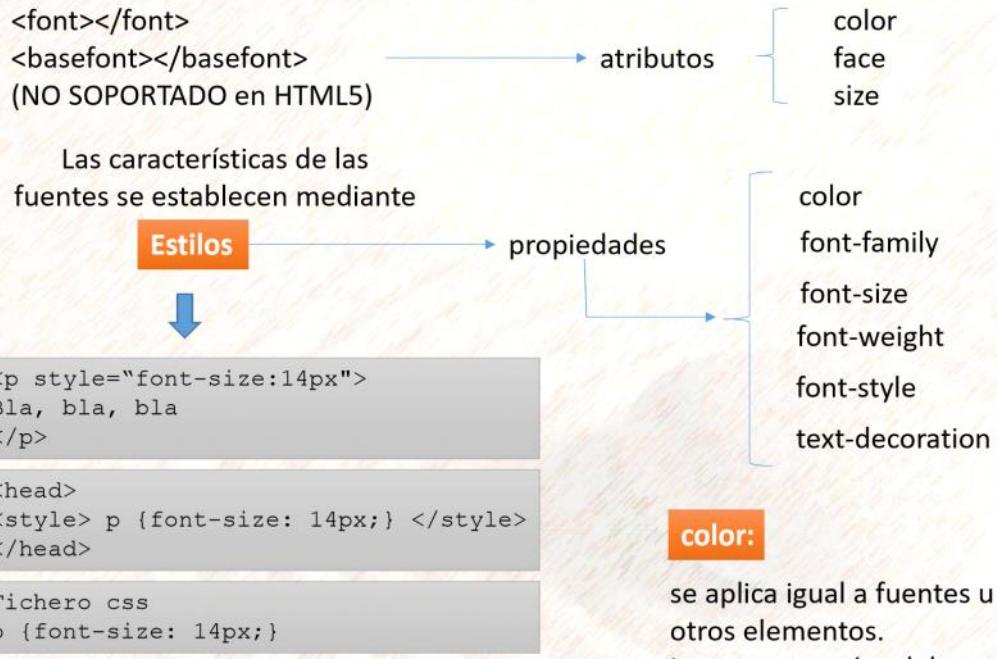
1. CSS aplicado a los textos
 - a. Formato de fuentes. Unidades de medida
 - b. Elementos bloque. Párrafos
2. Uso del espacio en CSS
 - a. Espacios (Margen y padding)
 - b. Bordes. Bordes redondeados
 - c. Posicionamiento estático. Float y clear
 - d. Elementos de CSS3. Sombras. Sombras de textos
 - e. Aplicación de estilos CSS en tablas
3. Uso básico del display
 - a. Etiqueta display
 - b. Block / Inline
 - c. Inline-block. Particularidades
 - d. Visibilidad
4. Colores
 - a. Codificación. Etiquetas CSS: Color / Background-color
 - b. Colores en CSS3. Transparencia. HSL. Gradientes
5. Herramientas CSS
6. Imágenes
 - a. CSS aplicable.
 - b. Imágenes como fondos



1. Formato de texto. Fuentes. Unidades de medida
 - a. *Font-family*. Fuentes externas
 - b. Tamaño de fuente. . Unidades de medida. Escalas tipográficas
 - c. Modificadores:
 - peso (negrita), estilo (cursiva) decoración (subrayado)
 - d. Tipografía avanzada. Tamaño de línea. Espaciado
2. Elementos bloque. Form de párrafos y listas
 - a. Alineación horizontal. Sangría
 - b. Alineación vertical
 - c. Formatos de listas



Formatos de texto: Fuentes



Formatos de texto: Estilos aplicados a las Fuentes (1)

Propiedades

font-family:

→ “fuente 1”, “fuente 2”, ..., fuente genérica

```
MiEstilo {font-family: Arial, Helvetica, sans-serif}
```

serif, e.g. Georgia, Times, Times New Roman ...
sans-serif, e.g. Verdana, Arial, Helvética ...
cursive, e.g. Comic Sans MS ...
fantasy, e.g. Impact, Haettenschweiler ...
monospace, e.g. Courier, Courier New ...

Letra Serif
Letra Sans Serif
Letra Cursive
Letra Fantasy
Letra Monospace

Se entrecosillan los nombres de fuentes compuestos de varias palabras



Incorporación de fuentes

@font-face:

selector especial que permite hacer referencia a una fuente para que se instale en el equipo cliente

Se utiliza junto con las propiedades

font-family: → nombre de la fuente
src: → url("url de la fuente")
format("formato")

Una vez definido el proceso que instalará la fuente, la utilizamos

```
@font-face {font-family: 'ChunkFiveRegular';
            src: url('fonts/chunkfive.svg') format('svg'); }
p { font-family: ChunkFiveRegular, Georgia, serif; }
```

Solo se deben proporcionar fuentes de licencia libre

→ www.fontsquirrel.com
www.fontex.org
www.openfontlibrary.org

Incorporación de fuentes: formatos

- 1: eot** `format ('embedded-opentype')`
- 2: woff** `format ('woff')`
- 3: ttf/otf** `format ('truetype')`
- 4: svg** `format ('svg')`

browser	eot	woff	ttf / otf	svg
Chrome		6+	6+	all
Firefox		3.6+	3.5+	
IE	5+	9+		
Opera			10+	10+
Safari			3.1+	3.1+
iOS			4.2+	all

Incorporación de fuentes: tools

www.fontsquirrel.com

→ generador de formatos y del código CSS

Tras subir una fuente como punto de partida, obtenemos un zip con todos los formatos y el CSS necesario para utilizarlos

WEBFONT GENERATOR

Usage: Click the "Add Fonts" button, check the agreement and download you choose the Expert option.



UPLOAD FONTS ↑

You currently have no fonts uploaded.

BASIC
Straight conversion with minimal processing

OPTIMAL
Recommended settings for performance and speed.

EXPERT...
You decide how best to optimize your fonts.

Agreement:

Yes, the fonts I'm uploading are legally eligible for web embedding.
Font Squirrel offers this service in good faith. Please honor the EULAs of your fonts.

```
@font-face {font-family: 'ChunkFiveRegular';
src: url('fonts/chunkfive.eot');
src: url('fonts/chunkfive.eot?#iefix') format('embedded-opentype'),
url('fonts/chunkfive.woff') format('woff'),
url('fonts/chunkfive.ttf') format('truetype'),
url('fonts/chunkfive.svg#ChunkFiveRegular') format('svg');}
```

Incorporación de fuentes: Google-Fonts

Google Fonts

<http://www.google.com/fonts>

All categories

- Serif
- Sans Serif
- Display
- Handwriting
- Monospace

OK Reset

Thickness

thin thick

Slant

straight slanted

Width

narrow wide

Latin

Cyrillic
Cyrillic Extended
Devanagari
Greek
Greek Extended
Khmer
Latin
Latin Extended
Vietnamese

Varios filtros permiten
ajustar la búsqueda
entre las cientos de
fuentes disponibles

Standard @import Javascript

3. Add this code to your website:

```
<link href='http://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet' type='text.
```

Elegimos la forma de incorporar
la fuente, como link o como
@import (similar a @Font-face)

Formatos de texto: Estilos aplicados a las Fuentes (2)

Propiedades

font-size:

absoluto
relativo

- px (pixels)
- in (*inches*, pulgadas)
- cm, mm
- pt (puntos), pc (picas)

1in = 2.54cm = 25.4mm = 72pt = 6pc

1px = 1/96 of an inch in CSS3

- % del tamaño de fuente definido en el padre del elemento (si no se indica nada, el valor por defecto corresponde a 16 px)
- **em** (el tamaño base corresponde a la anchura de la M en el elemento padre); ex (lo mismo respecto a la altura de la x). Dependen de la fuente
- **rem** ("root em") ; un valor em constante en todo el documento
- *larger / smaller* frente al tamaño del elemento padre
- *xx-small, x-small, small, medium, large, x-large, o xx-large*, una escala de 1 a 7 centrada en el valor por defecto, que corresponde a *medium*

Tamaños de fuentes: em / rem

En tipografía tradicional → es el ancho de la M mayúscula de la tipografía elegida

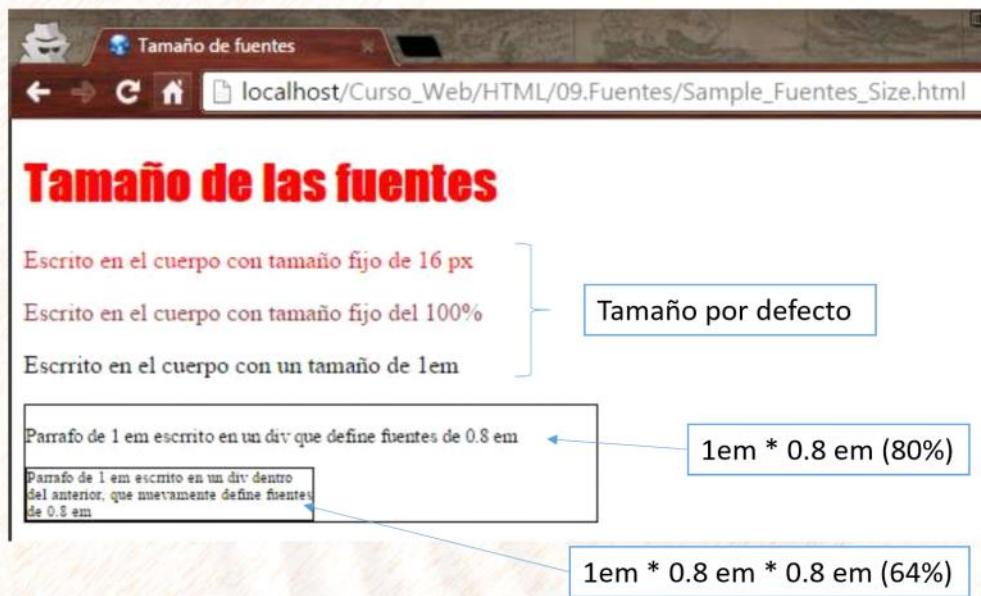
En la especificación CSS → distancia entre las líneas base cuando se aplica la fuente sin ningún espacio entre líneas

El tamaño de fuente en **ems** corresponde a un **factor de escala** a partir del tamaño de la letra en el elemento padre o contenedor

En el **body**, 1em o el 100% corresponde al tamaño de letra predefinido en el navegador, por defecto de 16px

El tamaño de fuente en **rems** (root ems) corresponde a un factor de escala a partir del tamaño de la letra en el elemento raíz, **body**

Ejemplo: tamaños de fuente



Escalas de tamaño en los tipos de letra



Escala “del 12” desarrollada por los tipógrafos europeos desde el s. XVI

- El cuerpo del texto corresponde a 12 px
- Los encabezados principales a 24 px

En los navegadores el tamaño por defecto del texto es de 16 px

En consecuencia, para usar la escala del 12, el valor por defecto para el cuerpo sería 75% (75% de 16 = 12) y para los encabezados principales sería 150 % (150 % de 16 = 24)

En diseños recientes se utiliza una escala “del 16”, para ajustarse al valor de partida de los navegadores

Escala del 12 v. escala del 16

PIXELS	PERCENTAGES	EMS
TWELVE PIXEL SCALE		
h1 24px	=	h1 1.5em
h2 18px	=	h2 1.3em
h3 14px	=	h3 1.17em
body 12px	=	body 100%
	body 75%	p 0.75em
SIXTEEN PIXEL SCALE		
h1 32px	=	h1 2em
h2 24px	=	h2 1.5em
h3 18px	=	h3 1.125em
body 16px	=	body 100%
	body 100%	p 1em

- El valor real depende de la resolución de la pantalla 800x600 1280x800....
- El usuario puede reducir o aumentar el valor por defecto del navegador

Formatos de texto: Estilos aplicados a las Fuentes (3)

Propiedades

font-weight

normal
bold
bolder / **lighter**
un número de 100 a 900

Generalmente **bolder** / **lighter** equivalen a **bold** / **normal**
Los valores **700-900** y **100-600** equivalen a **bold** / **normal**

font-style

normal (not italic)
italic
oblique ("false italics")

Italic intenta usar un formato de fuente específico de tipo italic.
Si no existe, italic = oblique = se inclinan las letras del formato normal

Etiqueta Font

Es el “atajo” (*shorthand*) que permite aplicar en una sola propiedad las distintas propiedades relativas a **font**



```
{font: italic bold 12px/30px Georgia, serif;}
```

This is a paragraph. This is a paragraph.

Formatos de texto: Estilos aplicados a las Fuentes (4)

Propiedades

text-decoration

- underline
- line-through (tachado)
- overline (línea sobre el texto)
- blink (texto parpadeante)
- none

none elimina cualquier decoración heredada
blink esta desaconsejado, e incluso es ignorado en algunos navegadores

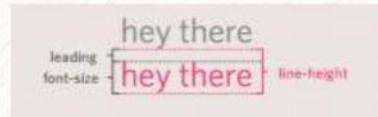
text-transform

- uppercase
- lowercase
- capitalize

underline
~~line through~~
overline
blink
none

UPPERCASE
lowercase
Capitalize

Tipografía avanzada (1): Tamaño de línea



line-height = font-size + leading

Leading es el espacio entre líneas.

El valor por defecto depende de la fuente; suele estar en torno a 0,25em (4px)

Propiedades

line-height

A partir del valor por defecto (0):

- Número de veces el valor inicial
- Número de pixeles
- % del valor inicial

Tipografía avanzada (1): Espacio entre letras y palabras

Kerning es el espacio entre las letras.

El valor por defecto depende de la fuente, pero suele estar en torno a 0,25em
Junto con el espacio entre palabras, se controla en los estilos, añadiendo una
nueva propiedad

Propiedades

letter-spacing

A partir del valor por defecto (0):

- Números + : aumento el espacio
- Números -: reducción del espacio

word-spacing

Valores en px, em, ...

Formatos de párrafo: alineamiento (1)

<p> párrafo </p>

Atributo:

align: alineación de la línea. Puede tomar tres valores
(NO SOPORTADO en HTML5)

- *left*
- *center*
- *right*

Formatos de párrafo

Alineamiento → distribución del texto dentro del párrafo

Propiedades

text-align

- left (valor por defecto)
- center
- right
- justify

This paragraph is justified. The edges align neatly at both the left and right, and extra space is inserted between words to make that happen. This paragraph is justified. The edges align neatly at both the left and right, and extra space is inserted between words to make that happen.

En **tipografía**, el indentado, sangrado o sangría es el desplazamiento a la derecha que se aplica a la primera línea o a todo el párrafo excepto la primera línea (sangría francesa o colgante).

text-indent

longitud (px, cm, em...)
porcentaje

Los valores pueden ser positivos o negativos

Alineamiento vertical

Lunes, 19 de febrero de 2018 20:26

vertical-align

baseline
sub
super
top
text-top
middle
bottom
text-bottom

Alinea verticalmente un elemento en línea respecto al bloque que lo contiene

NO alinea verticalmente el contenido de un elemento bloque, como <p> o <div>

Se utiliza principalmente en 2 situaciones

- Alineamiento entre sí de varios elementos inline-block
- Alineamiento entre table-cells cuando se utiliza display: table

Listas

.... Listas no ordenadas (Bulleted)

disc	• Circulo relleno (default)
circle	○ Circulo no relleno
square	■ Cuadrado relleno

.... Listas ordenadas (Numbered)

decimal	1, 2, 3, 4 (default)
decimal-leading-zero	01, 02, 03, 04
lower-roman	i, ii, iii, iv
upper-roman	I, II, III, IV
lower-alpha	a, b, c, d
upper-alpha	A, B, C, D
none	(nada)

list-style-type

Los *bullets* no son modificables mediante estilos: su formato es responsabilidad del navegador

Listas

list-style-position

outside (default)

marcadores fuera
del contenedor

• Coffee
• Tea
• Coca-cola

Inside

marcadores dentro
del contenedor

• Coffee
• Tea
• Coca-cola

list-style-image

Una imagen como marcador

url('sqpurple.gif')

Podemos acceder de forma abreviada a los tres aspectos

list-style

<list-style-type> <list-style-position> <list-style-image>

En resumen. Estilos mencionados

Estilos que afectan a la fuente, en elementos en línea o en bloque	Estilos que afectan al párrafo globalmente	Estilos que afectan a cualquier elemento en bloques (v.i.)
font-family	line-height	padding
font-size	letter-spacing	margin
font-weight	word-spacing	border
font-style	text-align	border-[]-width
text-decoration	vertical-align	border-[]-style
text-transform	text-indent	border-[]-color
color	list-style-type	

Uso del espacio

Lunes, 19 de febrero de 2018 11:37

- a. Formato de bloques en CSS.
 - 1. Espacios : Margen y padding
 - 2. Valores auto y centrado de una caja
- b. Bordes.
- c. Elementos de CSS3
 - 1. Bordes redondeados
 - 2. Bordes mediante imágenes.
 - 3. Sombras. Sombras de textos
- d. Posicionamiento estático.
 - 1. Float
 - 2. Clear
 - 3. Float y colapso: ClearFix
- e. Aplicación de estilos CSS en tablas



Espacio: márgenes y padding

domingo, 14 de mayo de 2017 10:52

Margen (*margin*), bordes (*border*) y aire (*padding*) en las cajas de texto

Espacios alrededor

padding

**padding-top,
padding-right
padding-bottom,
padding-left**

Espacio interior a un elemento,
entre su borde y su contenido,

Equipment

Unfortunately, most people lack the proper equipment. Most hobbyists have either a handheld tank sprayer that requires pumping, or a sprayer that fits on the end of a hose. The pump sprayer applies chemicals accurately, and enables the proper mixing to be done, but the range is limited and upper branches may be missed. Hose-end sprayers can reach high up, but it is difficult to properly gauge the mix of water to chemical. The mix of water to chemical is an essential factor, so we do not recommend hose-end sprayers. Either use a tank sprayer (and stand on a ladder if needed), rent equipment, or hire a professional to do the spraying.

Padding: added space is inside

margin

**margin-top
margin-right
margin-bottom,
margin-left**

Espacio exterior, alrededor de
un elemento

Equipment

Unfortunately, most people lack the proper equipment. Most hobbyists have either a handheld tank sprayer that requires pumping, or a sprayer that fits on the end of a hose. The pump sprayer applies chemicals accurately, and enables the proper mixing to be done, but the range is limited and upper branches may be missed. Hose-end sprayers can reach high up, but it is difficult to properly gauge the mix of water to chemical. The mix of water to chemical is an essential factor, so we do not recommend hose-end sprayers. Either use a tank sprayer (and stand on a ladder if needed), rent equipment, or hire a professional to do the spraying.

Margin: added space is outside

Todos ellos

en px
en %

Un valor para los cuatro
lados, los cuatro valores,
o asignamos valores
individualmente

Los márgenes y los paddings en % se calculan respecto al ANCHO del contenedor (padre)

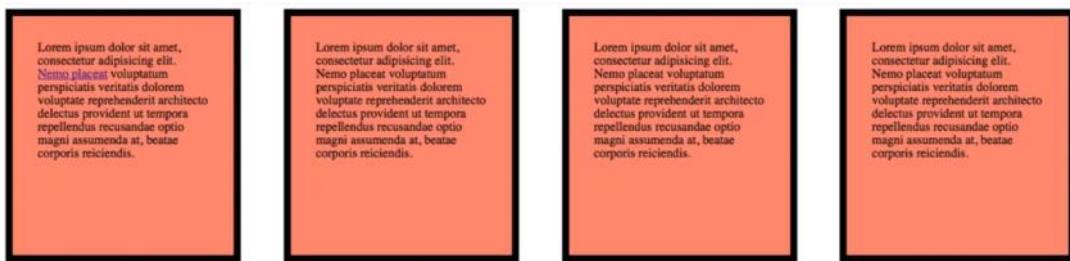
Márgenes y centrado de bloques

domingo, 14 de mayo de 2017 10:55

Aplicación del margen:

```
div {  
    width: 22%;  
    // sobra 12% -> 4% * 3  
    margin-right: 4%;  
  
    div:last-child {  
        margin-right: 0%;  
    }  
}
```

```
Opción mejor  
div {  
    width: 22%;  
}  
div:not(:last-child){  
    margin-right: 4%;  
}
```



Centrando una caja

Sobre una caja de tamaño definido (width:...) las propiedades **margin-left** y **margin right** con valor **auto**, tienen como consecuencia que la caja quede centrada en el elemento que la contiene

```
h1 {  
    text-align: center;  
}  
p {  
    width: 300px;  
    margin: 10px auto 10px auto;  
    padding: 50px;  
    border: 20px solid #0088dd;  
    text-align: left;  
}
```

Un párrafo del Neuromante

El cielo sobre el puerto tenía el color de una pantalla de televisor sintonizado en un canal muerto.

Estilos aplicados a los bordes (1)

border-style

- solid
- dotted (punteado)
- dashed (rayado)
- double
- groove (entrante)
- ridge (saliente)
- inset (sombra arriba)
- outset (sombra abajo)
- none

Wurlitzer Electric Piano

border-top-style border-right-style border-bottom-style border-left-style

Si no hay que aplicar el mismo estilo a los cuatro bordes, se puede hacer referencia individual a cada uno de ellos

Estilos aplicados a los bordes (2)

Una vez definido un estilo de borde, podemos seleccionar su anchura y color

border-width



Podemos indicar un valor para los 4 bordes o una sucesión de cuatro valores, correspondientes a *top, right, bottom, left*

```
p.one {border-width: 2px;}  
p.two {border-width: thick;}  
p.three {border-width: 1px 4px 12px 4px;}
```

border-top-width
border-right-width
border-bottom-width
border-left-width

Igual que en las otras propiedades del borde, podemos hacer referencia a la anchura de cualquiera de los lados de forma individualizada

Estilos aplicados a los bordes (3)

border-color

El color de los bordes se define de igual manera que el resto de los colores: nombre, rgb, hexadecimal o hsl, indicando 1 valor o cuatro valores sucesivos

**border-top-color
border-right-color
border-bottom-color
border-left-color**

```
p {border-color: #bbbbbaa #111111  
#ee3e80 #0088dd;}
```

De nuevo podemos hacer referencia al color de cualquiera de los lados de forma individualizada

Podemos hacer referencia a las tres propiedades globalmente, siempre que sean iguales los 4 bordes, siguiendo el orden indicado

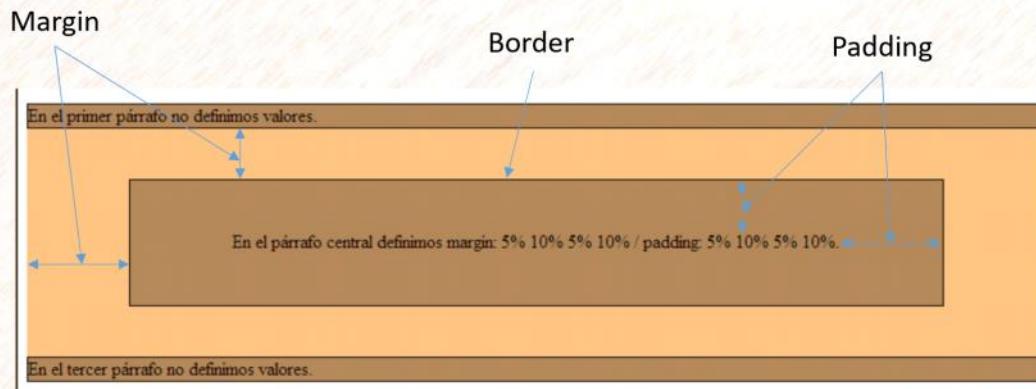
border

→ size style color.

```
p {border: 2px solid green; }
```

Espacios alrededor: Ejemplos

```
p.mg {margin: 5% 10% 5% 10%;  
padding: 5% 10% 5% 10%;}  
  
p { background-color: rgba(0,0,0,0.3);  
border-width: 1px;  
border-style: solid;
```



CSS3: bordes mediante imágenes



border-image

url de la imagen: url("url de la imagen")

→ puntos en los que cortarla

forma como contruir los lados (stretch, repeat, round)

Toma una imagen y la recorta en 9 fragmentos, 8 de los cuales (descartando el centro) utilizara para crear el borde del elemento afectado



dots.gif

```
border-image: url("images/dots.gif")  
11 11 11 11 stretch;}
```

```
border-image: url("images/dots.gif")  
11 11 11 11 round;}
```



CSS3: sombras



box-shadow

- desplazamiento horizontal (+ o -) en px
- desplazamiento vertical (+ o -) en px
- distancia de difuminación (*blur*)
- tamaño de la sombra (*spread*)
- (*inset*)
- color

Si se incluye la palabra clave *inset* se cambia el modo de la sombra de externo (*outset*) a interno

```
box-shadow:  
5px 5px 10px 5px  
rgb(150,150,150);
```



CSS3 sombras en los textos



text-shadow

desplazamiento horizontal (+ o -) en px
desplazamiento vertical (+ o -) en px
tamaño de la sombra (*spread*)
color

```
text-shadow: 5px 5px 5px  
rgb(150,150,150)
```

Sombras

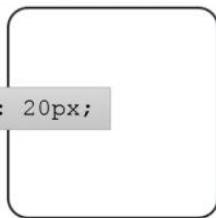
CSS3: Esquinas redondeadas



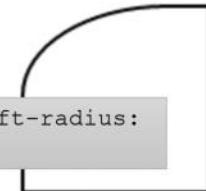
border-radius

→ valor en pixeles del radio(s) de la esquina(s)

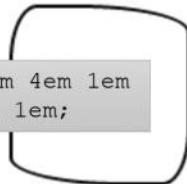
`border-radius: 20px;`



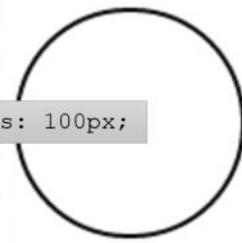
`border-top-left-radius: 80px 50px;;`



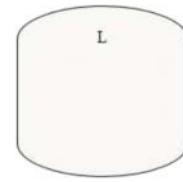
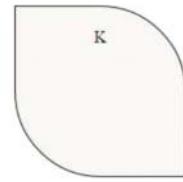
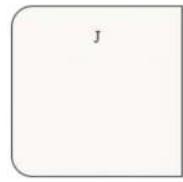
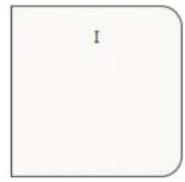
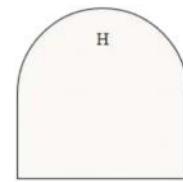
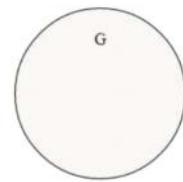
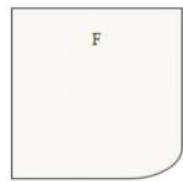
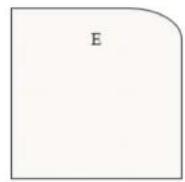
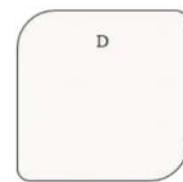
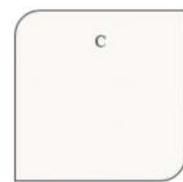
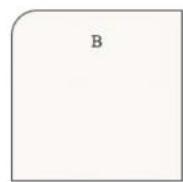
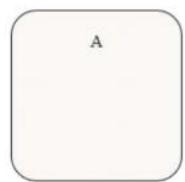
`border-radius: 1em 4em 1em
4em / 2em 1em 2em 1em;`



`border-radius: 100px;`



Ejemplos



[Curso_Web/HTML/11.Cajas_Textos/Sample_06.Esquinas_2.html](#)

Bordes y dibujo

El bisel de los bordes y la posibilidad de usar distintos colores o transparencias nos permite usarlos como elemento de dibujo

Div con bordes de distinto color



```
width: 50px;  
height: 50px;  
border-top: 30px solid green;  
border-right: 30px solid blue;  
border-bottom: 30px solid red;  
border-left: 30px solid purple;
```



El mismo div con su ancho por defecto (0): sólo se ven (“se colapsa”) los bordes de distinto color

Ejemplo: triángulos

Un borde de un color

Los 2 adyacentes transparentes (si son del mismo ancho, equilátero)



```
div.triangulo_left {  
    border-left: 100px solid rgb(240,173,78);  
    border-top: 50px solid transparent /*Ancho 1/2*/;  
    border-bottom: 50px solid transparent /*Ancho 1/2*/;  
}  
div.triangulo_top {  
    border-top: 100px solid rgb(240,173,78);  
    border-left: 50px solid transparent /*Ancho 1/2*/;  
    border-right: 50px solid transparent /*Ancho 1/2*/;  
}
```

Ejemplo: otras formas

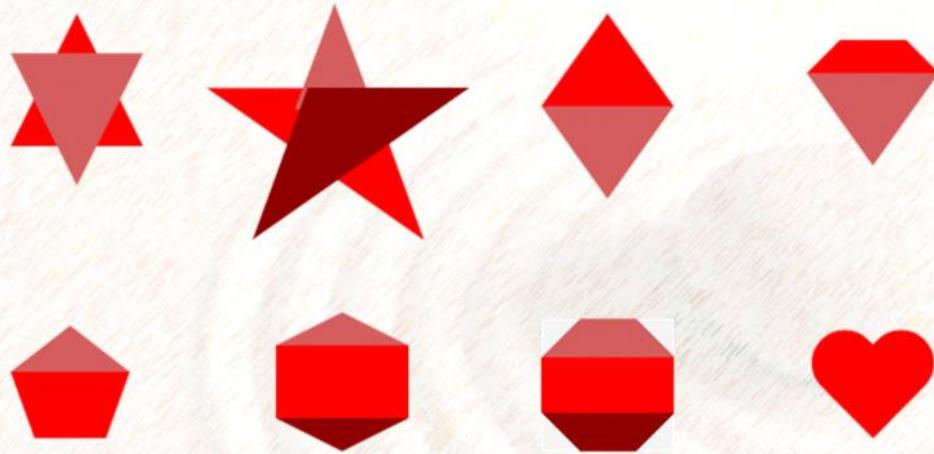
formas sencillas, basadas en triángulos



formas sencilla, basadas en border-radius

Ejemplo: otras formas

formas combinadas, basadas en
:after y :before



Formas en CSS: ayudas

Ejemplos y descripciones

<http://cybmeta.com/formas-basicas-con-css-triangulos-circulos-trapezios-rectangulos-cuadrados/>

The Shapes of CSS <https://css-tricks.com/examples/ShapesOfCSS/>

CSS Hexagon Tutorial <http://itauber.github.io/articles/css-hexagon.html>

Generadores

<http://www.nosolocss.com/blog/recursos/recopilacion-de-los-mejores-recursos-para-crear-formas-con-css>

CSS Shape Generator <http://www.samuelrossille.com/css-shape/>

Covelping Tools <https://covelping.com/tools/css-shapes-generator>

Ribbon builder <http://livetools.uiparade.com/ribbon-builder.html#>

Bubbler <http://www.ilikepixels.co.uk/drop/bubbler/>

Formas en CSS: ayudas

The Shapes of CSS

<https://css-tricks.com/examples/ShapesOfCSS/>

Covelping Tools

<https://covelping.com/tools/css-shapes-generator>

Ejemplos y descripciones

Generadores

<http://cybmeta.com/formas-basicas-con-css-triangulos-circulos-trapezios-rectangulos-cuadrados/>
<http://www.nosolocss.com/blog/recursos/recopilacion-de-los-mejores-recursos-para-crear-formas-con-css>

CSS Shape Generator

<http://www.samuelrossille.com/css-shape/>

Ribbon builder

<http://livetools.uiparade.com/ribbon-builder.html#>

CSS Hexagon Tutorial

<http://itauber.github.io/articles/css-hexagon.html>

Bubbler

<http://www.ilikepixels.co.uk/drop/bubbler/>

Posición de los elementos

Existen varios esquemas de posicionamiento

- Flujo normal
- Posicionamiento relativo
- Posicionamiento absoluto

Flujo normal

- cada elemento “bloque” inicia una nueva línea, colocándose inmediatamente debajo del elemento anterior
 - aunque indiquemos el ancho de un bloque el siguiente empezara debajo, incluso si tiene sitio para colocarse a continuación
 - es el esquema por defecto

Section 2

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

**Duis aute irure dolor in reprehenderit
in voluptate velit.**

Variaciones en el flujo

Para ajustar la distribución de los elementos a la disposición (*layout*) elegida, podemos emplear dos estrategias

- cambiar el esquema de posicionamiento con los valores de *position*, *relative*, *absolute* o *fixed*
- **alterar el flujo normal de los bloques**



Los elementos de tipo bloque tienden a ocupar la siguiente línea, saltando la ya ocupada por un elemento bloque anterior

La variación de flujo
se consigue alterando
este comportamiento

- **Con las propiedades *float* y *clear***
- Redefiniendo la propiedad *display* de los elementos

Float

float

→ left
right

Float indica que un elemento puede “flotar”, saltándose el flujo normal, situándose lo más lejos posible, a izquierda o derecha en su contenedor.

El resto de los elementos dentro del contenedor parece que fluyen en torno al elemento que está flotando

un elemento bloque no flotante se sitúa detrás del que flota y es sólo su contenido el que fluye respetando al elemento flotante. Este ejemplo es extraño efecto que se produce sobre una lista.

Para que los resultados sean consistentes, el elemento marcado como *float* suele tener un ancho establecido, e.g. mediante la propiedad *width* aplicada a un párrafo

FLOAT: POSICIONAMIENTO FLOTANTE

- Se desplazan todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.
- Si hay otras cajas flotantes, NO SE SUPERPONEN ENTRE SÍ, se colocan donde caben.
- FUERA DEL FLUJO NORMAL: pero de una manera especial y también hay “SOLAPAMIENTOS” pero de manera especial.

Otros elementos ocupan el lugar del elemento flotado:

- Elementos EN LÍNEA: Hacen sitio adaptando su ancho al espacio libre dejado por el elemento flotado
- Elementos EN BLOQUE: No les hacen sitio (ocupan su lugar) pero ADAPTAN su contenido para no solapar con el elemento flotado.

Float left: ejemplo

Los profesionales más buscados en el sector de la tecnología en 2014

Div add

Alberto Iglesias Fraga

```
h1, p.info  
{margin: 5%;}  
  
div.add  
{float:left;  
width:45%;  
border-style:solid;  
border-width: 2px  
0px;  
margin: 3%;}  
  
ul:nth-of-type(2)  
{list-style-  
position: inside;}
```

Periodista especializado en tecnología, o eso al menos quiero creer. Durante unos dos años fui redactor y community manager de ComputerWorld y CIO España, hablando de entres otros como servidores, redes y centros de datos. Desde entonces sigo hablando de las TIC –aunque algo menos friki– en TICbeat, iPhonazte y el blog Think Big de Telefónica.

TICbeat

Publicación digital independiente especializada en tecnología e innovación, en su industria, sus principales tendencias y todo lo que rodea a estos ámbitos. Desde su nacimiento en 2009, el site se ha convertido en un medio de comunicación de referencia que no solo aporta puntualmente información sobre los hechos más notables que acontecen en el sector TIC sino también una capa de análisis gracias al bagaje de su equipo, compuesto por periodistas y analistas con amplia experiencia en este nicho de mercado.

TICbeat nunca ha querido ser "uno de tantos" sites de tecnología que proliferan en español. Su enfoque, de hecho, es nítico y multidisciplinar. Tecnología de consumo y de empresas, Internet y social media, marketing y economía digital y colaborativa, emprendimiento y cibercultura son los principales temas que aborda la publicación a diario, siempre con un especial barniz dirigido a facilitar a su audiencia las armas para comprender a una sociedad, la actual, cada vez más cambiante y apasionante gracias al empleo de las tecnologías de la información.

Son muchos los análisis y estudios que se realizan prácticamente cada mes sobre los info más buscados por las empresas. En ese sentido, en nuestro país destaca el informe de Adecco, que a principios de este 2014 señaló varios empleos relacionados con las TI entre los más demandados por las organizaciones:

- consultor SEO-SEM.
- ingeniero para proyectos internacionales
- administradores de sistemas
- responsables de Business Intelligence y
- arquitectos Java

En un estudio presentado por WalletHub (realizado, eso sí, en Estados Unidos, aunque sus datos pueden extrapolarse fácilmente a cualquier país) hace apenas uno en el que se detallan cuáles son los profesionales más buscados por las comis especialmente en aquellos perfiles TI con menos experiencia o con perfil in

De acuerdo a estos datos, los profesionales más demandados en este 2014:

- desarrolladores de aplicaciones web
- analista de seguridad
- diseñadores web
- ingeniero de software
- ingeniero de redes
- programadores en general
- administrador de bases de datos
- administrador de sistemas

Pero el sector TIC no sólo copa las primeras posiciones en búsqueda de nuevos profesionales, sino que también agrupa las posiciones con mayor potencial de crecimiento para 2022, siempre según este mismo informe. Así pues, la demanda de analistas de seguridad crecerá un 36,5% en los próximos ocho años, mientras que los programadores harán lo propio en un 27,6%.

p info

ul(2)

Float right: ejemplo

Raymond Chandler

```
h1, div.bio  
{margin: 5%;}  
  
div.add  
{float:right;  
width:45%;  
border-style:solid;  
border-width: 2px  
0px;  
margin: 3%;}
```

Raymond Thornton Chandler (Chicago, 22 de julio de 1888 - La Jolla, California, 26 de marzo de 1959) fue un escritor estadounidense de novela negra.

Aunque estadounidense, tras el divorcio de sus padres fue llevado a Inglaterra, donde recibió una sólida formación literaria. Estudió en el Dulwich College de Londres (1900-05) a clásicos y modernos. También viajó a Francia y Alemania (1905-07) y se nacionalizó británico en 1907. Fue soldado en los Gordon Highlander de Canadá, empleado de banco, periodista, ejecutivo de una empresa de petróleos, de donde fue despedido por acoso a las secretarias, e intentó suicidarse. Trabajó como reportero para el London Daily Express y para la Bristol Western Gazette (1908-12). Publicó 27 poemas y su primer relato The Rose Leaf Romance antes de regresar a los Estados Unidos. Participó en la Primera Guerra Mundial y regresó a California, donde viviría ya el resto de su vida. Al morir su madre en 1924, se casó con Pearl Cecily Bowen (Cissy), dieciocho años mayor que él, y el matrimonio duró casi treinta años, hasta el fallecimiento de ella en 1954, aunque no tuvo hijos. En 1933, a los 45 años, se dedicó por entero a escribir. No fue un escritor rápido, y su estilo es muy cuidado y laborioso, frecuente en características rasgos de ingenio caustico que dieron a la novela negra una dignidad literaria desconocida hasta entonces. Su primer cuento fue Blackmailers don't shoot, para la revista Black Mask, un pulp dedicado a los relatos de acción; desde entonces no abandonó el género. Intentó imitar a Dashiell Hammett, pero su estilo es muy diferente; Hammett es seco e impresionista, y Chandler irónico y ciníco. Creó ya por entonces al detective privado Philip Marlowe. Entre 1933 y 1939, produjo 19 relatos.

Div add

La Novela Negra

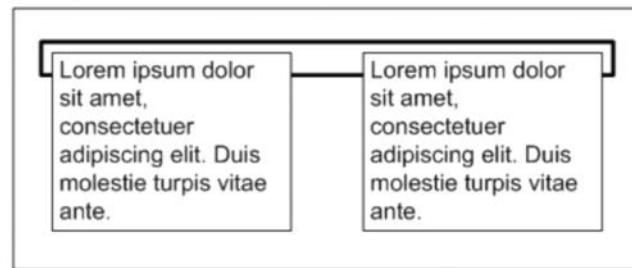
La novela negra (francés: Noir) o hard-boiled es, como la definió Raymond Chandler en su ensayo El simple arte de matar (1950), la novela del mundo profesional del crimen. Debe su nombre a que originalmente fue publicada en la revista Black Mask de Estados Unidos y en la colección Série Noire de la editorial francesa Gallimard, pero también al carácter oscuro de los ambientes en que transcurre, lejos de las casas señoriales que ambientaban las novelas policiacas típicas de la época.

El término se asocia a un tipo de novela policiaca en la que la resolución del misterio no es el objetivo principal y los argumentos son habitualmente muy violentos; la división entre buenos y malos de los personajes se difumina y la mayor parte de sus protagonistas son individuos derrotados y en decadencia en busca de la verdad o, cuando menos, algún atisbo de ella.

Div bio

Elementos flotantes y Divs

Si un div solo encierra elementos flotantes, que desaparecen del flujo normal, actúa como si no tuviera contenido, de modo que los elementos flotantes se salen del elemento contenedor.



Para evitarlo utilizamos la propiedad **overflow** en el elemento contenedor, dándole un valor de hidden o auto, en lo que se conoce como **limpiar los float**.

Clear

lunes, 19 de febrero de 2018 23:05

clear

left
right
both
none

Clear

Clear indica los lados de un elemento en los que no se pueden situar otros elementos flotantes.

<p>
<p>
<p clear:right>

Raymond Thornton Chandler (Chicago, 22 de julio de 1888 - La Jolla, California, 26 de marzo de 1959) fue un escritor estadounidense de novela negra.

Atormentado por la muerte de su madre, tras el divorcio de sus padres fue llevado a Inglaterra, donde recibió una sólida formación literaria. Estudió en el Dulwich College de Londres (1900-05) a clásicos y modernos. También viajó a Francia y Alemania (1905-07) y se nacionalizó británico en 1907. Fue soldado en los Gordon Highlanders de Canadá, empleado de banco, periodista, ejecutivo de una empresa de petróleo, de donde fue despedido por acosar a las secretarias, e intentó suicidarse. Trabajó como reportero para el London Daily Express y para la Bristol Western Gazette (1908-12). Publicó 27 poemas y su primer relato The Rose Leaf Romance antes de regresar a los Estados Unidos. Participó en la Primera Guerra Mundial y regresó a California, donde viviría ya el resto de su vida.

<div.add
float:right>

La Novela Negra

La novela negra (francés: Noir) o hard-boiled es, como la definió Raymond Chandler en su ensayo El simple arte de matar (1950), la novela del mundo profesional del crimen. Debe su nombre a que originalmente fue publicada en la revista Black Mask de Estados Unidos y en la colección Serie Noir de la editorial francesa Gallimard.1 pero también al carácter oscuro de los ambientes en que transcurre, lejos de las casas señoriales que ambientaban las novelas policiacas típicas de la época.

El término se asocia a un tipo de novela policiaca en la que la resolución del misterio no es el objetivo principal y los argumentos son habitualmente muy violentos; la división entre buenos y malos de los personajes se difumina y la mayor parte de sus protagonistas son individuos derrotados y en decadencia en busca de la verdad o, cuando menos, algún atisbo de ella.

Al morir su madre en 1924, se casó con Pearl Cecily Bowen (Cissy), dieciocho años mayor que él, y el matrimonio duró casi treinta años, hasta el fallecimiento de ella en 1954, aunque no tuvo hijos. En 1933, a los 45 años, se dedicó por entero a escribir. No fue un escritor rápido, y su estilo es muy cuidado y laborioso, frecuente en características rasgos de ingenio caustico que dieron a la novela negra una dignidad literaria desconocida hasta entonces. Su primer cuento fue Blackmailers don't shoot, para la revista Black Mask, un pulp dedicado a los relatos de acción, desde entonces abandonó el género. Intentó imitar a Dashiell Hammett, pero su estilo es muy diferente. Hammett es seco e impresionista, y Chandler irónico y caustico. Creó ya por entonces al detective privado Philip Marlowe. Entre 1933 y 1939, produjo 19 relatos.

LIMPIAR FLOATS

- CLEAR: Los elementos que siguen a una caja flotada fluyen a su alrededor, CLEAR CONTROLA ESE COMPORTAMIENTO:
- CLEAR: RIGHT -> NO HAGAS CASO A LAS CAJAS FLOTADAS A LA DERECHA.
- CLEAR: LEFT -> NO HAGAS CASO A LAS CAJAS FLOTADAS A LA IZQUIERDA.
- CLEAR: BOTH -> NO HAGAS CASO A NINGUNA CAJA FLOTADA.

Float y colapso: ClearFix

viernes, 16 de febrero de 2018 19:30

“Cuando un elemento tiene a todos sus hijos flotando, pierde su altura”

OVERFLOW : hidden / auto

(poco eficaz en el caso de menús desplegables)

CLEARFIX AFTER.

Al elemento "colapsado"

```
::after {  
    content : "";  
    display: block;  
    clear: both;  
    // height : 5px  
    color: gray  
}
```



Estilos aplicados a las tablas (1)

Estilos Tablas

Propiedades

height

width

border

background-color

background-image

color

% del total de la pantalla
pixels,

con todas sus
variantes, ya
conocidas

igual que en otros
elementos

border-[]-width
(-top, -right, -
bottom, or -left)

border-[]-style
(-top, -right, -bottom, -left)

border-[]-color
(-top, -right, -bottom, -left)

border
(las tres propiedades)

```
border: 2px green solid
```

Estilos aplicados a las tablas (2)

Estilos Tablas

Propiedades

padding⁽¹⁾

margin⁽²⁾

text-align⁽³⁾

corresponde a los
atributos
descontinuados

igual que en otros
elementos

- 1) cellpadding
- 2) cellspacing
- 3) align

Cell padding

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Cell spacing

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Estilos específicos

Algunos estilos que podemos aplicar a la tabla en su conjunto

empty-cells → show
hide
inherit

1	2
3	

border-spacing → pixels

border-collapse → collapse
separate

controlamos la distancia
entre celdas adyacentes

1	2
3	4

Ejercicio (16b)

Documentos con tablas (2): Usando un editor de HTML (e.g. Bluefish) modificaremos el documento HTML5 creado anteriormente en el que incluimos una tabla.

En este caso controlaremos mediante estilos el tamaño, (fijo o dinámico), y veremos como se ven afectadas las tablas al llenar las celdas y al redimensionar el navegador

Igualmente modificaremos como el aspecto; padding, border, encabezados diferentes, background-color en líneas alternas, :hover, text-align para textos y números

Objetivo: : Utilizar las tablas comprobando como pueden variar si se establecen tamaños relativos Conocer el uso de **estilos** para gestionar el tamaño y el formato de las **tablas**.

Ejercicio (16b): resultado

The screenshot shows a web browser window with the title "Tabla vacaciones" and the URL "localhost/Curso_Web/HTML/16.Tablas/Tabla%20con%20estilos.html". The main content is a table titled "Ejemplo de tabla compleja". The table has a header row "Resultados de la encuesta" and "Edad" with three columns: "12 a 25", "26 a 40", and "Más de 40". The first column contains the question "¿Cuál es tu destino preferido para las vacaciones?". The data rows show percentages for three destinations: Disney-París, El Caribe, and Europa, across the three age groups.

Resultados de la encuesta	Edad		
	12 a 25	26 a 40	Más de 40
Disney-París	50%	25%	25%
El Caribe	25%	50%	25%
Europa	25%	25%	50%

Uso básico del display

Lunes, 19 de febrero de 2018 11:37

1. Etiqueta display
2. Block / Inline
3. Inline-block. Particularidades
4. Visibilidad



Etiqueta display

domingo, 14 de mayo de 2017 13:17

Etiqueta display

display

- **Inline**
- **Block**
 - inline-block
 - run-in (block o inline según el contexto)
- **none**
-
- list-item
- **table**
 - table-caption
 - table-column-group
 - table-header-group
 - table-footer-group
 - table-row-group
 - table-cell
 - table-column
 - table-row
 - inline-table
- **flex**
 - inline-flex

Cambios de modo “en línea/bloque”

Podemos cambiar el modo en que se presenta un objeto, como bloque o en línea, con independencia de cual sea su comportamiento por defecto

display

inline: un elemento actúa como elemento en línea

block: un elemento actúa como elemento en bloque

inline-block: un elemento en bloque fluye como si fuera en línea, pero conserva el resto de sus propiedades en bloque

none: un elemento deja de ser visible

Es muy frecuente aplicar *display: inline* a una lista para conseguir un menú horizontal

```
<ul>
  <li>Home</li>
  <li>Products</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
li {display: inline;
margin-right: 10px;}
```

Home Products About Contact

Block / Inline

domingo, 14 de mayo de 2017 13:36

Elementos en Bloque (*display: block*)

- Ocupan todo el ancho disponible.
- Tienen dimensiones modificables (alto, & ancho).
- No permiten otros elementos a su lado (aunque especifique un ancho, siguen ocupando todo el espacio disponible).

Elementos en Línea (*display: inline*)

- Ocupan el espacio necesario para mostrar su contenido.
- No tienen dimensiones modificables (alto, ancho).
- Permiten otros elementos a su lado.
- *Padding* y *margin* solo empujan a elementos adyacentes en horizontal, NUNCA EN VERTICAL.

Inline-Block

domingo, 14 de mayo de 2017 13:22

DISPLAY: INLINE-BLOCK

- Mezcla de inline y block (sorpresa!!!).
- Tienen dimensiones modificables (alto, ancho).
- Permiten otros elementos a su lado. Si no especifico dimensiones, ocupa lo que ocupe su contenido.

Consideraciones

- VERTICAL ALIGN: Alinea elementos ENTRE ELLOS (no su contenido dentro)
- ESPACIOS "NO DESEABLES" entre elementos y en imágenes dentro de contenedores.

Alineación vertical entre elemento inline-block

The screenshot shows a code editor with two red rectangular containers. Each container has white text centered vertically within it. The text is aligned relative to each other, demonstrating vertical alignment between inline-block elements.

HTML

```
1<div>Lorem ipsum dolor sit amet, consectetur adipisicing elit. ipsa unde ullam omnis? Exercitationem.</div>
2<div>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolores pariatur asperiores expedita adipisci quod aliquam accusantium, tenetur, aspernatur soluta. Cumque, ratione alias alias aliquid illo qui, ipsa unde ullam omnis? Exercitationem.</div>
```

CSS (SCSS)

```
1div{
2  background-color: salmon;
3  color: white;
4  border: 1px solid;
5  height: 20em;
6  width: 20em;
7  display: inline-block;
8 }
```

Inline-block
Vertical-align : baseline (valor por defecto)

Espacios "no deseables" entre elementos

The screenshot shows a code editor with three blue rectangular containers. Each container has white text centered vertically within it. The text is aligned relative to each other, demonstrating vertical alignment between inline-block elements and how whitespace between them is handled.

HTML

```
1<div>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sapiente, ipsam officia blanditiis esse optio vero voluptatum minima necessitatibus, distinctio. Sunt, quis facilis enim! Ipsa illo, aperiam ab vero quibusdam optio.
```

CSS (SCSS)

```
1*{
2  box-sizing: border-box;
3 }
4div{
5  width: 25%;
6  height: 10em;
7  background-color: deepskyblue;
8  color: white;
9  padding: 1em;
10 display: inline-block;
11 }
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sapiente, ipsam officia blanditiis esse optio vero voluptatum minima necessitatibus, distinctio. Sunt, quis facilis enim! Ipsa illo, aperiam ab vero quibusdam optio.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sapiente, ipsam officia blanditiis esse optio vero voluptatum minima necessitatibus, distinctio. Sunt, quis facilis enim! Ipsa illo, aperiam ab vero quibusdam optio.

Los espacios entre los inline-blocks, al tratarlos como si fueran palabras separadas por saltos de linea: el salto de linea genera un espacio

SOLUCIONES

<https://davidwalsh.name/remove-whitespace-inline-block>

I've seen this come up a couple of times lately on Twitter and then [an interesting Dabblet](#) so I figured it would be an important thing to document.

Here's the deal: a series of `inline-block` elements formatted like you normally format HTML will have spaces in between them.

In other words:

```
HTML
<nav>
  <a href="#">One</a>
  <a href="#">Two</a>
  <a href="#">Three</a>
</nav>
```

```
CSS
nav a {
  display: inline-block;
  padding: 5px;
  background: red;
}
```

Particularidades de las imágenes por ser inline-blocks

Hueco bajo la imagen debido a su carácter "in line" de la imagen y a su vertical-align en baseline (valor por defecto)

Posibles soluciones

```
HTML
<div>
  
</div>
```

```
CSS
div{
  border: 1px solid;
  font-size: 2em;
  width: 200px;
}
```

```
HTML
<div>
  
</div>
```

```
CSS
div{
  border: 1px solid;
  font-size: 2em;
  width: 200px;
}
img{
  display: block;
}
```

```
3 font-size: 2em;
4 width: 200px;
5 }
6<img{
7 display: block;
8 }
9 }
```

Visibilidad

visibility → visible (default)
hidden

Tenemos dos formas de ocultar un elemento

display: none

El elemento desaparece como si no existiera

```
li.soon {display: none;}
```

```
<li class="soon">Services</li>
```

Home Products About Contact

visibility: hidden

El elemento no aparece pero si se muestra el espacio que normalmente ocuparía

```
li.soon {visibility: hidden;}
```

```
<li class="soon">Services</li>
```

Home Products

About Contact

Colores

Lunes, 19 de febrero de 2018 11:37

Colores. Codificación. Etiquetas CSS.

Codificación. Colores puros, intermedios y seguros

Definición en CSS COLOR (texto) y BACKGROUND-COLOR)

[Colores de fondo y cajas: Elementos en bloque: posición]

Colores en CSS3.

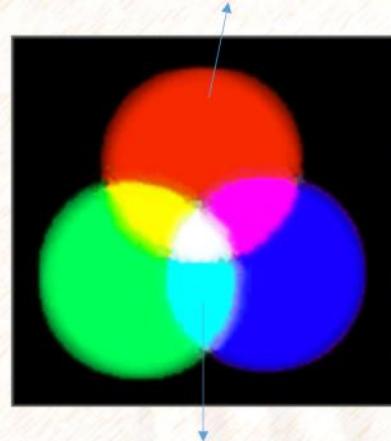
Transparencias. Gradientes.

Colores HSV-HSL.



Diseño Web: Color

Sistema Aditivo
Red, Green Blue (RGB)

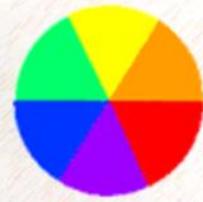


Cyan - Magenta - Yellow – Black
(CMYK)

Sistema sustractivo



Primarios
Red - Yellow - Blue



Secundarios
Green – Orange - Purple

Diseño Web: Color

El **balance de color** puede ayudar a nuestros usuarios a navegar con mayor facilidad nuestra página al delimitar áreas y al agrupar diferente tipo de información

El **contraste**, más que entre un claro y un oscuro, debe de ser muy sutil, muchas veces se logra entre variaciones de un mismo color,

Evitar colores muy **vibrantes** como fondo de página.

No usar colores **fluorescentes** en texto ni en el fondo

Seleccionar una **paleta de color** y mantenerse constante o a ella, utilizar variantes de los colores de esa misma paleta (porcentajes por ejemplo) en vez de introducir nuevos colores

Utilizar el **espacio vacío**, en blanco, como elemento de diseño.

Construir una paleta de colores

Adobe Color CC (<https://color.adobe.com/>)

Antes Kuler.adobe.com



Colores en CSS

Todas las referencias a colores pueden hacerse de tres formas

Por **nombre**, hay 147 colores predefinidos, interpretados de manera similar por cada navegador.

black, red, lime, blue, yellow, aqua, fuchsia, white

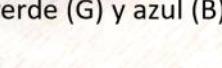
En notación **RGB** cada color es la combinación de Red, Green & Blue, en cantidades que van de 0 a 255

rgb(255,255,255) = Blanco
rgb(255,0,0) = Rojo
rgb(0,0,0) = Negro

En notación **hexadecimal**, cada una de las posibles combinaciones RGB representa en un hexadecimal de 6 cifras, 2 para cada color, comenzando siempre con el signo #

Blanco (255,255,255) = #ffffff
Negro (0,0,0,) = #000000

Colores “puros”: primarios y secundarios

#	rgb	name		
000000	0,0,0	black	Negro	
FF0000	255,0,0	red	Rojo	
00FF00	0,255,0	lime	Verde	
0000FF	0,0,255	blue	Azul	
FFFF00	255,255,0	yellow	Amarillo	
00FFFF	0,255,255	aqua	Cian	
FF00FF	255,0,255	fuchsia	Magenta	
FFFFFF	255,255,255	white	Blanco	

Son las combinaciones todo (255) o nada (0) de rojo (R), verde (G) y azul (B)

Colores intermedios

#	rgb		
000000	0,0,0	Black	
808080	128,128,128	Gris	
800000	128,0,0	Maroon	
800080	128,0,128	Purple	
008000	0,128,0	Green	
808000	128,128,0	Olive	
000080	0,0,128	Navy	
008080	0,128,128	Teal	

#	rgb		
EAEAEA	234,234,234	Gris claro	
COCOCO	192,192,192	Plata (Silver)	
808080	128,128,128	Gris	
4D4D4D	77,77,77	Gris oscuro	
333333	51,51,51	Gris oscuro	

Ejemplo, colores puros e intermedios

Podemos presentar una sucesión de Líneas de colores `<hr>`, con el estilo

- `color: ...;`
- `background-color: ...`
- `height: 3;`
- `width: 50%`

asignando un color a cada línea.

Comprobaremos así los colores puros e intermedios y su apariencia en los distintos navegadores.

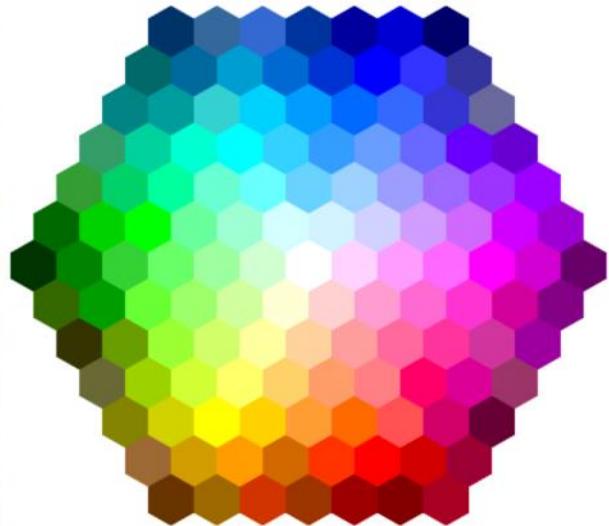
Color Red	<hr/>
Color Fuchsia	<hr/>
Color Purple	<hr/>
Color Yellow	<hr/>
Color Lime	<hr/>
Color Green	<hr/>
Color Teal	<hr/>
Color Olive	<hr/>
Color Maroon	<hr/>
Color Aqua	<hr/>
Color Blue	<hr/>
Color Navy	<hr/>
Color White	<hr/>
Color Silver	<hr/>
Color Gray	<hr/>
Color Black	<hr/>

Conjuntos de colores en HTML

En la época en que muchos monitores soportaban 256 colores, surgió la idea de **colores seguros** (“safe colors”): una lista de 216 colores “estándares”, reservando otros 40 colores fijos para el sistema.

Otra propuesta de estandarización fue, en notación **RGB** limitar las cantidades de Red, Green & Blue, a unos pocos valores admitidos: 0, 51, 102, 153, 204 a 255, con lo que se consigue una paleta de 127 colores

Colores:



Colores definidos en los estilos

Propiedades

background-color:

color

yellow
rgb(255,255,0)
#FFFF00

(corresponde al
foreground color, o
color del texto)

Se pueden incluir estas propiedades en
el estilo de muchas de las etiquetas;
e.g. en la etiqueta body estableceríamos
colores por defecto del documento

```
<body style="color: green; background-color: beige">
```

chapter-11/foreground-color.html

```
/* color name */  
h1 {  
    color: DarkCyan;  
/* hex code */  
h2 {  
    color: #ee3e80;  
/* rgb value */  
p {  
    color: rgb(100,100,90);
```

CSS

color

RESULT

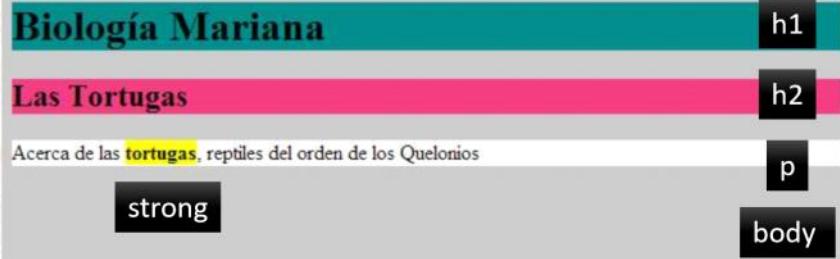
Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

Colores de fondo: elementos como “cajas”

```
<Style> body { background-color:  
rgb(200,200,200);}  
h1 { background-color: DarkCyan; }  
h2 { background-color: #ee3e80; }  
p { background-color: white; }  
strong{ background-color: yellow; }  
</style>
```



Al usar el background-color vemos como los elementos, tanto en bloque como en línea, se comportan **como cajas**

Elementos “en bloque”: posición

Para poder experimentar con los colores,
vamos a adelantar el uso de las posiciones absolutas

position

su valor en “**absolute**” permite que utilicemos las
siguientes coordenadas con respecto al contenedor para
establecer el punto de inicio

top
left

valores en pixeles correspondientes a la posición o
desplazamiento: coordenadas del extremo inicial
(superior izquierdo)
(sin valor cuando *position : static* (default))

Elementos “en bloque”: tamaño y posición (Ejemplo)

```
div {width: 100px;  
      height: 100px;  
      background-color: rgb(255,0,0);  
      margin: 40px;}
```

```
<div></div>
```

```
div {position: absolute;  
      top: 40px;  
      left: 40px;  
      width: 100px;  
      height: 100px;  
      background-color: rgb(255,127,0);}
```

```
<div></div>
```



CSS3: colores y transparencias



CSS3 introduce la opacidad de los colores, que varia entre 0 (0% de opacidad) y 1 (100% de opacidad)

opacity

valor de opacidad del color incluido en el mismo elemento y en los elementos hijos

background-color:

rgba()

Se puede indicar directamente la opacidad mediante la notación rgba, donde un cuarto número corresponde al canal alfa, determinante de la opacidad del fondo.
No afecta a los elementos hijos

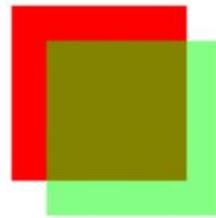
Por compatibilidad (algunos navegadores no soportan canal alfa, deben especificarse dos estilos, el primero sin canal alfa y el segundo en RGB)

Ejemplo de transparencias



```
.uno { width: 100px;  
      height: 100px;  
      background-color: rgb(255,0,0);  
      margin: 40px; }  
.dos { width: 100px;  
      height: 100px;  
      background-color: rgba(0,255,0, 0.5);  
      position: relative;  
      top: 20px;  
      left: 20px; }
```

```
<div class="uno">  
    <p class="dos"></p>  
</div>
```



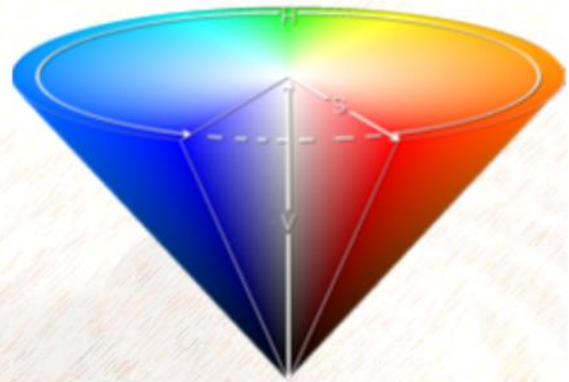
Colores HSV - HSL

Al margen del RGB existe otro modelo de definición del color en términos de sus **componentes**, conocido como

HSV: *Hue, Saturation, Value*
(Matiz, Saturación, Valor)

HSB: *Hue, Saturation, Brightness*
(Matiz, Saturación, Brillo)

fue creado en 1978 por *Alvy Ray Smith* y es una transformación no lineal del espacio de color RGB, que se puede expresar como progresiones de color



En la misma línea se encuentra el modelo

HSL (*Hue Saturation, Lightness* – **Matiz, Saturación, Luminosidad**)

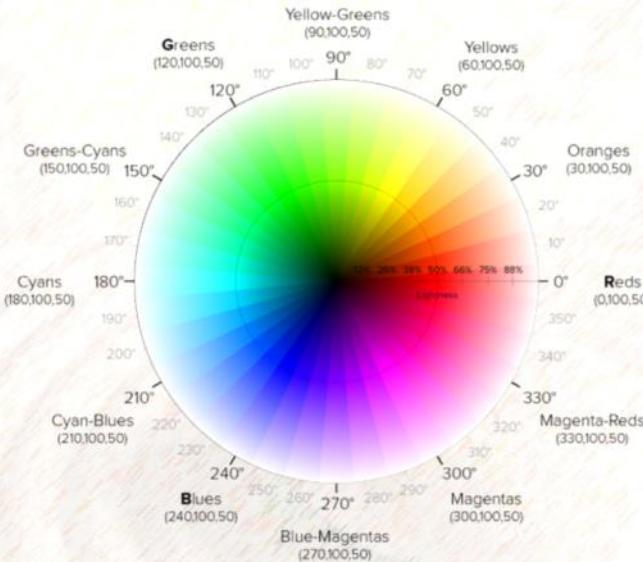


Colores HSL

En CSS3 se puede definir un color en términos de

- **Matiz (H)** medida como ángulo entre 0 y 360
- **Saturación (S)** expresada en %
- **Luminosidad (L)** también en %, entre el 0% (negro) y el 100% (blanco)
- y opcionalmente, el valor del canal **alfa**

hsl(x,y%,z%)
hsla(x,y%,z%,a)



```
background-color: hsl(0,0%,78%)
background-color: hsla(0,0%,78%,0.5)
```

CSS3: *gradientes lineales*



Propuesta de CSS3 aún no homogeneizada entre los navegadores

background: linear-gradient(direction, color-1, color-2, ...);

```
background: -webkit-gradient(linear, 0% 0%,  
0% 100%, from(#66cccc), to(#336666));  
background: -webkit-linear-gradient(#336666, #66cccc);  
background: -o-linear-gradient(#336666, #66cccc);  
background: -moz-linear-gradient(#336666, #66cccc);  
background: linear-gradient(#336666, #66cccc);
```

colores: Cualquiera de los sistemas (pueden incluir transparencias)

direction	To bottom (default) To top To right To left	To bottom right To top right To right top To left bottom ...
-----------	--	---

Ejemplo: gradiente lineal



Propuesta de CSS3 aún no homogeneizada entre los navegadores

```
#gradient {  
/* fallback color */  
background-color: #66cccc;  
/* fallback image */  
background-image: url(images/fallback-image.png);  
/* Firefox 3.6+ */  
background: -moz-linear-gradient(#336666, #66cccc);  
/* Safari 4+, Chrome 1+ */  
background-image: -webkit-gradient(linear, 0% 0%, 0% 100%,  
from(#66cccc), to(#336666));  
/* Safari 5.1+, Chrome 10+ */  
background: -webkit-linear-gradient(#336666, #66cccc);  
/* Opera 11.10+ */  
background: -o-linear-gradient(#336666, #66cccc);  
background: linear-gradient(#336666, #66cccc);  
height: 150px;  
width: 300px;}
```

Primero fijamos valores de fondo por si no hay gradientes o no se puede cargar imagen



CSS3: gradientes radiales



Propuesta de CSS3 aún no homogeneizada entre los navegadores

background: radial-gradient(**forma** **tamaño** **posición**, **color-1**, **color-2**, ...);

colores: Cualquiera de los sistemas (pueden incluir transparencias)

forma circle o ellipse

tamaño closest-side
posición farthest-side
closest-corner at % %
farthest-corner

```
background: radial-gradient(circle, #336666, #66cccc);
```



Herramientas CSS

Lunes, 19 de febrero de 2018 11:37

1. Herramientas de formato
2. Herramientas de color

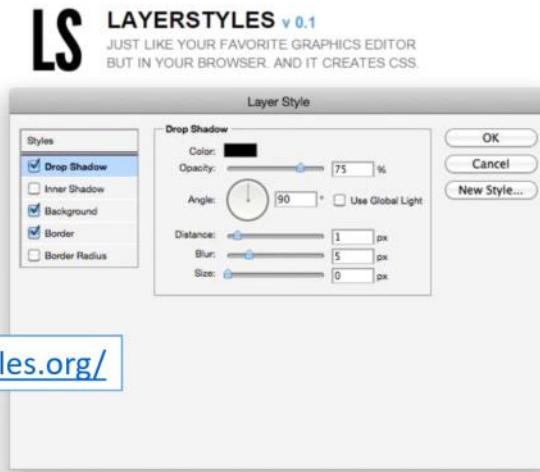


Herramientas de formato

lunes, 8 de mayo de 2017 12:28

Herramientas de formato (1)

Existen sitios web donde diseñamos visualmente los bordes, sombras y fondos Y obtenemos el código CSS correspondiente



<http://layerstyles.org/>

Herramientas de formato (2)



The ultimate *CSS tools* for web designers



Gradient Generator

Use multiple colors and opacity stops to get amazing gradients.

These gradients can be used as background images of banners, wallpapers, buttons or tables and in many other applications.

Border Radius

Super easy to use and a super time saver. Change all the borders selected at the same time.

Noise Texture

Create subtle background patterns with dirty pixels and noise, changing the color and values and previewing the results in real time.

Box Shadow

Blur radius changes, color changes, shadow size. Everything that you need to create great drop shadows in a single place.

<http://www.cssmatic.com/>

<http://www.cssmatic.com/>

<http://www.css3maker.com/>

Herramientas de colores

domingo, 25 de febrero de 2018 10:39

Selección de colores

Color pickers: Herramientas on-line que permiten seleccionar un color y conocer el valor correspondiente en los distintos formatos de representación



<http://hslpicker.com/>

<http://hslpicker.com/>

Imágenes

lunes, 19 de febrero de 2018 11:37

Imágenes. y CSS

Tamaño. Modelo de caja: inline-block.

Posicionamiento: Float. Clear

Fragmentos de imágenes

Imágenes como fondos. Formas de uso. Posiciones.

CSS3. Múltiples imágenes como fondos

Imágenes como títulos.



Imágenes

height

width

tamaño de la imagen (como cualquier otro elemento)

- permite dimensiones diferentes de las propias de la imagen
- incluso si son las mismas, agiliza la carga de las páginas y mantiene la estructura en caso de problemas con la carga de una imagen
- permite reducir el numero de diferentes tamaños empleados, homogeneizando la página web y ajustando mejor las imágenes a la disposición (*layout*)

Ejemplo de 3 tamaños, que podrían corresponder a las clases grande, mediana y pequeña



Posicionamiento de Imágenes

En la mayoría de los navegadores, las imágenes son elementos **inline-block**

margin

float

clear

Imágenes flotando y con su margen, igual que los párrafos que las acompañan

igual que en el caso de los párrafos



Si queremos que su comportamiento sea el de un bloque podemos usar

display:block

E.g. para centrar una imagen indicando margin:auto

Ejemplos: Imágenes y bloques

Imágenes y css



Imagen antes de un párrafo, sin aplicarle ningún estilo. Se comporta como elemento inline-block, por lo que ocupa todo el ancho de su contenedor (aunque no se ve)



La misma imagen dentro de un div al que se le aplica un margen; el div se adapta al tamaño del elemento inline-block, por lo que de nuevo ocupa todo el ancho.

Ejemplos: Imágenes y bloques



La misma imagen aplicandole un margen: aunque el margen rodea solo la imagen, el bloque que la contiene (el elemento inline-block) sigue ocupando todo el ancho



Imagen a la que aplicamos la propiedad float; el espacio que ocupa el elemento corresponde al tamaño de la imagen y el texto puede fluir en torno a él. Por tanto, ante el float la imagen se comporta completamente como un bloque de tamaño definido.

Ejemplo Float

```
<figure float:left>  
<p>  
<p>
```

La Novela Negra



La novela negra (francés: Noir) o hard-boiled es, como la definió Raymond Chandler en su ensayo *El simple arte de matar* (1950), la novela del mundo profesional del crimen. Debe su nombre a que originalmente fue publicada en la revista *Black Mask* de Estados Unidos y en la colección *Série Noire* de la editorial francesa Gallimard. I pero también al carácter oscuro de los ambientes en que transcurre, lejos de las casas señoriales que ambientaban las novelas policiacas típicas de la época.

El término se asocia a un tipo de novela policiaca en la que la resolución del misterio no es el objetivo principal y los argumentos son habitualmente muy violentos; la división entre buenos y malos de los personajes se difumina y la mayor parte de sus protagonistas son individuos derrotados y en decadencia en busca de la verdad o, cuando menos, algún atisbo de ella.

```
<figure float:right>  
<p>  
<p>
```

La Novela Negra

La novela negra (francés: Noir) o hard-boiled es, como la definió Raymond Chandler en su ensayo *El simple arte de matar* (1950), la novela del mundo profesional del crimen. Debe su nombre a que originalmente fue publicada en la revista *Black Mask* de Estados Unidos y en la colección *Série Noire* de la editorial francesa Gallimard. I pero también al carácter oscuro de los ambientes en que transcurre, lejos de las casas señoriales que ambientaban las novelas policiacas típicas de la época.

El término se asocia a un tipo de novela policiaca en la que la resolución del misterio no es el objetivo principal y los argumentos son habitualmente muy violentos; la división entre buenos y malos de los personajes se difumina y la mayor parte de sus protagonistas son individuos derrotados y en decadencia en busca de la verdad o, cuando menos, algún atisbo de ella.



Ejemplo de Clear

```
<figure float:left>
<p>
<p clear:left>
```

La Novela Negra



La novela negra (francés: Noir) o hard-boiled es, como lo definió Raymond Chandler en su ensayo El simple arte de matar (1950), la novela del mundo profesional del crimen. Debe su nombre que originalmente fue publicada en la revista Black Mask de Estados Unidos y en la colección Série Noire de la editorial francesa Gallimard, pero también al carácter oscuro de los ambientes en que transcurre, lejos de las casas señoriales que ambientaban las novelas policiacas típicas de la época.

El término se asocia a un tipo de novela policiaca en la que la resolución del misterio no es el objetivo principal y los argumentos son habitualmente muy violentos, la división entre buenos y malos de los personajes se difumina y la mayor parte de sus protagonistas son individuos derrotados y en decadencia en busca de la verdad o, cuando menos, algún alivio de ella.

Partes de imágenes

Clip <shape>

rect (top, right, bottom, left)

Permite definir un rectángulo, en posición absoluta, correspondiente a la única parte de una imagen que será

Dragon de Komodo..



```
div {position:absolute;  
clip: rect(10px,350px,200px,250px);}
```

Imágenes como fondos

jueves, 11 de mayo de 2017 13:34

Imágenes como fondos

background-image → url (url de la imagen)

Se pueden incluir estas propiedades en el estilo de muchas de las etiquetas;

- En la etiqueta body estableceríamos el fondo por defecto del documento

```
<body style="background-color: beige;  
background-image: granite.gif;">
```

- En un div, establecemos el fondo de un área determinada ...

A nivel de diseño hay que ser muy cuidadoso con el uso de imágenes de fondo

Es fundamental el **contraste**



Imágenes y colores

1

background-color

Colores planos, anteriores a CSS3

background-color

Degrados y transparencias (CSS3)

2

Puede no funcionar en navegadores antiguos

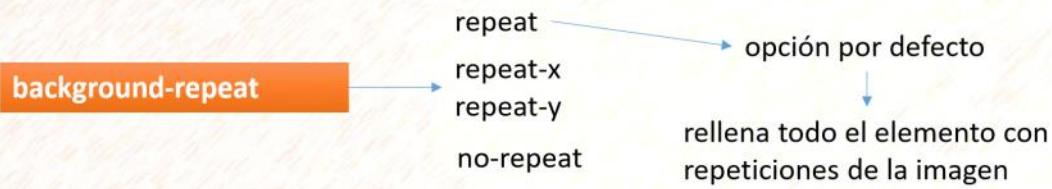
background-image

Imágenes en sus diversos formatos

Puede haber problemas con la imagen

Para evitar problemas de contraste siempre se especifica 1 + 2

Gestión de imágenes como fondos



- permite usar imágenes muy pequeñas, con muy poca carga sobre el tráfico de red
- Especialmente importante en móviles

`background-clip: content-box;`

Rellena solo el área del contenido,
Respetando el padding

`no-repeat`

`background-size`

Determina el tamaño
de la imagen única,

longitud
porcentaje
cover
contain

Imágenes como fondos: posición

background-position

Coordenadas en px 0 %
de la esquina sup-izda

Muy útil para el uso de sprites



background-attachment

fixed
scroll

Todos los datos del fondo

background

atajo para indicar de una sola vez todos los datos del fondo

- 1: background-color
- 2: background-image
- 3: background-repeat
- 4: background-attachment
- 5: background-position

```
body {  
background: #ffffff  
url("images/tulip.gif")  
no-repeat top right;}
```

RESULT

Planting Guide

Magnolia

Magnolia grandiflora, commonly known as the Southern magnolia or bull bay, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large, fragrant flowers. Widely cultivated around the world, over a hundred cultivars have been bred and marketed commercially. The timber is hard and heavy, and has been used commercially to make furniture, pallets, and veneer.



Ranunculus

Ranunculus asiaticus (Persian Buttercup) is a species of buttercup (Ranunculus) native to the eastern Mediterranean region in southwestern Asia, southeastern Europe (Crete, Karpathos and Rhodes), and northeastern Africa. It is a herbaceous perennial plant, growing to 45 cm tall, with simple or branched stems. The basal leaves are three-lobed, with leaves higher on the stems more deeply divided; like the stems, they are downy or hairy. The flowers are 3–5 cm diameter, variably red to pink, yellow, or white, with one to several flowers on each stem.

Tulip

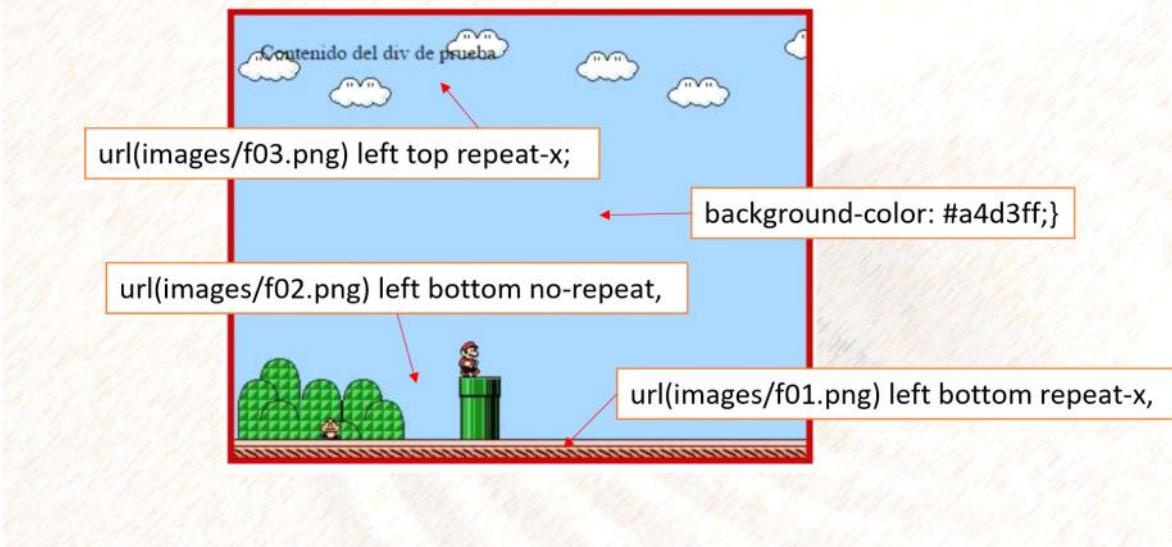
Tulipa gesneriana L. or "Döder's tulip" is a plant belonging to the family of Liliaceae. This species has uncertain origins, possibly from Asia, and has become naturalized in south-west Europe. Most of the cultivated species, subspecies and cultivars of tulip are derived from *Tulipa gesneriana*. The flower and bulb can cause dermatitis through the allergen, tulipside A, even though the bulbs may be consumed with little ill-effect. The sweet-scented bisexual flowers appear during April and May. Bulbs are extremely resistant to frost, and can tolerate temperatures well below freezing - a period of low temperature is necessary to induce proper growth and flowering, triggered by an increase in sensitivity to the phytohormone auxin. The bulbs may be dried and pulverised and added to cereals or flour.

NOTA: la etiqueta background sobrescribe todo el background

Varias imágenes como fondos

Se pueden usar diversas imágenes con diversos comportamientos para construir un único fondo

Background: → url (url de la imagen), url (url de la imagen), ...



Varias imágenes: código

```
{  
background: url(images/suelo.png) left bottom repeat-x,  
url(images/mario.png) left bottom no-repeat,  
url(images/nubes.png) left top repeat-x;  
background-color: #a4d3ff;  
}  
  
{  
background-image: url(images/suelo.png),  
    url(images/mario.png), url(images/nubes.png);  
background-position: left bottom, left bottom, left top;  
background-repeat: repeat-x, no-repeat, repeat-x;  
background-size: auto, 400px, auto;  
background-color: #a4d3ff;  
}
```

Imágenes como títulos

domingo, 25 de febrero de 2018 11:48

Imágenes como títulos

Al sustituir un título por una imagen con el correspondiente texto de título, es importante no eliminar el contenido de h1, sino ocultarlo, de cara a su utilización en procesos de indexado.

Para ello pueden usarse estilos y código como los que aparecen a continuación

```
h1 {  
    width: 450px;  
    height: 100px;  
    background: #FFF url("/imagenes/titular.png") no-repeat top left;  
}  
h1 span {  
    display: none;  
}
```

```
<h1><span>Título de la página</span></h1>
```



- Herramientas y
 - Gestionando paquetes con npm
 - Automatizando tareas con webpack
 - Chrome Developer Tools: ese gran desconocido
- Metalenguajes CSS
 - SASS
 - Bootstrap/Materialize

Scripts de npm

jueves, 29 de marzo de 2018 21:05

Podemos ejecutar comandos de un paquete mediante la definición de scripts en el archivo package.json

'npm start' no necesita especificar 'run', es el script standar para lanzar un proyecto npm

```
"scripts": {  
  "sass": "node-sass -o ./dist/ ./src/",  
  "stylus": "stylus -w ./src/style.styl -o ./dist/style.css",  
  "start": "supervisor main.js"  
}  
  
$ > npm run sass  
$ > npm run stylus  
$ > npm start
```



podemos reemplazar los gestores de tareas (*task runners*) por simples scripts NPM para automatizar tareas.

<https://www.keithcirkel.co.uk/how-to-use-npm-as-a-build-tool/>

Keith Cirkel
JavaScript Cyber Shepherd

[Twitter](#) [GitHub](#)

09 Dec 2014 in Node.js, npm, JavaScript

How to Use npm as a Build Tool

```
package.json - Visual Studio Code  
Archivo Sección Ver Ir Ayuda  
package.json  
1 {  
2   "name": "boilerplate-npm",  
3   "version": "1.0.8",  
4   "description": "Boilerplate de Automatización y Optimización de Tareas Front end con Scripts npm",  
5   "main": "package.json",  
6   "scripts": {  
7     "test": "echo \"From no test specified\" && exit 1",  
8     "imagein": "node-sass node-sass-cil babel-cil babel-preset-latest pug pug-cil browser-sync parallelshell onchange uncss postcss-cil autoprefixer buildify useref useref-cil html-minifier copy imagein imagein-cil imagein-pngquant imagein-jpeg-recompress",  
9     "imagein-ws": "imagein-ws imagein-ws-sugo imagein-gificle rimraf mkdirp",  
10    "dev:proxy": "node i -D mode=sass mode=sass-cil babel-cil babel-preset-latest pug pug-cil browser-sync parallelshell onchange uncss postcss-cil autoprefixer buildify useref useref-cil html-minifier copy imagein imagein-cil imagein-pngquant imagein-jpeg-recompress",  
11    "sass": "node-sass -w -o ./src/css/ ./src/scss/",  
12    "babel": "babel -w ./src/js/ -d ./src/jst",  
13    "pug": "pug -c ./src/css --no-lint ./src",  
14    "watch:dev": "browser-sync start --server --directory --startPath /src",  
15    "dev:proxy": "browser-sync start --proxy localhost --directory --startPath /src/taller-frontend/boilerplate-npm/src --serveStatic / --files ./src",  
16    "watch:metatag": "parallelshell \\"npm run sass\\" \\"npm run babel\\" \\"npm run pug\\"",  
17    "reload": "browser-sync reload",  
18    "watch:reload": "onchange ..src -- npm run reload",  
19    "watch:devserve": "parallelshell \\"npm run watch:metalang\\" \\"npm run dev:server\\" \\"npm run watch:reload\\"",  
20    "watch:metalang": "parallelshell \\"npm run sass\\" \\"npm run dev:proxy\\"",  
21    "uncssfiles": "ncss ./src/*.html > ./dist/css/styles.css",  
22    "autoprefixer": "postcss -u autoprefixer --autoprefixer.browsers > 5%, le 10%" -r ./dist/css/styles.css",  
23    "mincss": "node buildify",  
24    "useref": "useref ./src/*/*+>html.php \"/dist",  
25    "htmlmin": "html-minifier --htmls --remove-comments --collapse-whitespace --input-dir ./dist --output-dir ./dist/",  
26    "static": "node static \"/dist/img/*.(img|png)\" -o ./dist/img/ -p pngquant -p jpeg-recompress",  
27    "imagein": "imagein \"/src/img/*.(img|png)\" -o ./dist/img/ -p webp",  
28    "webp": "imagein \"/src/img/*.(svg)\" -o ./dist/img/ -p svgo",  
29    "svgo": "imagein \"/src/img/*.(gif)\" -o ./dist/img/ -p gificle",  
30    "gificle": "imagein -resize \"300x\" \"/dist/img/*.(jpg",  
31    "imagein": "imagein \"/src/img/*.(png)\" -o ./dist/img/ -p imagein",  
32    "clean": "rimraf ./dist/*",  
33    "build:folders": "mkdirp ./dist/js ./dist/img",  
34    "build:frontend": "npm run uncssfiles && npm run autoprefixer && npm run minicss-js && npm run useref && npm run htmlmin",  
35    "build:media": "npm run static && npm run imagein && npm run webp && npm run svgo && npm run gificle",  
36    "build:dist": "npm run clean && npm run build:folders && npm run build:frontend && npm run build:media",  
37    "serve": "browser-sync start --server --directory --startPath ./dist --serveStatic ./dist --files ./dist",  
38    "proxy": "browser-sync start --proxy localhost --directory --startPath /src/taller-frontend/boilerplate-npm/dist --serveStatic / --files ./dist"  
39  },  
40  "author": "Jonathan Mircha <jonmircha@gmail.com>",  
41  "license": "MIT"  
42}  
43
```

Identificación de las tareas

jueves, 29 de marzo de 2018 21:14



- Tareas de Compilación

- Sass
- Babel
- Pug...

- Tareas en Tiempo Real

- Servidor Web
- Servidor Proxy
- Recarga en vivo
- Observar cambios

- Tareas de Publicación

- CSS

- Concatenación
- Depuración
- Prefijos
- Minificación

- JS

- Concatenación
- Minificación
- Ofuscación

- HTML

- Reemplazar referencias
- Minificación

- Archivos

- Archivos Estáticos
- Limpieza de archivos
- Creación de carpetas

- Imágenes

- Optimización
- Compresión
- Conversión
- Redimensión

node-sass
babel-cli
babel-preset-latest
pug
pug-cli
browser-sync
parallelshell
rimraf
mkdirp
uncss
postcss-cli
autoprefixer
buildify
useref
useref-cli
html-minifier
copy
imagemin
imagemin-cli
imagemin-pngquant
imagemin-jpeg-recompress
imagemin-webp
imagemin-svg
imagemin-gifsicle

Scaffolding

jueves, 29 de marzo de 2018 21:26

```
src
  es6
  img
  templates
  scss
  statics
dist
  css
  js
  img
```

Sass - scss

jueves, 29 de marzo de 2018 21:30

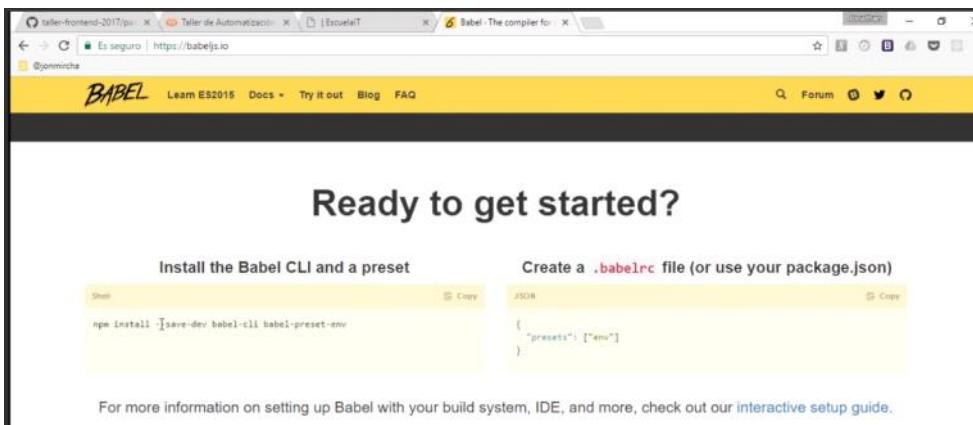
```
npm install -D node-sass babel-cli babel-preset-latest parallelshell
```

al mismo nivel que el package.json se crea .babelrc (se crea desde VSC)

```
{  
  "presets": [ "latest" ],  
  "plugins": []  
}
```

```
watch  
↓  
"sass": "node-sass -w -o ./dist/css ./src/scss",  
output
```

```
watch  
↓  
"babel": "babel -w ./src/es6/ -d ./dist/js/",  
output
```

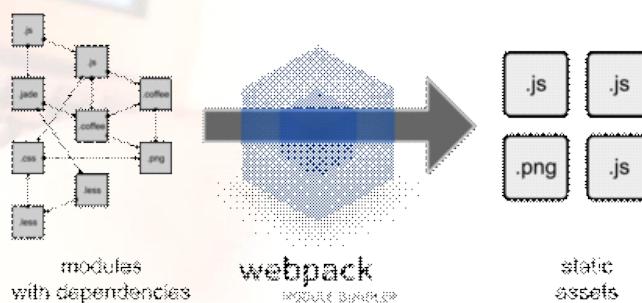


agrupación de las tareas

```
"watch:metalang": "parallelshell \"npm run sass\" \"npm run babel\"",
```

WebPack

martes, 27 de marzo de 2018 11:40



webpack es una **herramienta de agregación** de recursos para aplicaciones web

permite generar una distribución única a partir de un conjunto establecido de **assets**.

<https://webpack.js.org/>

Entre los recursos que es capaz de gestionar webpack, podemos encontrar

formatos soportados por defecto

- HTML
- JavaScript
- CSS,

formatos que necesitan ser transformados

- SASS
- TypeScript*
- Jade

Con webpack definiremos un único **pipeline de procesamiento de recursos**,

obteniendo un **único fichero** empaquetado y minificado con todos los recursos necesarios para nuestro desarrollo

Documentación

<https://webpack.js.org/concepts/>

Sponsor webpack and get apparel from the [official shop](#) or get stickers [here!](#) All proceeds go to our open collective!

webpack v4.4.1

DOCUMENTATION CONTRIBUTE VOTE BLOG

CONCEPTS CONFIGURATION API GUIDES LOADERS PLUGINS

Concepts

At its core, webpack is a static module bundler for modern JavaScript applications. When webpack processes your application, it recursively builds a *dependency graph* that includes every module your application needs, then packages all of those modules into one or more *bundles*.

Learn more about [JavaScript modules](#) and [webpack modules](#) [here](#).

Since v4.0.0 webpack does not require a configuration file. Nevertheless, it is [incredibly configurable](#). To get started you only need to understand four **Core Concepts**:

- Entry
- Output
- Loaders
- Plugins

This document is intended to give a **high-level** overview of these concepts, while providing links to detailed concept specific use cases.

Entry

An **entry point** indicates which module webpack should use to begin building out its internal *dependency graph*. After entering the entry point, webpack will figure out which other modules and

ag-grid is proud to partner with webpack



Más información

<https://www.genbetadev.com/javascript/webpack-gestion-integrada-y-eficiente-de-tus-assets>

<https://carlosazaustre.es/primeros-pasos-con-webpack/>

Instalacion

martes, 27 de marzo de 2018 19:04

Prerrequisitos

- *NodeJS*
- *npm*

Instalación global

```
npm install webpack -g
```

Instalación específica en un proyecto

```
npm install webpack --save-dev  
npm install webpack-cli --save-dev
```

La instrucción que ejecutará webpack será

```
node_modules/.bin/webpack
```

Ejemplo básico

martes, 27 de marzo de 2018 19:08

Elementos del proyecto

index.html
app.js

```
<!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>Uso de Webpack</title>
</head>
<body>
  <header><h1>Uso de Webpack</h1></header>
  <script src="bundle.js"></script>
</body>
</html>
```

referencia la bundle, resultante
del uso de webpack

```
(function () {
  console.log("Ejemplo del uso de Webpack")
})()
```

script npm
(incluido en package.json)

"build": "webpack app.js -o bundle.js"

A diferencia de versiones anteriores, el fichero
de salida se indica mediante --output (-o)

El comando se ejecuta como

```
npm run build
```

```
> webpack app.js -o bundle.js

Hash: 487a20a73616cf0ffe96
Version: webpack 4.3.0
Time: 104ms
Built at: 2018-3-29 23:12:40
    Asset      Size  Chunks             Chunk Names
  bundle.js  586 bytes     0  [emitted]  main
Entrypoint main = bundle.js
          [0] ./app.js 69 bytes {0} [built]
```

Fichero de configuración

jueves, 29 de marzo de 2018 23:15

```
const path = require('path');

webpack.config.js      module.exports = {
                        entry: './app.js',
                        output: {
                            path: path.resolve(__dirname, 'dist'),
                            filename: 'bundle.js'
                        },
                        mode: 'development'
};

"build": "webpack --config webpack.config.js"
```

```
app.js                import { addMensaje } from './module.js'
                      addMensaje()
```

```
module.js             export function addMensaje() {
                      console.log("Ejemplo del uso de Webpack")
}
```

```
> webpack --config webpack.config.js

Hash: 9cae6e6282eb9287eb4a
Version: webpack 4.3.0
Time: 116ms
Built at: 2018-3-29 23:39:22
          Asset      Size  Chunks      Chunk Names
bundle.js   3.57 KiB   main  [emitted]  main
Entrypoint main = bundle.js
[./app.js]  54 bytes {main}  [built]
[./module.js]  82 bytes {main}  [built]
```

Modo "escucha"

jueves, 29 de marzo de 2018 23:48

```
"scripts": {  
  "build": "webpack --watch --colors"  
}
```

Al correr npm run build tendremos ejecutando webpack en modo watch lo que significa que a cada cambio que hagamos en nuestro código, webpack creará el bundle de nuevo pero de una manera más rápida que por ejemplo con Browserify o Gulp ya que lo mantiene en memoria. Y con el flag --colors vemos de una manera más gráfica el resultado en la consola:

Elementos de la configuración

jueves, 29 de marzo de 2018 23:40

Entry
Output
Loaders
Plugins
Mode

punto o puntos de entrada
Ejemplo de entrada múltiple

Nombre de fichero
Path absoluto

- Development
- Production

```
const config = {  
  entry: {  
    app: './src/app.js',  
    vendors:  
      './src/vendors.js'  
  }  
};  
  
const config = {  
  output: {  
    filename: 'bundle.js',  
    path: '/home/proj/public/assets'  
  }  
};
```

Loaders

jueves, 29 de marzo de 2018 23:45

Webpack permite que trabajemos con cualquier tipo de archivo (CSS, preprocesadores CSS, preprocesadores de JavaScript, imágenes, etc...) Con indicarle que *loader* debe utilizar e instalarlo, nos es suficiente.

```
module.exports = {
  module: {
    rules: [
      { test: /\.css$/, use: 'css-loader' },
      { test: /\.ts$/, use: 'ts-loader' }
    ]
  }
};
```

Cualquier recurso utilizado en la aplicación debe definirse y ser tratado como un módulo, independientemente de si es un fichero JavaScript, CSS o cualquier otro recurso.

Para que los estilos CSS sean minificados e incluidos en la distribución generada, tendremos que

- hacer un **require/import** de estos ficheros desde el punto de entrada principal (app.js) o
- incluirlos como "entry point"

Lo mismo pasará si queremos incluir alguna imagen, por ejemplo.

De esta manera, webpack los procesará y permitirá su aplicación y carga desde JavaScript directamente, evitando el tener referenciados los ficheros CSS mediante link o style dentro de la página HTML principal y minimizando así el número de peticiones que se hacen al servidor para completar la carga de la página.

Pre-procesadores de CSS : SASS

En el caso de sass, además de style-loader y css-loader, se necesitan node-sass y el loader sass-loader

```
npm install style-loader css-loader sass-loader node-sass --save-dev
```



```
app.js → import '../scss/style.scss'
```

Plugins

jueves, 29 de marzo de 2018 23:44

Junto con los *loaders*, que permiten incluir en el proceso diferentes tipos de archivos, existen *plugins* destinados a realizar cualquier otro tipo de tarea, como la optimización y minificación en respuesta a diversas variables de entorno.

Los *plugins* son incluidos mediante un *require* y se instancian y se configuran en la opción *plugins*

Excluir el CSS del *bundle*

Especialmente en **producción**, mezclar dos tipos de outputs como son el procesamiento de JavaScript y el CSS, no es del todo deseable, ya que es importante favorecer el **cacheo de los recursos** en el navegador y si enviamos todos los recursos juntos, este fichero completo va a cambiar muchas más veces y se deberá descargar completo cada vez.

Para que los estilos se generen un fichero .css separado, tendremos que usar un plugin para webpack llamado **extract-text-webpack-plugin**. Este plugin permite detectar todas las definiciones de estilos, procesarlas y extraerlas del bundle.js general, permitiendo ser guardadas en el fichero que especifiquemos:

```
npm install -D extract-text-webpack-plugin@next
```

```
const ExtractTextPlugin = require("extract-text-webpack-plugin");

const extractSass = new ExtractTextPlugin({
  filename: "style.css",
  // disable: process.env.NODE_ENV === "development"
});

module.exports = {
  ...
  module: {
    rules: [...]
  }
},
plugins: [
  extractSass
];
}

use: extractSass.extract({
  use: [
    {loader: "css-loader",},
    {loader: "sass-loader"}
  ],
  // use style-loader in development
  fallback: "style-loader"
})
```

Se mantiene la importación del CSS desde JS, aunque luego se extraiga

```
app.js → import '../scss/style.scss'
```

Se añade la llamada al fichero CSS desde el HTML

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

Servidor de desarrollo

jueves, 29 de marzo de 2018 23:51

También tenemos la opción de crear un servidor web de desarrollo con webpack. Para ello debemos instalar otra dependencia que es webpack-dev-server:

```
npm install --save-dev webpack-dev-server
```

Y modificar nuestro fichero webpack.config.js con el siguiente bloque:

```
devServer: {  
    host: '0.0.0.0',  
    port: 8080,  
    inline: true  
},
```

Esto nos va a crear un servidor (basado en Node.js y Express) de desarrollo en local, en el puerto 8080 que servirá nuestra carpeta raíz.
También se le puede indicar la carpeta inicial del servidor

```
contentBase: path.join(__dirname, "dist"),
```

Para iniciararlo creamos un nuevo script npm

```
"start" : "node_modules\\.bin\\webpack-dev-server",
```

Angular eject

viernes, 30 de marzo de 2018 12:15

<https://github.com/angular/angular-cli/wiki/eject>

Extrae la aplicación del contexto de angular-cli y en su lugar expone la configuración y los scripts de webpack

El comando ng eject tiene los mismos modificadores que ng build para generar la adecuada configuración inicial de web pack



aot
app
base-href
deploy-url
environment
extract-css
force
i18n-file
i18n-format
locale
missing-translation
output-hashing
output-path
poll
progress
sourcemap
target
vendor-chunk
common-chunk
verbose
watch

pre-procesadores de CSS

Los pre-procesadores CSS nos permiten crear hojas de estilo de una **manera dinámica**.

- la hoja de estilos se escribe en un lenguaje más eficaz que CSS, específico del pre-procesador
- este lenguaje se procesa para ser convertido en CSS estándar, que es aplicado por el navegador

Entre los más conocidos se encuentran **SaaS** y **LESS**, usados en Bootstrap, junto con **Stylus**, usado por Express/Node.JS



características comunes

Aunque cada uno emplea su propio lenguaje, con una sintaxis específica, se a parecen a CSS, del que toman muchos elementos, incluyendo los selectores y las propiedades

tienen como objetivo ampliar las funcionalidades de las hojas de estilo, de por si limitadas

siempre generan al final un código CSS 100% compatible con los estándares

Ventajas

- Sintaxis más limpia para selectores compuestos
- Nuevas funcionalidades (variables, funciones...)
- Mejoras en la gestión de ficheros
- Reutilización de código
- Flexibilidad y facilidad de mantenimiento

Ejemplo → Bootstrap

SASS

El otro pre-procesador CSS también utilizado por Bootstrap es SASS (*Syntactically Awesome StyleSheets*)

Su funcionamiento es similar a LESS, con el uso de variables, anidamiento de estilos y *mixins*.

Existen las lógicas variaciones en cuanto a la sintaxis (e.g las variables comienzan por \$)



<http://sass-lang.com/>

Es posible utilizarlo de 3 formas

1. en la consola de comandos
2. como módulo de Ruby,
3. como plugin de cualquier framework compatible con Rack (como por ejemplo Ruby on Rails y Merb).

LESS



<http://lesscss.org/>

LESS es una ampliación de las hojas de estilo CSS, para que puedan funcionar como un lenguaje de programación

- permitiendo el uso de variables,
- funciones,
- operaciones aritméticas, entre otras,

para acelerar y enriquecer los estilos en un sitio web.

Aspectos diferenciales

LESS es mínimo en sus dependencias y dispone de una versión para el lado cliente

LESS tiene una excelente documentación en su sitio Web

LESS es muy flexible en su sintaxis

Sass

domingo, 14 de mayo de 2017 19:41

<http://sass-lang.com/>

Es una sintaxis para el marcado de estilos

Exporta su funcionalidad a CSS estándar

- Anidado de propiedades
- Funciones y variables
- Importación de hojas de estilo y parciales
- Extensiones y herencia
- Operadores

Tiene 2 **sintaxis** diferentes:

SASS

primera sintaxis empleada por este preprocesador, que tenía similitud con el lenguaje "Haml".
sin ; y {}



body

```
width: 80%
margin: auto
article
  width: 50%
  background-color: bisque
  border: 1px solid
footer
  p
    font-size: 0.8 em
```

SCSS (Sassy CSS)

sintaxis introducida en la versión 3, y que actualmente se considera el estándar.

para acercarse más a la sintaxis habitual de CSS, conserva el uso de ; y {}
no basándose en la indentación o en espacios en blanco para funcionar correctamente.



body

```
{ 
  width: 80%; 
  margin: auto; 

  article { 
    width: 50%; 
    background-color: bisque; 
  } 

  footer { 
    p { 
      font-size: 0.8 em; 
    } 
  } 
}
```

Instalación

domingo, 14 de mayo de 2017 19:41

Como parte de alguna de las siguientes aplicaciones

- [Koala](#) (*Open Source*)
- [Scout](#) (*Open Source*)
- [LiveReload](#) (*Paid, Open Source*)
- [Compass.app](#) (*Paid, Open Source*)
- [Ghostlab](#) (*Paid*)
- [Prepros](#) (*Paid*)

Independientemente,
desde el terminal

Ha existido diversas opciones

- la implementación original en Ruby
- Las realizadas posteriormente en JS:
 - sass-node
 - sass (la nueva implementación oficial)

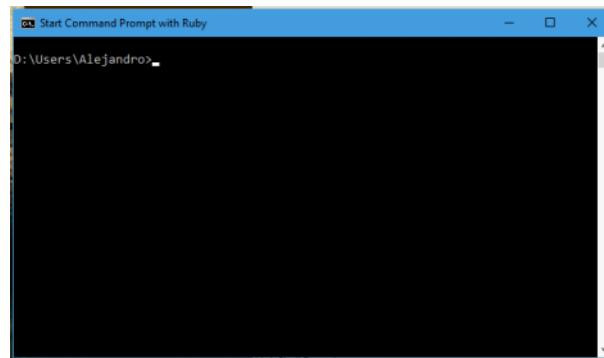
Implementación en Ruby

jueves, 7 de junio de 2018 22:26

En Windows, previamente es necesario instalar Ruby, e.g. mediante [Ruby Installer](#), que incluye una *command line PowerShell application*, como herramienta para utilizar las librerías de Ruby.

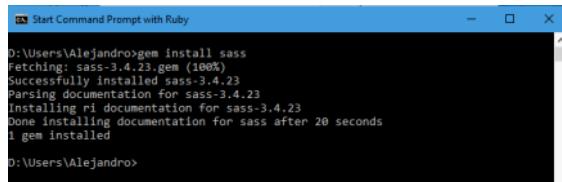


The RubyInstaller for Windows website features a logo with four colored squares (red, green, blue, yellow) followed by the text "RubyInstaller for Windows". Below the logo, there's a sub-headline: "The easy way to install Ruby on Windows". A descriptive text explains it's a self-contained Windows-based installer including the Ruby language, execution environment, documentation, and more. There are two main buttons: a red "Download" button and a blue "Add-ons" button. To the right, there's an illustration of a wooden chest with a red arrow pointing down into it, symbolizing installation.

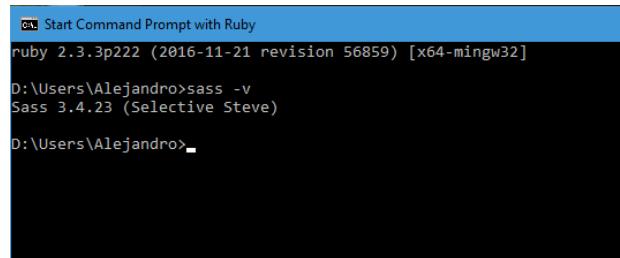


Desde la terminal se utiliza gem, el gestor de paquetes de Ruby:

```
gem install sass
```



Podemos comprobar la instalación consultando la versión (-v) de sass o su ayuda (-h)



Finalmente, se puede añadir la extensión correspondiente en el editor de código

A screenshot of the Visual Studio Code Marketplace. The search bar shows "Sass". The result is "Sass" by "robinbentley.sass-indented". The extension has 149086 installs and a 5-star rating. The description says: "Indented Sass syntax highlighting, autocomplete & snippets". There are buttons for "Deshabilitar" and "Desinstalación". Below the extension card, there's a section for "Indented Sass syntax highlighting, autocomplete & snippets for VSCode". A note at the bottom states: "VSCode only has built in support for SCSS syntax so I have put this together for people who use the indented syntax."

Uso desde el terminal de Ruby

domingo, 14 de mayo de 2017 20:55

sass inicio.scss : salida por terminal

```
D:\Desarrollo\Front_End_alce65\HTML5_Basic\06-metalenguajes\sass>sass inicio.scss
body {
  width: 80%;
  margin: auto; }
body article {
  width: 50%;
  background-color: bisque; }
```

sass inicio.scss inicio_scss.css : salida en un archivo en modo indentado ()como el terminal
sass inicio.scss inicio_scss.css --no-cache -t compact : salida con una línea por cada selector
sass inicio.scss inicio_scss.css --no-cache -t compressed : salida en una única línea
sass inicio.scss inicio_scss.css --no-cache -t expanded salida típica de css

```
body {
  width: 80%;
  margin: auto;
}

body article {
  width: 50%;
  background-color: bisque;
}

/*# sourceMappingURL=inicio_scss.css.map */
```

Automatización

miércoles, 17 de mayo de 2017 11:42

El comando --watch queda a la escucha para compilar automáticamente cualquier cambio que se realice en el fichero origen

```
$>sass --watch inicio.scss:inicio_scss.css  
>>> Sass is watching for changes. Press Ctrl-C to stop.
```

Modifico el fichero sass



El compilador detecta y procesa los cambios

```
body {  
    width: 80%;  
    margin: auto;  
  
    article {  
        width: 50%;  
        background-color: bisque;  
    }  
    footer {  
        p{  
            font-size: 0.8 em;  
        }  
    }  
}
```

```
>>> Change detected to: inicio.scss  
      write inicio_scss.css  
      write inicio_scss.css.map  
>>> Change detected to: inicio.scss  
      write inicio_scss.css  
      write inicio_scss.css.map
```



```
body {  
    width: 80%;  
    margin: auto; }  
  
body article {  
    width: 50%;  
    background-color: bisque; }  
  
body footer p {  
    font-size: 0.8 em; }  
  
/*# sourceMappingURL=inicio_scss.css.map */
```

Lo habitual es definir una carpeta SASS y otra CSS para ejecutar el watch en modo "carpetas"

```
$>sass --watch sass:css  
>>> Sass is watching for changes. Press Ctrl-C to stop.
```

Detección de errores

miércoles, 17 de mayo de 2017 11:44

Se produce en tiempo de compilación,
con la información adecuada en consola

```
wrote: css/variables.css.map
>>> Change detected to: scss/variables.scss
      error scss/anidados.scss (Line 20: Invalid CSS after "...": expected "{", was "} // Fin de los...")
      error scss/variables.scss (Line 28: Undefined variable: "$fondobase".)
```

En función del navegador, este puede reflejar los errores



The screenshot shows a browser's developer tools console with the URL `desarrollo.cyborg8.loc/Front_End_alce65/HTML5_Ba`. The error message is:

```
Error: Undefined variable: "$fondobase".
      on line 28 of scss/variables.scss
```

The code listed is:

```
23:
24:   * {
25:     box-sizing: border-box;
26:   }
27:   body {
28:     background-color: $fondobase;
29:   }
30:
31:   header {
32:     color: $texto_resaltado;
33:     width: $full_columns;
```

SASS

Pre-compilador de CSS

Loreum ipsum dolor sit amet, consectetur adipisicing elit. Vero porro pariatur impedit amet sint aspernatur laborum harum quia culpa dicta deserunt ex suscipit recusandae fugit error autem voluptatum, quasi. Nobis.

node-sass

viernes, 30 de marzo de 2018 16:44

<https://github.com/sass/node-sass>

Proporciona a Node.js (y otros JS) enlace (*binding*) a la librería *LibSass*, la versión en C del preprocesador de CSS

node-sass

Supported Node.js versions vary by release, please consult the [releases page](#)



Instalación

```
npm install -D node-sass
```

Al margen de Node.js, puede utilizarse como un *script* de npm

```
"sass": "node-sass -w -o ./src/css ./src/scss",
```

A diagram illustrating the command structure. Two red arrows point from the words 'watch' and 'output' to their respective options in the npm script command. The 'watch' arrow points to '-w', and the 'output' arrow points to '-o'.

sass (nueva implementación)

jueves, 7 de junio de 2018 22:29

La web oficial de SASS indica como método de instalación

```
npm install -g sass
```

El resultado de sass --versión nos indica la versión instalada

1.5.1 compiled with dart2js 2.0.0-dev.59.0

Su uso admite 2 formatos

```
sass origen.scss destino.css
```

```
sass origen.scss:destino.css  
sass carpeta_origen:carpeta_destino
```

Este sería el formato para
seleccionar las carpetas de
origen y destino

Por el momento no existe una opción que permita un modo watch,
aunque es fácil conseguirlo utilizando nodemon

<https://nodemon.io/>

```
"scripts": {  
  ...  
  "sass": "sass scss:css",  
  "watch:sass": "nodemon -e scss -x \"npm run sass\""  
},
```

extensión
monitorizada

programa ejecutado
en consecuencia

Nodemon is a utility that will monitor for any
changes in your source and automatically
restart your server or execute other programs.

npm install -g nodemon

Import



SASS permite importar ficheros SASS o CSS, con un comportamiento diferente

`@import 'fichero.scss', ...`
(aplicado a ficheros sin extensión)

el código CSS correspondiente al
fichero se incorpora en el punto
indicado en el CSS resultante

`@import 'fichero.css', ...`

la directiva @import correspondiente
se incorpora como primera línea del
CSS resultante

Importación desde selectores

`.class1 {
 @import 'fichero'
}`

El contenido del import se
contextualiza, creándose dentro
del selector que lo importa

Import: _partials

Sass

`@import '_fichero.scss', ...`
(aplicado a ficheros sin extensión)

Ficheros exclusivamente
para importación

- Se nombran siempre comenzando por _
- No se compilan a css
- Se importan desde otro fichero sass por referencia a su nombre, sin necesidad de incluir el guion bajo

Comentarios

Sass

SASS admite los 2 formatos de comentario (de 1 línea y mixtilínea) posibles en JS

```
// Comentario de 1 línea  
  
/* Comentario de varias líneas  
Idéntico que los utilizados en CSS  
*/
```

Los primeros, exclusivos de SASS, no se trasladan al fichero CSS compilado

Anidamiento

miércoles, 17 de mayo de 2017 11:45

Anidamiento

Sass

```
article {  
    div {  
        width 80%;  
    }  
}  
  
aside {  
    > div {  
        width: 50%;  
    }  
}
```

En **.SCSS** conserva el uso de ; y {}

En el segundo caso serían únicamente los div descendientes directos los que se verían afectados

La sintaxis **.SASS** no utiliza ; ni {}
El anidamiento depende del indentado

Variables

miércoles, 17 de mayo de 2017 11:47

Variables

Sass

\$<nombre>: valor

```
$ texto_resaltado : #5B83AD;
$ fondo_oscuro : #B43104;

header { color: $texto_resaltado;
          background-color: $fondo_oscuro;
        }
```

SASS.

Pre-compilación de CSS

Interpolación:
#{variable}

```
/*Variables*/
$div-name: test-img;
$texto_resaltado: #5B83AD;
$fondo_oscuro: rgb(180,49,4);
$full_columns: 80%;

/* Operaciones */
$fondo_suave: lighten($fondo_base, 10%);
$texto_normal: $texto_resaltado + #11b;

// Variable con un valor por defecto
$div-name: otro-test-img !default;

// Interpolación de variables: #{}
##{$div-name} {
```

Operadores y Funciones

Sass

Operaciones → + - * /

```
$light-blue: $nice-blue + $311;
```

```
p {color: $light-blue; }
```

Parrafo sin ningún estilo

Entre las funciones del lenguaje, podemos ver las que manipulan los colores

Funciones
predefinidas
Manejo del color

- **darken()** / **lighten()**: añaden blanco o negro
- **saturate()** / **desaturate()**: añaden colorido (saturación) o lo reducen haciendo los colores más grises,
- **fadein()** / **fadeout()**, incrementa o reduce la transparencia,
- **spin()** modifica el matiz (hue) del color.

<https://styde.net/lista-completa-de-funciones-nativas-de-sass/>

Ejemplo: Operadores y Funciones

SASS

Pre-compilador de CSS

```
 Lorem ipsum dolor sit amet, consectetur adipisciing elit. Vero porro pariatur impedit  
amet sint aspernatur laborum harum quia c  
fugit error autem voluptatum, quasi Nobis
```

/*Variables*/

```
$fondo_base: coral;  
$texto_resaltado: #5B83AD;  
$fondo_oscuro: rgb(180,49,4);  
$full_columns: 80%;
```

/* Operaciones */

```
$fondo_suave: lighten($fondo_base, 10%);  
$texto_normal: $texto_resaltado + #11b;
```

```
Natus modi repellat molestias minus, possa  
placet nisto ipsa temporibus quas necessum  
omnis at voluptates, libero neque, quae, ten
```

```
Voluptatem non qui illo. Architecto similie  
accusantium numquam, dolores, magnam, a  
similique reiciendis nulla mollitia esse, fac
```

```
Praesentium ex, quibusdam dolorem autem  
perspicans quia. Ullam cupiditate repellent  
corrupti odio illo autem repudiandae. Sit et
```

```
Aut dolore tempore vel facilis incident itaq  
nisto tenetur in consequatur molestiae duca  
numquam! Solita facilis voluptatem provis
```

Combinación de selectores (1)

Sass

Extend

La pseudoclase de SASS `extend` permite incorporar un selector al conjunto de los estilos previamente definidos en otro.

crea combinaciones de selectores mediante "coma" para aplicarles un conjunto de reglas, en lugar de duplicar esas reglas

Aplica al selector y a todas sus variantes

```
.selectorA {  
    color: red;  
}  
  
.selectorB {  
    @extend .selectorB;  
    font-size: 24px;  
}
```

SASS: extend - Ejemplos

En la clase aplicada al div se define el color de texto rojo y el subrayado en respuesta al ratón. El color de fondo se a definido genéricamente en los divs del grid.

En la clase aplicada al div se define el tamaño de fuente 24 y el color salmon.

La clase aplicada al div es la combinación de las dos anteriores, incluyendo el hoover de la primera y añadiendo color de letra blanco y borde

La clase aplicada al div es la combinación de la primera, en su forma hoover y de la segunda, añadiendo color de letra blanco y borde

**La pseudoclase
*extend***

Mixines como funciones

Sass

Definición

```
@mixin <nombre> (@parámetro: valor) {  
    propiedades Less  
}
```

La función se define pudiendo indicarle parámetros y darles un valor por defecto

Ejecución

(dentro de cualquier otra clase)
@Include <nombre> (valor)

Al invocar la función se puede modificar el valor de los parámetros

Mixines como funciones

Definición

```
@ mixin <nombre> (@parámetro: valor) {  
    propiedades Less  
}
```

\$parámetro

La función se define pudiendo indicarle parámetros y darles un valor por defecto

Ejecución

(dentro de cualquier otra clase)
@Include <nombre> (valor)

Al invocar la función se puede modificar el valor de los parámetros

SASS: Mixins - Ejemplos

```
@ mixin div_basico() {  
    @width: 50%;  
    @height: 100px;  
    border: 2px solid @nice-blue;  
    color: @rojo-ladrillo;  
    padding: 0.5em 1em;  
    margin: 1em auto;  
}  
  
@ mixin borde_redondeado(@radio:10px){  
    -webkit-border-radius: @radio;  
    -moz-border-radius: @radio;  
    border-radius: @radio;  
}
```

```
.caller2 {  
    @include div_basico();  
    @include borde_redondeado();  
}  
  
.caller3 {  
    @include div_basico();  
    @include borde_redondeado(30px);  
}
```

Otro Parrafo sin ningún estilo

Segundo ejemplo de la utilización de un mixin

Tercer ejemplo de la utilización de un mixin

Anidamiento + mixines

Sass

```
@mixin div_basico(@ancho:50%) {...}  
@mixin borde_redondeado(@radio:10px) {...}  
  
article {  
    div {  
        @include div_basico();  
        @include borde_redondeado()  
    }  
}  
  
aside {  
    > div {  
        @include div_basico(80%);  
        @include borde_redondeado(30px)  
    }  
}
```

Parrafo sin ningún estilo

Parrafo dentro de un div.
dentro del article

Otro Parrafo sin ningún estilo

Parrafo dentro de un div. dentro del aside

Estructuras de control: condicional

Sass

```
@if (<condición>) {  
    selector { reglas}  
} @elseif (<condición>) {  
} @else {}
```

El código definido dentro de la
estructura sólo será procesado para
dar CSS si se cumple la condición

Condiciones

operadores → \$variable == valor
<, >, <=, >=

funciones booleanas →
isnumber ispixel
isstring isem
iscolor ispercentage
iskeyword isunit
isurl isruleset

Estructuras de control: bucles e iteraciones

Sass

```
@for $i from 1 through n {  
    código repetido  
}
```

El código definido dentro de la estructura será procesado para dar CSS el número de veces indicados

```
@while <condición>{  
    selector { reglas}  
    Cambios en la variable usada en la condición  
}
```

```
@each $item in <lista> {  
    código $item  
}
```

En cada iteración, el valor de \$item corresponderá a los sucesivos valores de la lista. Igualmente en caso de mapas, los pares de valores se recogen en \$key, \$value

```
@each $key, $value in <mapa> {  
    código $key, $value  
}
```

Arquitectura CSS

miércoles, 17 de mayo de 2017 11:49

Preprocesadores -> facilidad de importación de archivos

Conjunto de reglas (convención) respecto a la organización de las reglas en distintos ficheros

SMACSS de Jonathan Snook <https://smacss.com/>

OOCSS de Nicole Sullivan

BEM (Block, Element, Modifier):) de la empresa Yandex

una nomenclatura más estructurada del HTML y por tanto, del CSS

<http://getbem.com/>

ITCSS (Inverted Triangle CSS) de Harry Roberts <https://csswizardry.com/>
una manera de organizar las CSS en función de su especificidad

BEMIT es la combinación de la nomenclatura BEM con la jerarquía de ITCSS
la combinación de la nomenclatura BEM con la jerarquía de ITCSS.

Especificidad CSS

miércoles, 17 de mayo de 2017 12:10



La especificidad tiene cuatro componentes; por ejemplo a, b, c y d. El componente "a" es el más distintivo y el "d", el que menos.

- El componente "a" es bastante sencillo: es 1 para una declaración en un atributo style; si no, es 0.
- El componente "b" es el número de selectores de id en el selector (los que empiezan con #).
- El componente "c" es el número de selectores de atributo, incluidos los selectores de clase y pseudoclases.
- El componente "d" es el número de tipo de elementos y pseudoelementos del selector.

Selector	a	b	c	d	Especificidad
h1	0	0	0	1	0,0,0,1
.foo	0	0	1	0	0,0,1,0
#bar	0	1	0	0	0,1,0,0
html >head+body ul#nav *.home a:link	0	1	2	5	0,1,2,5

Para fanático de La Guerra de las Galaxias, Andy Clarke creó un gráfico explicativo con personajes del Imperio para ilustrar la especificidad de CSS

<https://stuffandnonsense.co.uk/archives/images/specificationwars-05v2.jpg>

Ejemplos:

```
p{ ... }          /* 0,0,0,1 */
body p{ ... }    /* 0,0,0,2 */
.intro{ ... }     /* 0,0,1,0 */
.p.intro{ ... }   /* 0,0,1,1 */
#contenido{ ... } /* 0,1,0,0 */
#contenido p{ ... } /* 0,1,0,1 */
div#contenido p{ ... } /* 0,1,0,2 */
div#contenido p.intro{ ... } /* 0,1,1,2 */
```

```
body div#contenido p.intro{ ... }      /* 0,1,1,3 */
body div#contenido p.intro:hover{ ... }  /* 0,1,2,3 */
body div#contenido p.intro:first-line{ ... } /* 0,1,1,4 */
```

ITCSS (CSS de Triángulo Invertido)



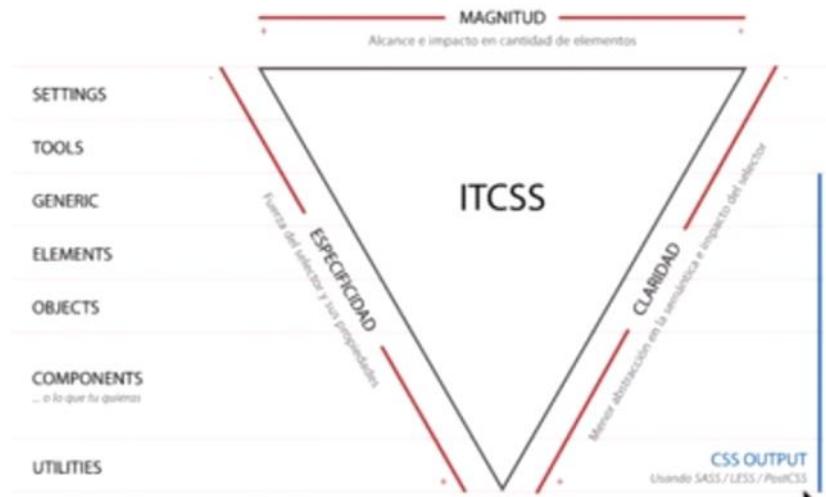
Harry Roberts <https://csswizardry.com/>

inuit.css es un framework open source que sigue la convención de nombres BEM y los principios OOCSS desarrollado Harry Roberts, un joven desarrollador web y arquitecto front-end. Es un framework modular que consiste en una serie de objetos y abstracciones, no de elementos de diseño como Bootstrap. "Generic" incluye normalize.css, el clearfix, etc. "Base" son reglas básicas para títulos, tablas y otros elementos. Finalmente "Objects" es la parte más interesante, con los objetos CSS con código profusamente comentado. inuit está desarrollado en Sass, podés modificar las variables y extenderlo con tu propia UI. <http://inuitcss.com/>



- Una arquitectura sana, escalable y manejable
- Una filosofía, no una librería
- Un meta framework, un frameworks para frameworks
- Increíblemente simple

Carpetas con los sucesivos elementos del css



1. **Settings** – Se usa para las variables globales, elementos de configuración (Si usamos pre procesador, si no, podemos saltarnos esta capa)
2. **Tools** – Mixins y Funciones (solo si usamos preprocesadores)

3. Generic – Estilos CSS: reset, box-sizing, normalize.css...
4. Base – Elementos HTML sin clases, selectores.
5. Objects – Patrones de diseño, la estructura. Solo si usamos CSS dirigido a objetos (OOCSS)
6. Components – Carruseles, calendarios... y sets de UI... Aquí metemos nuestros estilos concretos, para una lista concreta de un producto, por ejemplo.
7. Trumps – Ayudas y lo que usamos para sobreescribir. La capa de los importantes, la más específica de todas. Aquí habrá código muy concreto, para partes del DOM muy específicas.

<http://creativity.babel.es/creativity/itcss-primer-paso-hacia-bemit/>

Conferencia de Harry Roberts

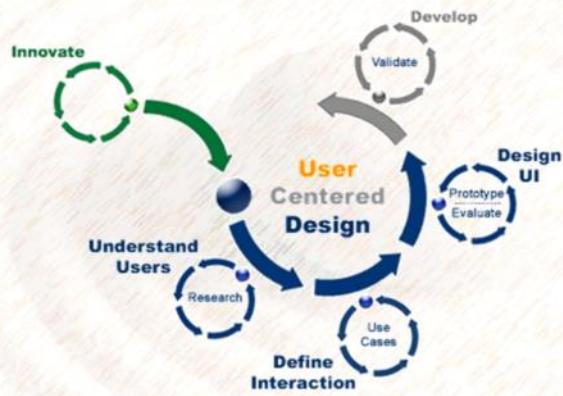
<http://csswizardry.net/talks/2014/11/itcss-dafed.pdf>

Diseño Web Centrado en el Usuario

Orientado a los usuarios del producto, para obtener productos usables y satisfactorios para ellos.



- Se aplica durante todas las fases del desarrollo, desde las primeras etapas:
 - planificación
 - diseño
 - desarrollo,
 - evaluación
- Los usuarios participan durante todo el proceso.
- Es iterativo.
- Es multidisciplinar.



Aproximaciones a la UX en la Web

La dualidad inicial entre sitios estáticos que acumulan información en forma de documentos y sitios dinámicos orientados a bases de datos

dos concepciones diferentes de la experiencia de usuario

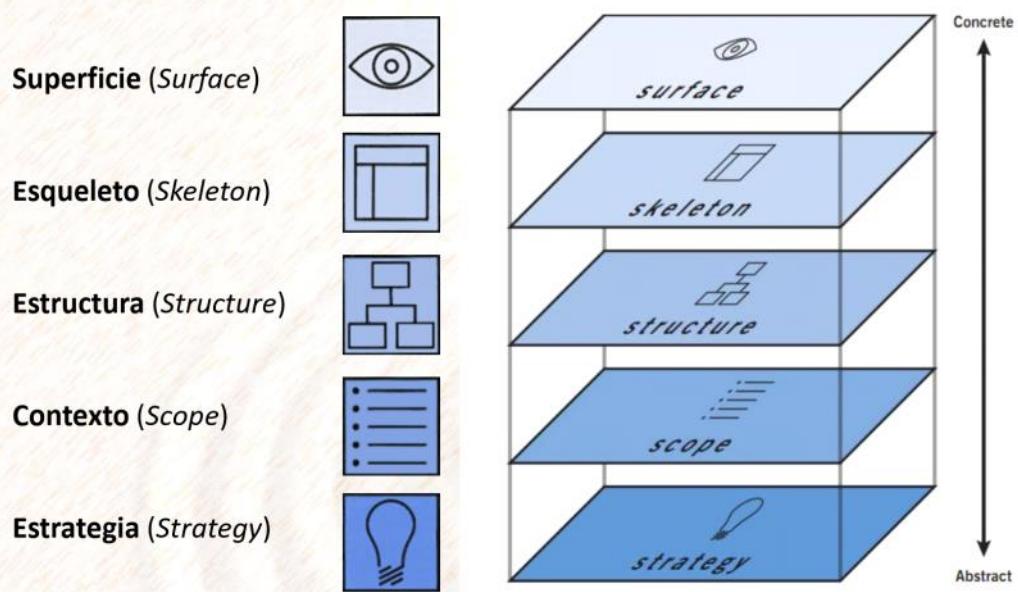
- como un problema de diseño de aplicaciones e interfaces
- como una cuestión de recuperación y distribución de la información



Jesse James Garrett

The Elements of User Experience User-Centered Design for the Web and Beyond (2nd ed.). Jesse James Garrett. New Riders, 2011.

Planos de la UX



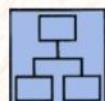
Planos de la UX



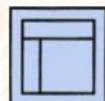
Estrategia: Define los objetivos del sitio & usuarios, hace investigación (entrevistas, grupos focales), crea personas.



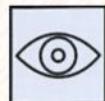
Alcance o Contexto: Decide las características funcionales y de contenido, crea las historias de usuario y escenarios de uso.



Estructura: Desarrolla la estructura del sitio, arquitectura de información, usa el “Card Sorting” y las historias de usuario. Crea el mapa del sitio.



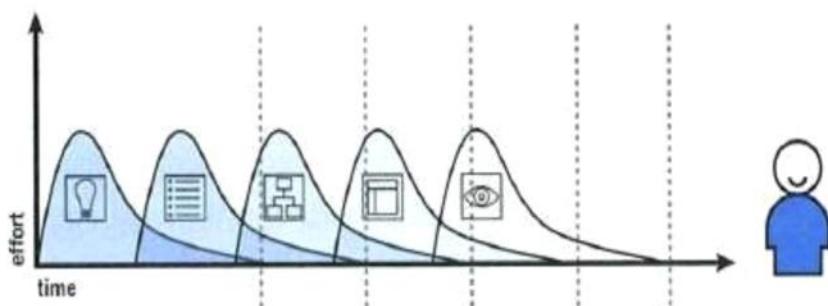
Esqueleto: Prototipos o “wireframes”, navegación, ubicación de fotografías, texto, botones, etc.



Superficie: Diseño visual, concepto, colores, tablas de ánimo (mood boards), etc.

Planos de la UX en el tiempo

Los planos no se suceden uno a continuación del otro...

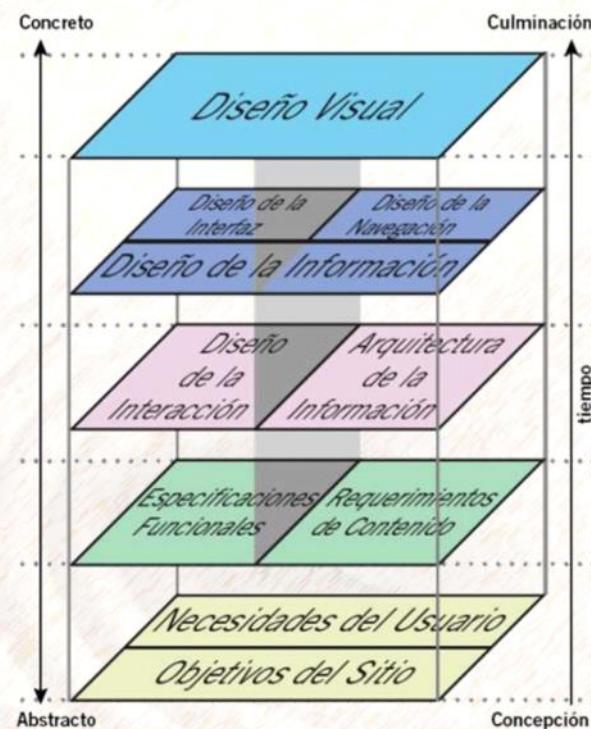


... El trabajo en el siguiente plano debe comenzar antes de que el trabajo en el plano anterior ha concluido.

Elementos de la UX

Los elementos de la experiencia de usuario en su doble vertiente

- diseño de aplicaciones e interfaces
- recuperación y distribución de la información

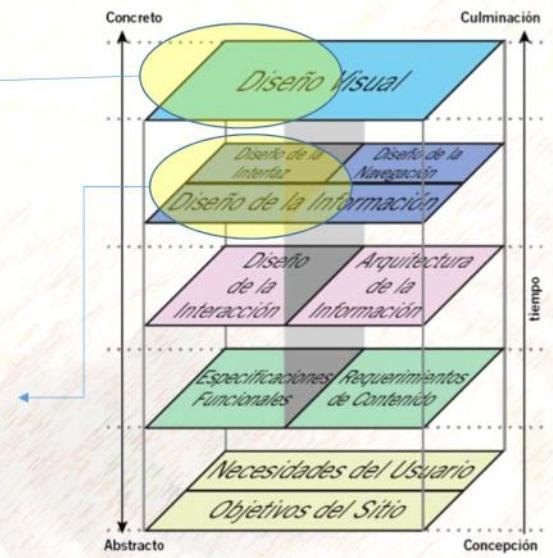


Web como interfaz de software (1)

Diseño Visual: tratamiento gráfico de los elementos de la interfaz (el "look" del "look & feel")

Diseño de la Interfaz: como en el Estudio de Interacción Humano-Computador tradicional: diseño de los elementos de la interfaz para facilitar la interacción del usuario con la funcionalidad

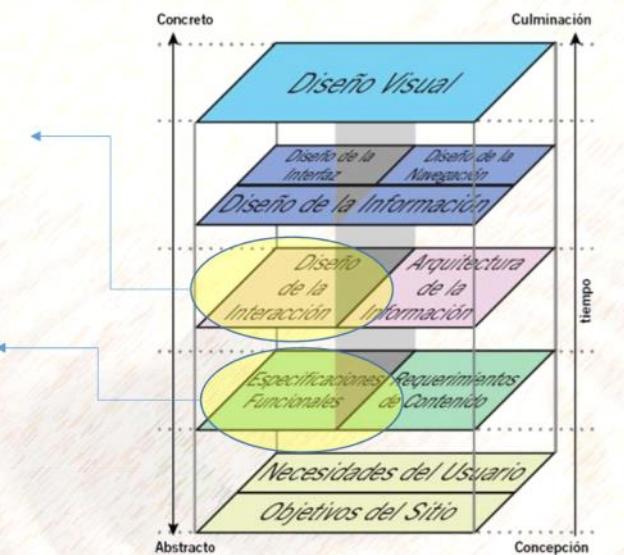
Diseño de la Información: en el sentido de Tufte, diseño de la presentación de la información para facilitar el entendimiento



Web como interfaz de software (2)

Diseño de la Interacción: desarrollo del flujo de las aplicaciones para facilitar las tareas del usuario, definiendo cómo el usuario interactúa con la funcionalidad del sitio

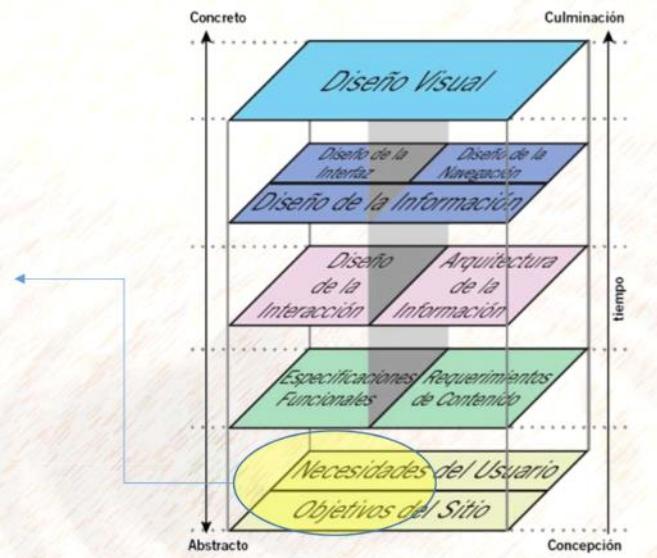
Especificaciones Funcionales: "set de herramientas": descripciones detalladas de las funcionalidades que el sitio debe incluir para satisfacer las necesidades del usuario



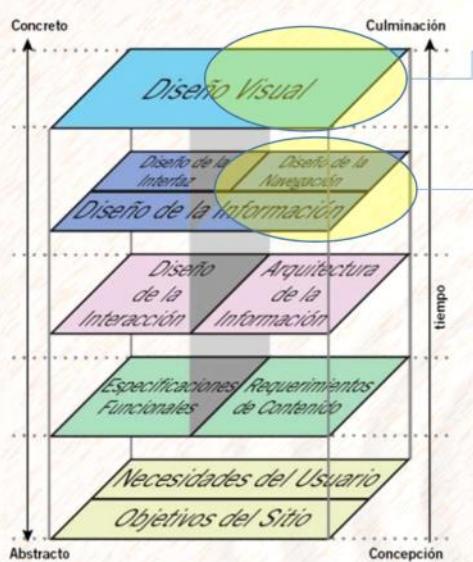
Web como interfaz de software (3)

Necesidades de Usuario: objetivos para el sitio externamente derivados, identificadas a través de la investigación de los usuarios, etno/tecnos/psicográficos, etc.

Objetivos del Sitio: de negocios, creativos, u otros generados de manera interna para el sitio



Web como sistema de hipertexto (1)

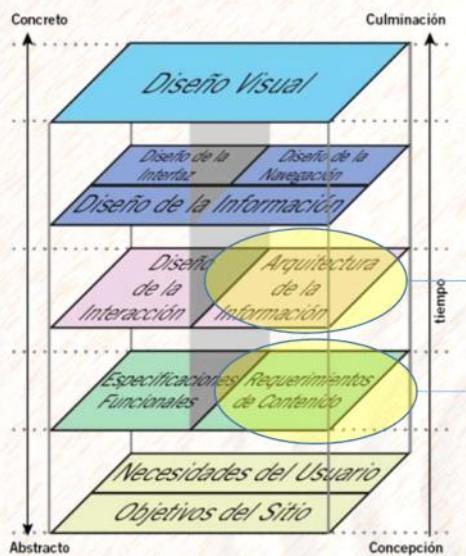


Diseño Visual: tratamiento visual de los elementos de texto y gráficos en la página y componentes de navegación

Diseño de la Navegación: diseño de elementos de interfaz para facilitar el movimiento de los usuarios a través de la arquitectura de la información.

Diseño de la Información: nuevamente, en el sentido de Tufte, diseño de la presentación de la información para facilitar el entendimiento

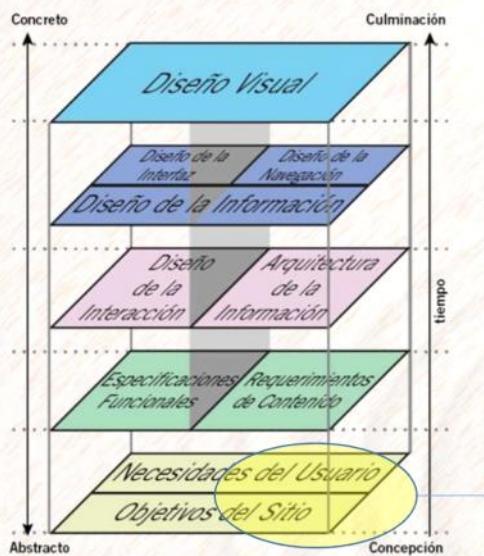
Web como sistema de hipertexto (2)



Arquitectura de la Información: el diseño estructural del espacio de información para facilitar el acceso intuitivo al contenido

Requerimientos de Contenido: definición de los elementos de contenido requeridos para satisfacer las necesidades de usuario

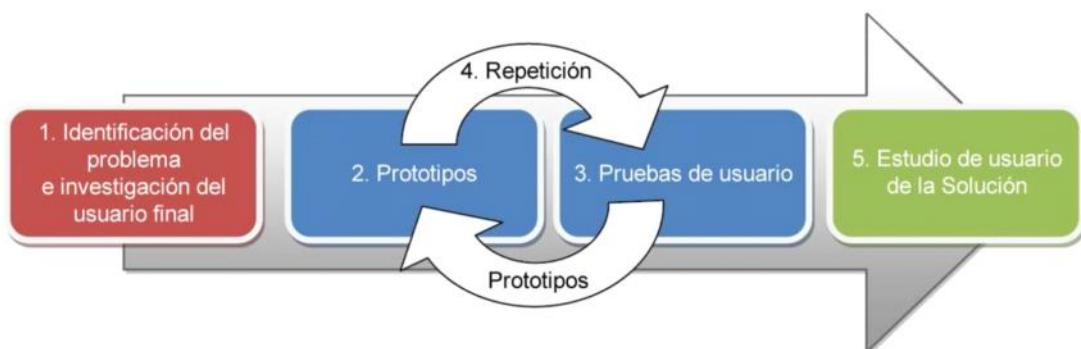
Web como sistema de hipertexto (2)



Necesidades de Usuario: objetivos para el sitio externamente derivados, identificadas a través de la investigación de los usuarios, etno/tecnos/psicográficos, etc.

Objetivos del Sitio: de negocios, creativos, u otros generados de manera interna para el sitio

DCU Flujo de trabajo



El desarrollo y la implicación de los usuarios son efectuados de una manera **iterativa**:

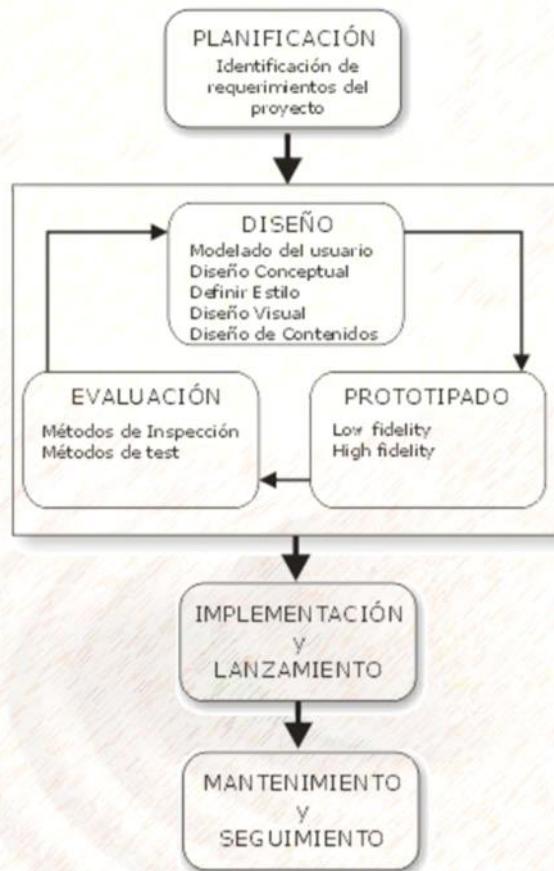
el ciclo se repite hasta que los objetivos de usabilidad del proyecto hayan sido logrados.

- 1) *Concepto*
- 2) *Investigación*
- 3) *Prototipo*
- 4) *Pruebas/Construcción hasta que esté correcto,*
- 5) *Lanzamiento,*
- 6) *Pruebas Post Lanzamiento.*

Fases del diseño centrado en el usuario

Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.

[Hipertext.net](#) (2004)
Universitat Pompeu Fabra



Otros Enfoques del Diseño Web

Diseño orientado al usuario



Diseño orientado a objetivos

Es una perspectiva más comercial, ligada objetivos de negocio, que aparece principalmente en sitios corporativos, portales y tiendas electrónicas

Diseño orientado a la implementación

Se basa en el desarrollo de software a medida, como un servicio orientado a la implementación de aplicaciones informáticas para los requerimientos específicos de un cliente



Diseño web. Recursos

Planteamientos previos

- Diferencias entre diseño orientado a presentación e impresión
- La necesidad de comprobar la compatibilidad con distintos navegadores.

The screenshot shows the W3C website's navigation bar with options for desktop, mobile, and print views. Below the bar are links for STANDARDS, PARTICIPATE, MEMBERSHIP, and ABOUT W3C. The main content area shows the breadcrumb path: W3C > Standards > Web Design and Applications. A large blue header box contains the text 'WEB DESIGN AND APPLICATIONS'. Below it, a sub-header says 'On this page → technology topics • news • upcoming events and talks'. On the left sidebar, under the 'STANDARDS' heading, there is a link to 'Web Design and Applications'. At the bottom of the sidebar, there is a section titled 'Algunos recursos' with links to various websites: 'W3C - Web Design And Applications', 'Cool Homepages', 'Web Pages That Suck', 'Web Page Design for Designers', and 'Web Design and Development'.

Views: desktop mobile print

STANDARDS PARTICIPATE MEMBERSHIP ABOUT W3C

W3C » Standards » Web Design and Applications

WEB DESIGN AND APPLICATIONS

On this page → technology topics • news • upcoming events and talks

Algunos recursos

[W3C - Web Design And Applications](#) [Web Page Design for Designers](#)

[Cool Homepages](#) [Web Pages That Suck](#) [Web Design and Development](#)

Mejora progresiva

Mejora progresiva
(*Progressive enhancement*)
Diseño web adaptativo
(*Adaptive Web Design*, AWD)

- Estructura en tres capas
 - Marcado semántico
 - Estilos y scripts independientes
- Usabilidad
- Accesibilidad

CHAPTER 1:
**THINK OF THE USER,
NOT THE BROWSER**



Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement.
Aaron Gustafson. Easy Readers, 2011.

Planificación de Proyectos

Gestión de proyectos. Alojamiento Web.

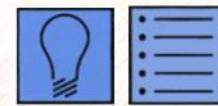
Arquitectura de la Información. Metodología y
documentación: Mapas del sitio y esquemas.

Planificación



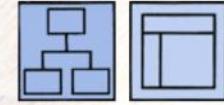
Corresponde principalmente a las primeras de las tareas descritas

- Gestión de proyectos (*Project management*)



planos estrategia y alcance

- Arquitectura de la información (*Information architecture*)



planos estructura y esqueleto

AC6

Gestión de proyectos: objetivos

- Evaluación de las **metas y objetivos** del proyecto

En proyectos complejos, los objetivos se recogen formalmente en una **Solicitud de Propuesta** (Request for Proposal, RFP)

En cualquier caso hay una serie de objetivos generales del sitio web (¿Qué hará el sitio?)

- Vender productos / servicios
- Aumentar la conciencia pública de una empresa / producto / servicio
- Adquirir empleados
- Entretenir
- Educar
- Comunicarse con los clientes
- Difundir información
- Proporcionar actualizaciones de productos / servicios

Condicionantes del proyecto

En cualquier caso, junto a sus objetivos, hay una serie de condicionantes y circunstancias iniciales básicos para definir un proyecto

- Explicación del **entorno de alojamiento** (¿Dónde "vivirá" el sitio?)
 - **Contenido, componentes y funcionalidades** requeridos (¿Cómo funcionará el sitio?)
 - Descripción de la **audiencia objetivo** (¿Quién utilizará el sitio?)
 - **Plazo de actividad** propuesto (¿Cuánto tiempo permanecerá activo el sitio?)
- La empresa
- Su perfil y su trayectoria; sus fortalezas y debilidades; sus competidores.
 - La persona de contacto: su nivel de experiencia con la Web y en proyectos similares
 - La marca de la compañía: su logo y su imagen corporativa, ¿existen, se necesita rediseñarlos o hay que crearlos?

Gestión de proyectos: audiencia

- Identificación del **público objetivo** del sitio web

Aspectos técnicos
de los destinatarios

- plataformas a través del cual se accede al sitio (Windows, Mac, Linux, Móviles...)
- software a través del cual se accede al sitio (navegadores)
- área de pantalla / resoluciones: la "ventana" a través del cual se ve el sitio

Aspectos demográficos y de
mercado de los destinatarios

- edad
- género
- estudios
- situación económica
- cultura

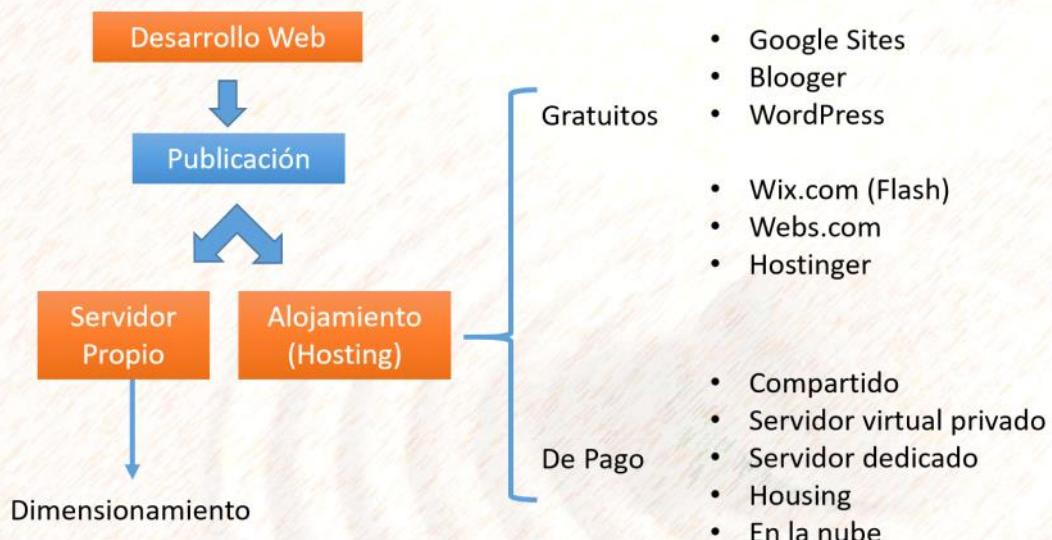
Gestión de proyectos: alojamiento

Importante condicionante desde los planteamientos iniciales del proyecto

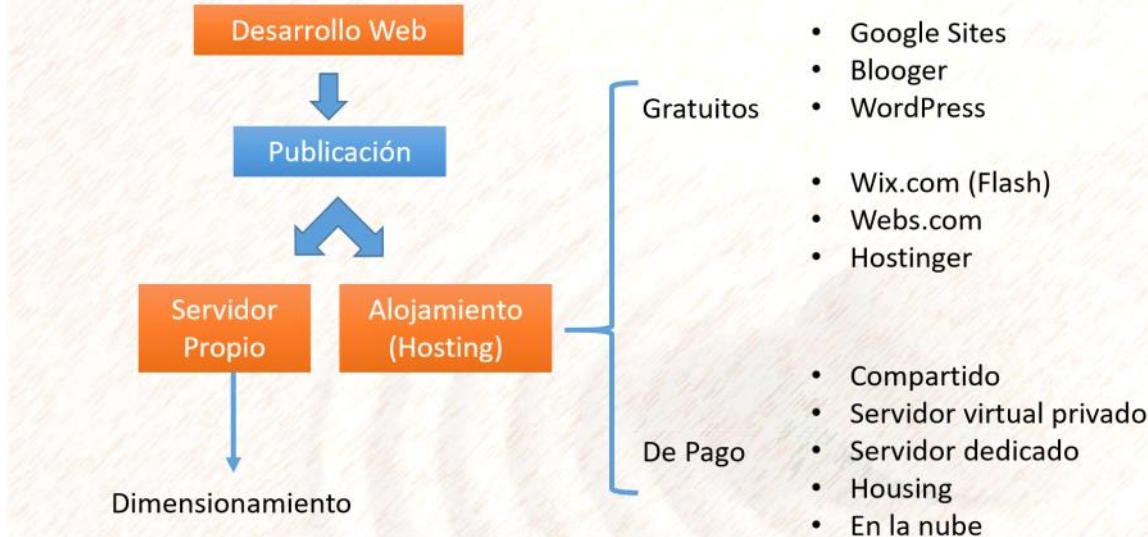
Diseño orientado a la implementación



Publicación y Escalabilidad



Publicación y Escalabilidad



Publicación en la Web



Alojamiento Web gratuito

The screenshot shows the Hostinger website homepage. At the top, there's a navigation bar with links for 'Inicio', 'Hosting Web', 'Servidores', 'Afiliados', 'iPideyo Ya!', 'Foro', and 'Contacto'. There are also fields for 'Email' and 'Contraseña' with a 'Entrar' button, and links for 'Crear cuenta' and 'Olvide la contraseña'.

2GB de espacio (con un límite en el número de ficheros) y un tráfico máximo de 100 GB.

Soporte de PHP y MySQL sin restricciones.

Creador de sitios y Cpanel para la administración.

Hosting Web Gratis

Espacio: 2Gb. Tráfico: 100Gb.
PHP, MySQL, Creador de Sitios,
¡En Español y Sin Publicidades!

iPideyo ya!

Nueva generación de hosting gratis

Olvida las restricciones sobre el hosting gratuito. Hostinger es diferente. Proveemos un servicio fiable y completo con un fantástico soporte al usuario.

Soporte completo de PHP y MySQL

A diferencia de otros servidores gratuitos, tenemos PHP y MySQL sin ninguna restricción. Proveemos acceso total a la última versión de PHP y MySQL.

Sin publicidad en tus páginas

Sin links de texto. Sin incómodas pop-ups. Sin banners. Sin publicaciones obligatorias en foros. Tu sitio estará 100% libre de publicidades para siempre.

Instalador de Scripts y CMS

Con sólo unos pocos clicks puedes instalar WordPress, Joomla, Drupal, Prestashop y mucho más. Pueden configurarse sitios de aspecto profesional y completamente funcionales en cuestión de minutos. ¡Nunca ha sido tan fácil!

<http://www.hostinger.es/>

Alojamiento: Hostinger (1)

The screenshot shows a web interface for creating a new account. On the left, there is a sidebar with a message about a DDOS attack and a date (2013-12-04). The main area has a blue header with the text: "Para crear una cuenta nueva, elegimos un subdominio en los dominios ‘libres’ del proveedor o nuestro dominio propio, previamente registrado". Below this, there are fields for "Subdominio www.", "Contraseña", and "Contraseña de nuevo". There is also a radio button for "Subdominio" and a dropdown menu for domain endings like ".esy.es", ".url.ph", ".bl.ee", ".meximas.com", ".96.lt", ".besaba.com", and ".hol.es". A CAPTCHA code "ruk3m" is displayed with a text input field below it. A "Crear" button is at the bottom right.

Alojamiento: Hostinger (2)

The screenshot shows the Hostinger account dashboard. At the top, there's a navigation bar with links: Mi Perfil, Hosting, Servidores, Dominios, Facturas, Conocimiento, Noticias, Mesa de Ayuda, Afiliados, and Cerrar Sesión. Below the navigation is a table titled "Cuentas Activas" (Active Accounts) with columns: Dominio, Plan, Expira en, Estado, Notas, and Acciones. A single row is shown for the domain "boracay.hol.es" with a "Gratis" plan, no expiration date, and an "Activo" status. An "Administrar" button is in the Actions column. A blue callout box points to the "Crear nueva cuenta" (Create new account) button in the top left of the main content area, with the text "En la misma cuenta podemos tener varios dominios" (In the same account we can have several domains). Another blue callout box points to the "Acceso a CPanel" (Access to cPanel) button in the top right, with the text "Acceso a CPanel". Below the table is a section titled "Cuentas" (Accounts) with icons for Detalles, Para Empezar, Mejorar Cuenta, Cambiar Contraseña, Cambiar Tema, and Cambiar Idioma. To the right, there's a vertical sidebar with links: Emails, Dominios, Sitio web, Archivos, Avanzado, and Otros.

Cuentas Activas					
Dominio	Plan	Expira en	Estado	Notas	Acciones
boracay.hol.es	Gratis	-	Activo	-	Administrar

[Crear nueva cuenta](#) En la misma cuenta podemos tener varios dominios

Acceso a CPanel

Cuentas

Detalles Para Empezar Mejorar Cuenta Cambiar Contraseña

Cambiar Tema Cambiar Idioma

Emails
Dominios
Sitio web
Archivos
Avanzado
Otros

Dominios propios

Detalles	Detalles DNS (nameservers)
1.hostinger.es	31.170.163.241
ns2.hostinger.es	64.191.115.234
ns3.hostinger.es	173.192.183.247
ns4.hostinger.es	31.170.164.249
Puedes apuntar tu dominio (Registro DNS "A") a este IP: 31.170.165.20	

Detalles del Sitio	
Accede a tu sitio en	http://boracay.es
Accede a tu sitio con www.	http://www.boracay.es
Vista previa de tu sitio en	Vista Previa
Dirección IP del Sitio	31.170.165.20
Nombre del Servidor	server17.hostinger.es

Hemos creado una cuenta de un dominio sin registrar:
en principio no accedemos al sitio

Registro “A” en DNS: 31.170.165.20 – boracay.es
Si lo hacemos en hosts, localmente tendremos acceso al “dominio fantasma” boracay.es

Hostinger: Archivos y bases de datos

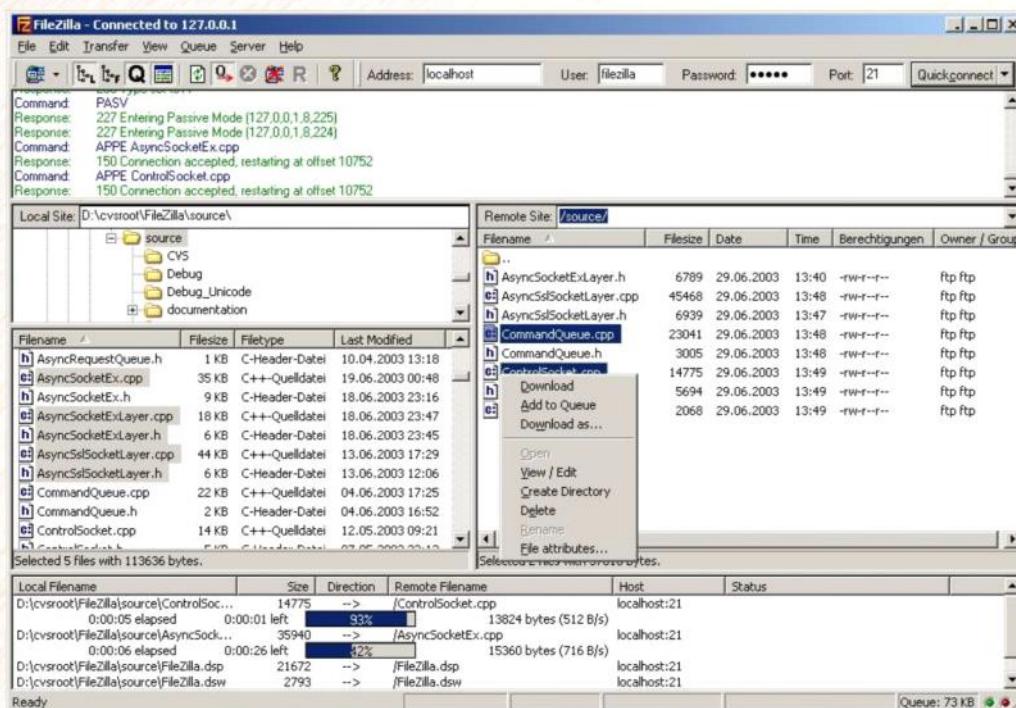




Ejercicio. Publicación en Hostinger

- Creamos una cuenta en hostinger
 - Añadimos un dominio fantasma y comprobamos como configurar el fichero host y el DNS para poder acceder.
Para el acceso DNS, utilizamos el cliente Windows 8 con las herramientas de administración remota del servidor.
 - Añadimos un subdominio y comprobamos su acceso sin necesidad de hacer nada más
 - Creamos un sitio web / publicamos una página web
 - Creamos una base de datos y comprobamos como administrarla con phpMyAdmin
- 
- 

Publicación en la Web. Clientes FTP



Publicación y Escalabilidad



Informática en la nube (Cloud computing)



- Infraestructuras externalizadas
- Auto-servicio por parte de los clientes
- Capacidad de medir el servicio
- Pago por uso
- Rápida elasticidad
- Distribución geográfica

Cloud Computing

Servicios en la nube

0. **Almacenamiento en la nube.** Sería un servicio tan básico que no suele considerarse dentro del modelo de *cloud computing*.
1. **SaaS (Software como Servicio):** se trata de cambiar las aplicaciones tradicionalmente instaladas en una infraestructura propia a versiones en la nube. E.g. MS Office 365, Google Apps
2. **IaaS (Infraestructura como Servicio):** es el hospedaje de aplicaciones existentes dentro de máquinas virtuales en la nube. Siguen siendo aplicaciones normales, ejecutadas en un sistema operativo común, sólo que se ejecutan dentro de máquinas virtuales en la nube. E.g. Amazon EC2 y RackSpace.
3. **PaaS (Plataforma como Servicio):** se trata de los servicios en la nube que permiten crear aplicaciones específicamente desarrolladas para trabajar en entornos "Cloud" utilizando una API de desarrollo específica cada una de las plataformas disponibles. E.g. *Windows Azure Platform* de Microsoft, *App Engine* de la *Google Cloud Platform* o *Force.com* de Salesforce.

Almacenamiento en la nube

Google Drive	https://www.google.com/drive/
OneDrive (Microsoft)	https://onedrive.live.com
Dropbox	https://www.dropbox.com/
Box	https://www.box.com/
SugarSync	https://www.sugarsync.com/
Wuala (LaCie)	https://www.wuala.com/
iCloud (Apple)	https://www.icloud.com/
UbuntuOne	https://login.ubuntu.com/

Google Drive: ligado a Gmail – 15 GB - GoogleDocs

OneDrive: Ligado a Hotmail – 15 GB – MS Office Online

Compartir en Google Drive

The screenshot shows the sharing settings for a folder named "Microforum". A context menu is open with options: "Compartir...", "Obtener enlace", "Mover a...", and "Destacar".

Configuración de uso compartido

Enlace para compartir
https://drive.google.com/folderview?id=0BxhwKd_ErgouRkJ0c0ZDVnBCb1E&usp=sharing

Comparte este enlace a través de:

Quién tiene acceso

Público en la Web: cualquier usuario de Internet puede encontrar los elementos y consultarlos.

Cambiar... → Es propietario

Uso compartido de enlaces

Si: público en la Web
Cualquier usuario de Internet puede encontrar el elemento y acceder a él sin necesidad de iniciar sesión.

Si: cualquier usuario con enlace
Cualquier usuario que disponga del enlace puede acceder al elemento sin necesidad de iniciar sesión.

No: ciertos usuarios
Compartido con ciertos usuarios.

Acceso: Cualquier usuario (no requiere acceso) Puede ver ▾

Nota: los elementos con cualquier opción de uso compartido de enlaces pueden publicarse igualmente en la Web. [Más información](#)

Compartimos la carpeta de forma pública desde las opciones avanzadas de compartir

Hosting en Google Drive

Cualquier usuario puede buscar y ver esta carpeta.

https://drive.google.com/open?id=0BxhwKd_6rgouRkJ0c0ZDVnBCb1E

Configuración para compartir

[https://drive.google.com/open?id=0BxhwKd_6rgouRkJ0c0ZDVnBCb1E
&authuser=0](https://drive.google.com/open?id=0BxhwKd_6rgouRkJ0c0ZDVnBCb1E&authuser=0)

Extraemos
ID de documento

Componemos nuestra URL

www.googledrive.com/host/0BxhwKd_6rgouRkJ0c0ZDVnBCb1E

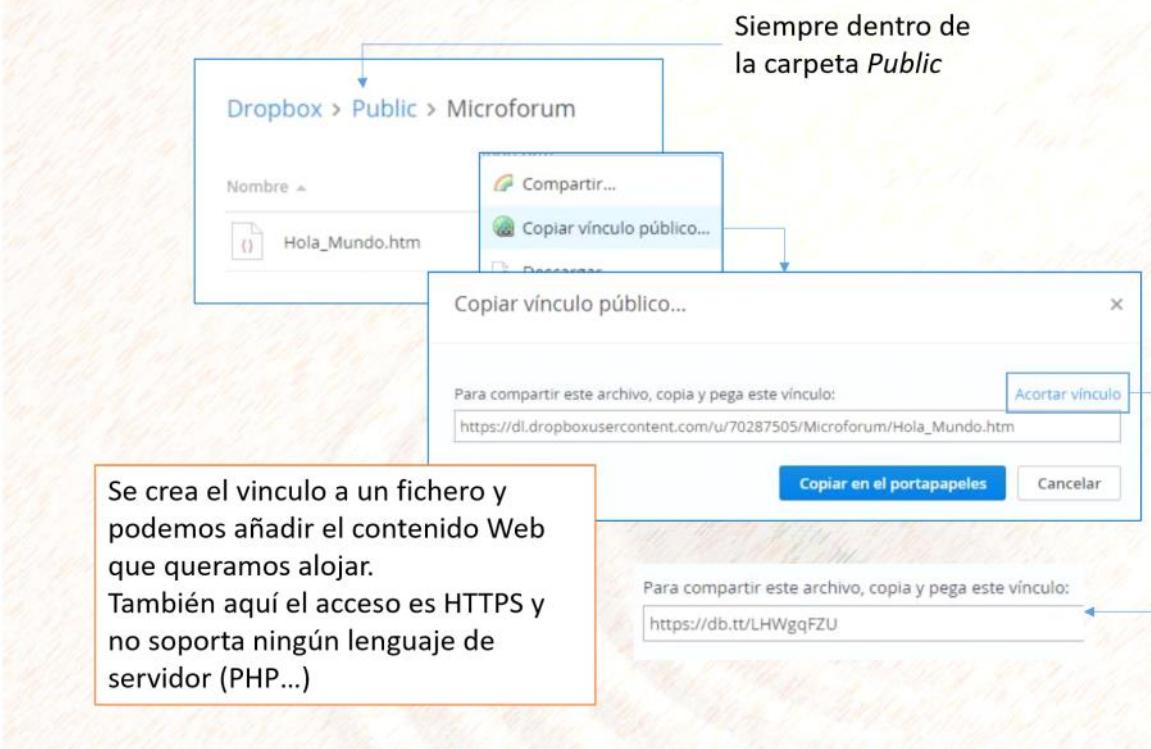


Microforum

TITLE

Podemos añadir el contenido Web
que queramos alojar. El acceso es
HTTPS y no soporta ningún
lenguaje de servidor (PHP...)

Hosting de Dropbox





Ejercicio (opcional). Publicación de contenidos

Creación de un sitio Web. Publicación de contenidos

- a) Creación de un sitio web en un proveedor de hosting (e.g. **Hostinger**)
- b) Publicamos el documento html creado anteriormente.
 - Hola Mundo; dentro de su propia carpeta “001_Hola_Mundo”

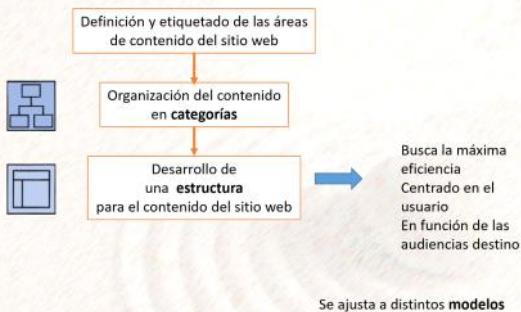
GDrive como sitio Web. Publicación de contenidos

- a) Creación en Google drive una carpeta compartida
- b) Publicamos el documento html creado anteriormente.
 - Hola Mundo; dentro de su propia carpeta “001_Hola_Mundo”

Objetivo: Tener los conocimientos mínimos sobre creación de sitios web y publicación en ellos de documentos HTML.

AC6

Arquitectura de la Información (AI)



Diapositiva 92

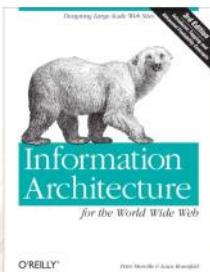
AC6 Web Design Demystified
Wendy Willard
McGraw-Hill, 2011
Alejandro Cerezo, 07/06/2015

AC9

Definición de la AI

1. El **diseño estructural** de espacios compartidos de información.
2. La **combinación** de organización, etiquetado, búsqueda y sistemas de navegación el sitios web e intranets.
3. El **arte y la ciencia** de la organización y etiquetado de sitios Web, intranets, comunidades en línea, y software, para promover la usabilidad y facilidad de encontrar información.
4. Una **comunidad emergente** de practicantes enfocados en traer los principios del diseño y la arquitectura al paisaje digital.

Lou Rosenfeld & Peter Morville



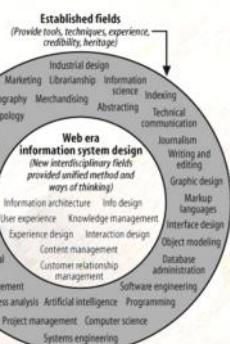
Diapositiva 93

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/06/2015

AC9

Disciplinas

- Diseño Gráfico
- Biblioteconomía (Bibliotecología)
- Periodismo y redacción técnica
- Informática y ciencias de la computación
- Ingeniería de la usabilidad (Interacción humano-computador)
- Marketing
- Arquitectura
- Gestión de productos
- Antropología
- Sociología
-



Diapositiva 94

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/06/2015

Componentes de la AI

- Esquemas y estructuras de organización → Cómo se realiza una clasificación en categorías para estructurar la información
- Sistemas de Etiquetado → Cómo los usuarios representan información
- Sistemas de Navegación → Cómo los usuarios navegan o se mueven a través de la información
- Buscadores → Cómo los usuarios buscan información

Para tener éxito, es necesario el conocimiento de los diversos **estándares** de la industria para crear, almacenar, acceder y presentar la información.

Sistemas de organización

- Los sistemas de organización se basan 100% en la comprensión del medio → esta comprensión es subjetiva y el lenguaje es dinámico y ambiguo

Tipos de sistemas de organización (*Organization Schemes*)

- Sistemas Exactos →
 - Alfabético
 - Cronológico
 - Geográfico
- Sistemas Ambiguos →
 - Por temas
 - Por tareas
 - Por audiencia
 - Por metáforas
 - Híbridos

Estructuras de Organización

- **Jerarquías** o taxonomías (aproximación de arriba-abajo)
- Modelos de **bases de datos** (aproximación de abajo-arriba)
- **Hipertextos**

Jerarquías o taxonomías

- Idealmente son divisiones mutuamente excluyentes
 - Existen ítems ambiguos en cuanto a su categoría: taxonomías polijerárquicas
 - Balance entre exclusividad e inclusividad
- Relaciones padre-hijo entre ellas

Muy adecuadas como base para establecer un "modelo mental" de la web

Jerarquía sencilla

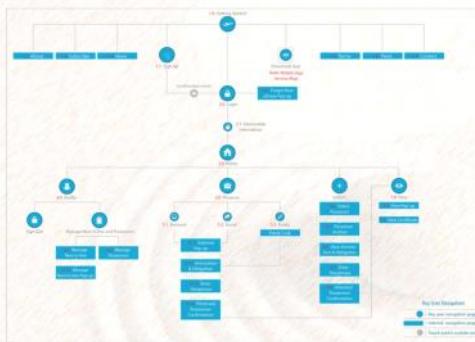
Organización jerárquica que refleje la estructura de la información que se quiera recoger en la Web



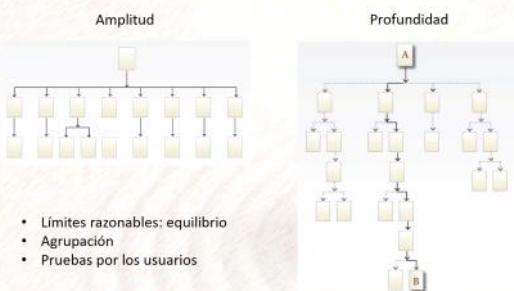
Ejemplos más complejos



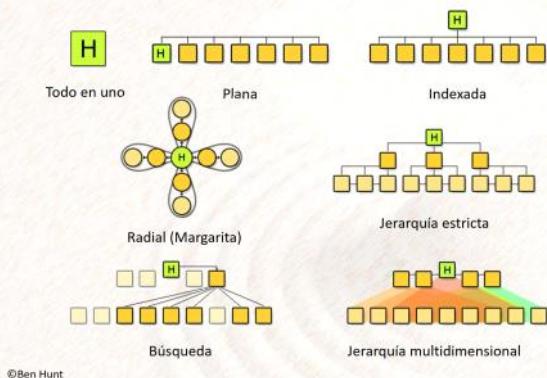
Ejemplos más complejos



Amplitud y profundidad



Arquitectura de la Información: Modelos



©Ben Hunt

Otras estructuras

Modelos de bases de datos

- Modelos de **bases de datos** (aproximación de abajo-arriba)
- Aplicado a los metadatos (conjunto de limitado de etiquetas) permite:
 - ✓ índices alfabéticos creados automáticamente
 - ✓ enlaces asociativos tipo "ver también"
 - ✓ búsqueda por campos
 - ✓ filtros avanzados y ordenación de resultados

Muy utilizado en gestores de contenido (CMS)

Hipertextos

fragmentos de información se unen a través de vínculos, en un complejo tejido de relaciones absolutamente no lineales y de naturaleza muy personal -> usados como estructuras complementarias

Clasificaciones sociales, Web 2.0

folksonomía o sistemas de clasificación colectiva

"taxonomía social u organización de la información de manera colectiva y colaborativa".

Cristóbal Cobo & Hugo Pardo



delicious

flickr

Significa que se va clasificando la información en la medida que las personas van aportando a través de los **tags** o etiquetas

Técnicas: creación de jerarquías

Ordenamiento de tarjetas (Card Sorting)

método fiable y barato, para encontrar patrones en la forma en que los usuarios esperan encontrar el contenido o funcionalidad

Las tarjetas representan una serie de ítems que se deben agrupar en función de la similitud temática

card sorting **cerrado**

+ en grupos previamente determinados y rotulados



Los participantes de la prueba deberán haber sido previamente seleccionados en etapas previas del proyecto, y tener un perfil representativo de los usuarios finales del sitio web.

card sorting **abierto**

+ libremente, en tantos grupos como cada usuario estime oportuno

Herramientas: jerarquías (1)

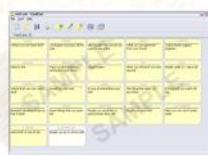
Ordenamiento de tarjetas (Card Sorting)

Manualmente

una serie de tarjetas en papel que deben agrupar en función de la similitud temática de los conceptos representados en cada una.

Herramientas y servicios Web

- EZSort (EZCalc y Usort) – IBM
- [UzCardSort](http://uzilla.mozilla.org/cardsort.html) para Mozilla, (<http://uzilla.mozilla.org/cardsort.html>)
- [CardZort](http://www.cardzort.com/) (<http://www.cardzort.com/>)
- [Web Sort / Optimal Sort](http://www.websort.net/) (<http://www.websort.net/>)



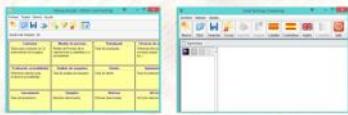
Herramientas: jerarquías (1)

Ordenamiento de tarjetas (Card Sorting)

Herramienta de CardSorting multiplataforma
(basada en java) realizada por el grupo de investigación GRIHO -
Universidad de Leida (<http://www.griho.udl.cat/>)



CardSortingGRIHO
(<http://www.grihotools.udl.cat/mpiuwa/wp-content/uploads/2014/08/CardSortingGRIHO.7z>)



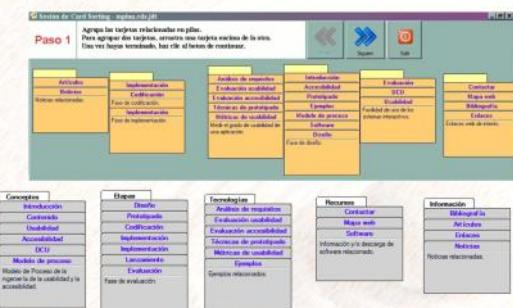
Ejercicio. Card Sorting

Arquitectura de la información: *Card Sorting*.

Utilizamos el programa citado y un conjunto de tarjetas correspondientes a una web de Diseño Centrado en el Usuario del grupo de Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad ([MPlua+a](#)).

Tomado del [Curso Introducción a la Interacción Persona-Ordenador](#) del grupo de investigación GRIHO - Universidad de Leida (http://griho2.udl.es/ipo/proto_card_sorting.html)

Resultado: Card Sorting



Etiquetado (Rótulación)

- Las etiquetas (o rótulos) son una forma de representación
- Cada uno representa una parte de información en el sitio web mayor que él mismo
- Comunican información con eficacia;
- Pueden ser **Textuales / Iónicos / Mixtos**
- Uso natural para crear sistemas de organización y navegación.

Utilización

- Enlaces contextuales
- Encabezamientos o Títulos
- Sistemas de navegación
- Términos de indización
 - ✓ Palabras claves
 - ✓ Tags
 - ✓ Metadatos
 - ✓ Taxonomía
 - ✓ Vocabulario controlado
 - ✓ Tesoro

Creación de Rótulos

- El lenguaje es simple pero **ambiguo**, además dinámico, existen sinónimos, equivalencias, neologismos, etc.
- No deben reflejar el pensamiento o punto de vista de los propietarios del sitio, si no de los **usuarios** que lo visitaran: conocer a la **audiencia** del sitio, para acotar lo mas posible el lenguaje a utilizar. Si la audiencia es amplia, es necesario hacer un buen intento
- Existen distintos **contextos, contenidos y usuarios**.
- Enfocarse no en una etiqueta en especial, sino en el **etiquetado como un sistema**: pensar semánticamente el conjunto de rótulos o etiquetas.
- Definir una **guía de estilos** de etiquetas o rótulos

Metodología Crear una tabla con tres columnas: etiqueta, explicación de la etiqueta y contenido que representa, para detectar inconsistencias y duplicidades.

Sistemas de navegación

En su creación, interaccionan

- o estrategia
- o estructura
- o diseño,
- o implementación.

- arquitectura de la información,
- diseño de la interacción,
- diseño de la información,
- diseño visual
- ingeniería de la usabilidad

- Dan acceso a la información del sitio
- Aprovechan el funcionamiento habitual de los navegadores
- Parten de la jerarquía del sitio, incorporando la flexibilidad de los hiperenlaces
- Permiten al usuario saber dónde se encuentra y donde puede ir.
- Reflejan nuestro modelo mental.
- Refleja el concepto de marca del sitio.
- Influye sobre la credibilidad del sitio.

Sistemas de navegación

- Sistemas de navegación embebidos

Navegación local	Navegación global
Navegación contextual	Contenido

 - global
 - local
 - contextual

- Sistemas de navegación suplementarios
 - sitemaps
 - índices
 - guías
 - búsquedas

- Sistemas de navegación avanzados
 - personalización
 - navegación visual
 - navegación social

Implementación

- barras y tabs
- menús dinámicos
- menús desplegables

- paso a paso
- por página
- con acceso directo a la página
- siguiendo una ruta
- árboles y directorios
- nube de tags
- índices a-z
- por conceptos, tesauros

Diapositiva 113

AC9

Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Sistemas de búsqueda

- El sistema de búsqueda, no puede ser una respuesta a un sistema de navegación defectuoso.
- Surge ligado a la **complejidad**: si tenemos una cantidad de páginas que va en aumento, con diversidad de temas, seguro que necesitas un sistema de búsqueda.



Diapositiva 114

AC9

Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Sistemas de búsqueda: tipos

- **Reactivos**: reaccionan frente a la conducta informativa del usuario.
- **Proactivos**: ofrecen proactivamente la información al usuario sin que tenga que reclamarla continuamente:
 - **Sistemas de difusión selectiva de información (DSI)**, ofrecen una actualización informativa automatizada sobre un tema concreto a partir de la **sindicalización** de contenidos.
 - **Sistemas de workshop**, suministran automáticamente la información dentro de un proceso a partir del perfil del usuario.
 - **Agentes inteligentes**: infieren el perfil de interés informacional de un usuario a partir de criterios como su histórico de comportamiento o su similitud con el de otros usuarios.

- **RSS**, es reactivo porque has de suscribirte y proactivo porque desde ese momento los recibes automáticamente.

Diapositiva 115

AC9

Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Vocabularios y metadatos

Junto a los 4 elementos básicos en la AI de cualquier sitio

- Esquemas y organización
- Sistemas de Etiquetado
- Sistemas de Navegación
- Buscadores

es posible añadir un elemento de mayor complejidad

Diapositiva 116

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Más información

Algunos sitios web en español

Observatorio

[Arquitectura de la información en entornos web](#)

Mario Pérez-Montoro

usableaccesible desde cero

[Arquitectura de información.](#)

Fundamentos

Olga Carreras



[Prototipado de sitios web](#)

Walter Maner



[La diagramación en la arquitectura de información](#)

Rodrigo Ronda León

AC9

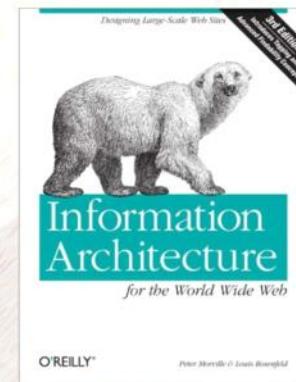
Arquitectura de la Información: Procedimientos y Metodología

Etapas
Iniciales

- ✓ Investigación
- ✓ Estrategia
- ✓ Diseño

Etapas
Finales

- ✓ Implementación
- ✓ Administración



Diapositiva 118

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Investigación: Research Framework

Contexto

Investigación antecedentes

Reuniones y presentaciones

Reuniones directorio

Evaluación tecnología

Contenido

Evaluación heurística

Análisis de contenido y meta datos

Mapeo de contenido

Análisis competencia

Usuarios

Análisis de *Logs*

Casos de uso y personas

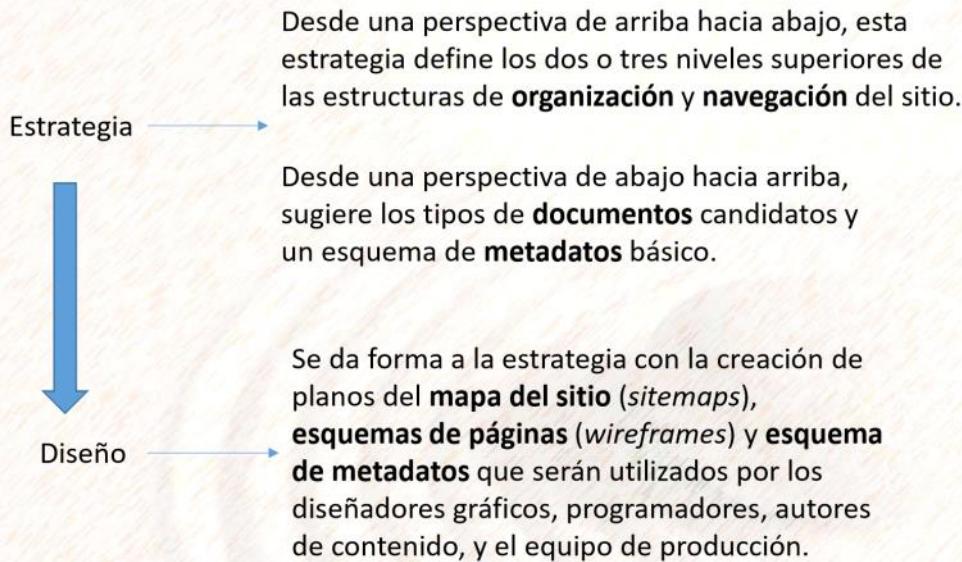
Investigación contextual

Entrevistas y pruebas a usuarios

Diapositiva 119

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Estrategia y diseño



Diapositiva 120

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Alejandro Cerezo, 12/04/2015

Planificación del proyecto: documentación

De la etapa inicial de planificación : **Arquitectura de la información** (*Information architecture*)



Documentación sobre distintos aspectos del proyecto

Estructura → **plano / mapa del sitio** (*sitemap*)

Disposición → **esquema de páginas** (*wireframes*)

Navegación → **guiones gráficos** (*storyboards*)

Diapositiva 121

AC6 Web Design Demystified
Wendy Willard
McGraw-Hill, 2011
Alejandro Cerezo, 07/04/2015

AC9

Plano /Mapa del sitio: concepto

Planos (*blueprints*) o mapas del sitio (*sitemaps*)*



muestra la relación entre las páginas y otros componentes del contenido

- refleja al mismo tiempo
- la organización
 - la navegación
 - el etiquetado

* mapa del sitio (*sitemap*) es también el nombre de un sistema complementario de navegación

Diapositiva 122

AC9 Information Architecture for the World Wide Web, 3rd Ed.
Peter Morville & Louis Rosenfeld
O'Reilly Media, 2006
Diagrama: Garrett, 2004-2015

Plano /Mapa del sitio como lenguaje visual

objetivo representar "las principales áreas de organización y rotulado"

enfocados a los aspectos estructurales y de funcionamiento del producto

Lenguaje visual

- se representan con textos, cajas y flechas.
- se definen con precisión el significado de cada uno de ellos
- se pueden detallar estos significados como leyenda del gráfico

vocabulario visual

Vocabulario visual de Garrett

Simbología básica para diagramar conceptos de arquitectura de información y diseño de interacción

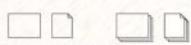
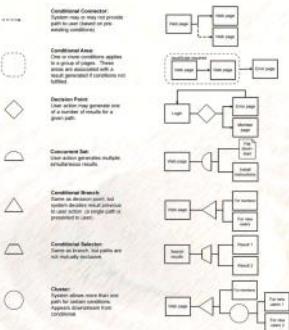


Figura 1.1 Comenzar a aprender el lenguaje de diagramas

Figura de Garrett. La foto de página 7 de la guía de diagramas.

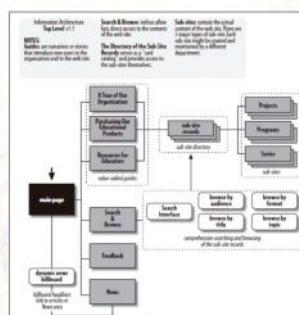
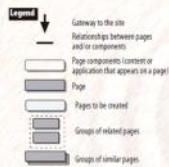
<http://www.jjg.net/ia/visvocab/spanish.html>

Proporciona librerías para incorporar este "lenguaje" en Visio, PowerPoint, OmniGraffle...



Modelo de Morville & Rosenfeld

Information Architecture for the World Wide Web. Peter Morville & Louis Rosenfeld. O'Reilly. 1998



Mapa del sitio: tools

Herramientas generales de diagramas

- [Visio](https://products.office.com/es-es/visio) (<https://products.office.com/es-es/visio>)
- [Gliffy](https://www.gliffy.com/) (<https://www.gliffy.com/>)
- [Lucidchart](https://www.lucidchart.com) (<https://www.lucidchart.com>)
- [ConceptDraw](http://www.conceptdraw.com/) (<http://www.conceptdraw.com/>)
- [Creately](http://www.creately.com/) (<http://www.creately.com/>)
- [Caco](https://cacoo.com) (<https://cacoo.com>)

Herramientas específicas

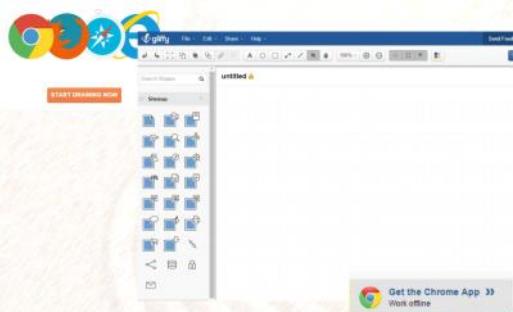
- [Slickplan](http://slickplan.com/) (<http://slickplan.com/>)
- [Writemaps](http://writemaps.com/) (<http://writemaps.com/>)

Herramientas de mapas conceptuales

- [XMind](http://www.xmind.net/) (<http://www.xmind.net/>)
- [yEd](http://www.yworks.com/en/products/yfiles/yed/) (<http://www.yworks.com/en/products/yfiles/yed/>)
- [VUE](http://sourceforge.net/projects/tuftsvue/) (<http://sourceforge.net/projects/tuftsvue/>)
- [Cmap](http://cmap.ihmc.us/) (<http://cmap.ihmc.us/>)
- [Freemind](http://freemind.sourceforge.net/) (<http://freemind.sourceforge.net/>)

Sitemaps con Glify

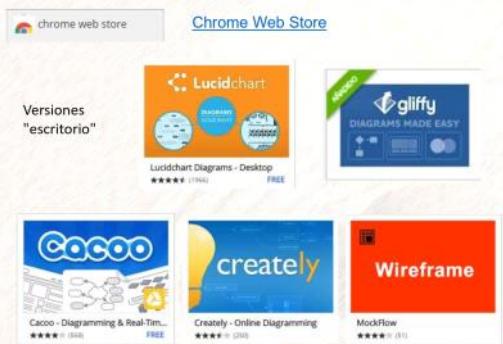
Gliffy Works Directly In Your Web Browser



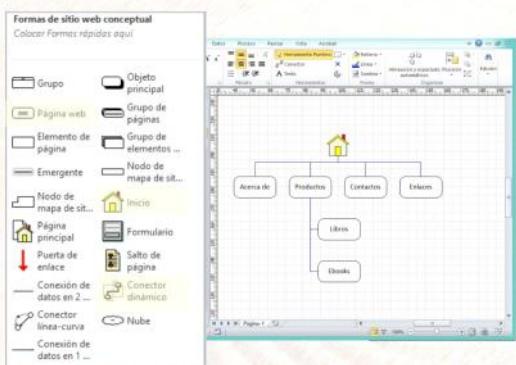
Diapositiva 126

AC6 Web Design Demystified
Wendy Willard
McGraw-Hill, 2011
Aljandro Cerezo 07/04/2015

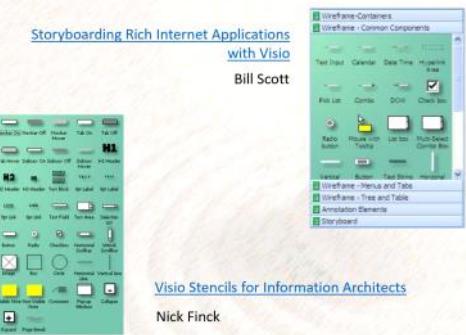
Chrome Apps



Sitemaps con Visio



Ampliaciones en Visio



Otras propuestas

[La diagramación en la arquitectura de información](#)

Rodrigo Ronda León

Después de hacer referencia a muchas de las propuestas existentes, presenta la suya propia

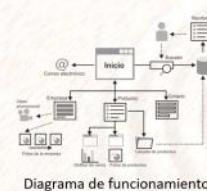
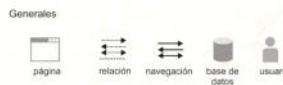


Diagrama de funcionamiento



Google XML Sitemaps

Como herramienta de documentación del sitio web

sitemaps generados a partir de sitios ya implementados

Google creó [Google Sitemaps](#)

Permite a los desarrolladores web construir y publicar listas de enlaces de sus sitios, debido a que algunas webs tienen un número grande de páginas dinámicas que sólo están disponibles cuando los usuarios emplean los formularios correspondientes.

En estos casos los archivos de *Sitemaps* en XML pueden usarse para indicar a una araña web cómo deben ser encontradas las páginas.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.example.com/foo.html</loc>
  </url>
</urlset>
```

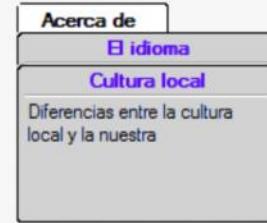
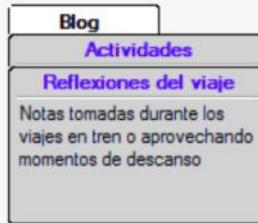
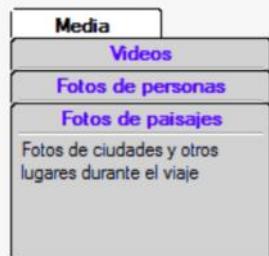
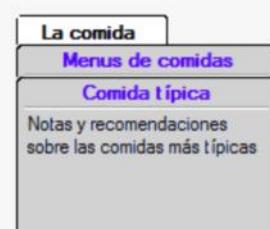
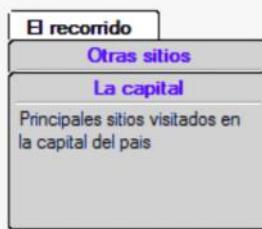
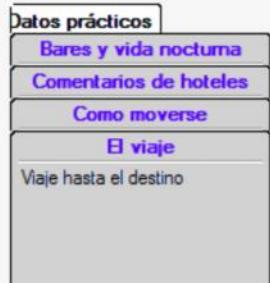
Ejercicio. Mapa del sitio

Página Web de un viaje a un país

- Analizamos la arquitectura de la información, e.g. con los elementos propuestos en el gráfico.
- Ordenamos los elementos en **categorías**
- Reflejamos el resultado en un posible **mapa del sitio (Sitemap)**

La capital Principales sitios visitados en la capital del país	El viaje Viaje hasta el destino	Cultura local Diferencias entre la cultura local y la nuestra
Actividades Notas sobre las actividades de cada día (Cuaderno de viaje)	Comentarios de hoteles Comentarios de los Hoteles	Otras sitios Otras ciudades y sitios visitados
Como moverse Desplazamientos entre los distintos puntos del recorrido	El idioma Palabras básicas para un viajero	Fotos de paisajes Fotos de ciudades y otros lugares durante el viaje
Bares y vida nocturna Un vistazo a los bares y la vida nocturna de las ciudades visitadas	Fotos de personas Fotos de conocidos o desconocidos durante el viaje	Menús de comidas Menús recogidos en restaurantes y cafeterías durante el viaje
Reflexiones del viaje Notas tomadas durante los viajes en tren o aprovechando momentos de descanso	Comida típica Notas y recomendaciones sobre las comidas más típicas	Videos Fragmentos tomados con la video-cámara digital

Resultado: jerarquía



Resultado: mapa del sitio



Diagramas de presentación

objetivo → mostrar el contenido de las páginas

Diferencias en cuanto a denominación y clasificación

Prototipos (*prototypes*):

- Prototipos de baja fidelidad o estáticos (*wireframes, mockup*)
- Prototipos de fidelidad intermedia (diseño gráfico)
- Prototipos de alta fidelidad o dinámicos (Web, HTML)

Fase de Arquitectura de la información (*Information architecture*)

- esquema de páginas (*sketches, wireframes*)
 - guiones gráficos (*storyboards*)
- Fase de Diseño / codificación
- maquetas (*mockup*)
 - prototipos (*prototypes*)

Del borrador (*sketch*) al esquema

Diseña la disposición básica (layout) del sitio Web: donde se sitian el contenido principal, el Sistema de navegación y los elementos adicionales



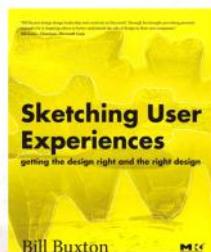
Diagramas de presentación de baja fidelidad: **estáticos**, sin elementos gráficos que muestran contenido y comportamiento.

Combinamos consideraciones gráficas y técnicas: se piensa en bloques, como los que internamente utiliza HTML, y al mismo tiempo en los elementos semánticos de HTML. La página web se divide en **bloques** que contienen menús (nav), artículos e imágenes (article), elementos adicionales (aside), además de una cabecera (header) y un pie de página (footer).

Borradores (*sketches*)

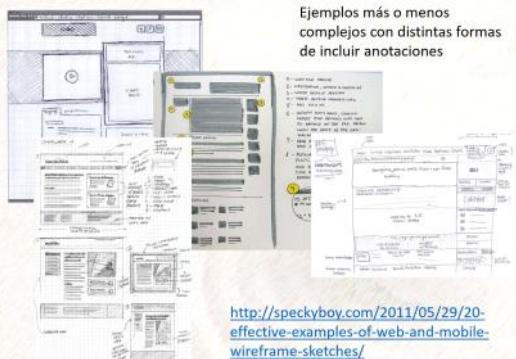
Una representación esquemática de una página web Sirven como

- herramienta de comunicación y discusión entre arquitectos de información, programadores y diseñadores
- primera presentación e intercambio de ideas con los clientes
- pruebas de usuarios.



Sketching User Experiences: Getting the Design Right and the Right Design. William Buxton.
Elsevier/ Morgan Kaufmann, 2007

Esquemas de página "a mano"



La hoja de cálculo como herramienta de diseño

ACB

A la hora de generar un borrador (sketch) con la disposición básica (layout) del sitio Web, Dan Rahmel sugirió el empleo de las hojas de cálculo como alternativa al tradicional dibujo



Diapositiva 142

ACB Advanced Joomla!
Dan Rahmel,
Apress, 2013
Alejandro Cerezo, 11/04/2015

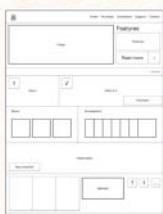
Esquemas de página (wireframes)

Un **wireframe** es más elaborado e incluye el inventariado de contenido, es decir, qué contenido debe estar presente en cada página.

- los elementos de la misma como cabeceras, enlaces, listas, formularios, etc.;
- el etiquetado de los vínculos o de los títulos;
- el *layout*, es decir, la ubicación, colocación y agrupación de los elementos de la página,
- la estrategia de navegación
- la priorización de contenidos dentro de la misma.
- puede incluir notas asociadas a los elementos relativas al comportamiento, indicando cómo deben mostrarse o definiendo su comportamiento funcional.

Cuando tenemos una secuencia de *wireframes* hablamos de un *Storyboard*.

Esquemas de página digitales



Esquemas más simples



Esquemas ligados a una rejilla (grid)



Esquemas más complejos

<http://speckyboy.com/2011/05/29/20-effective-examples-of-web-and-mobile-wireframe-sketches/>

Esquema del sitio: tools

- [Balsamiq](http://www.balsamiq.com/) (<http://www.balsamiq.com/>)
- [Axure](http://www.axure.com/Default.aspx) (<http://www.axure.com/Default.aspx>)
- [FlairBuilder](http://www.flairbuilder.com/) (<http://www.flairbuilder.com/>)
- [Pidoco](http://pidoco.com/) (<http://pidoco.com/>)
- [ProtoShare](http://www.protoshare.com/) (<http://www.protoshare.com/>)
- [Mockingbird](https://gomockingbird.com/) (<https://gomockingbird.com/>)
- [Hotgloo](http://blog.hotgloo.com/) (<http://blog.hotgloo.com/>)
- [Mockflow](http://www.mockflow.com/) (<http://www.mockflow.com/>)
- [Cacoo](https://cacoo.com/) (<https://cacoo.com/>)
- [Createely](http://www.createely.com/) (<http://www.createely.com/>)
- [Pencil Project](http://pencil.evolus.vn/en-US/Home.aspx) (<http://pencil.evolus.vn/en-US/Home.aspx>)
- [DENIM from DUB](http://dub.washington.edu:2007/denim/) (<http://dub.washington.edu:2007/denim/>)
- [Framebox](http://framebox.org/) (<http://framebox.org/>)
- [Gliffy](https://www.gliffy.com/) (<https://www.gliffy.com/>)

- Visio
- Gliffy
- Photoshop
- InDesign
- Fireworks
- Illustrator
- CorelDraw
- ...

Pencil Project



Pantalla inicial del programa una vez instalado en Windows

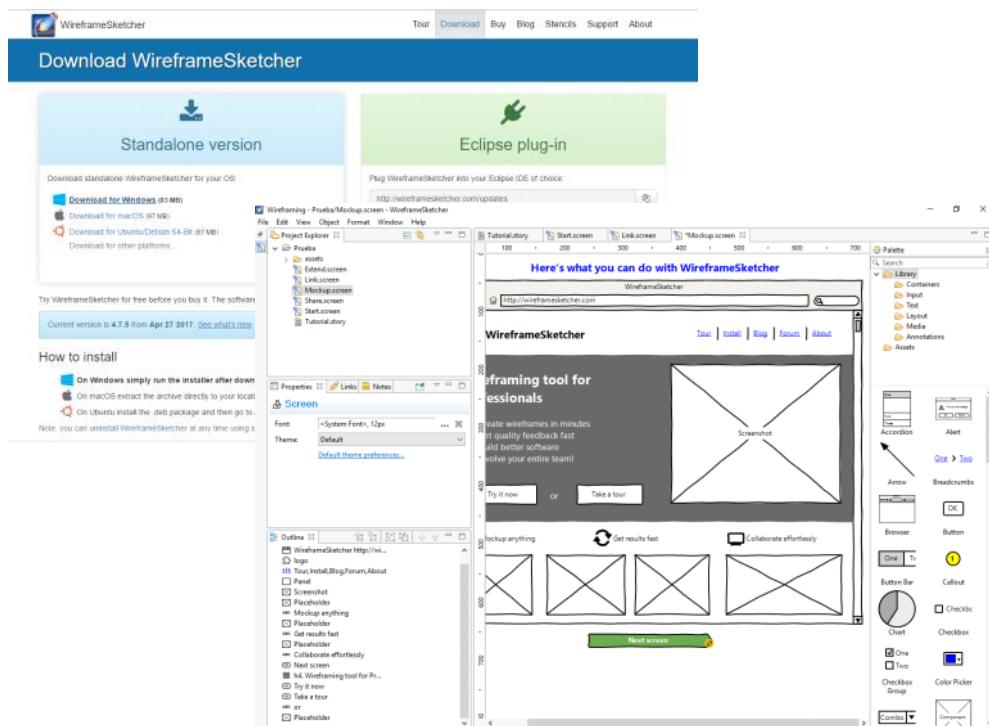
Prototipos de alta fidelidad

Van más allá de la fase de Planificación (Arquitectura de la información), correspondiendo ya a las **fases de Diseño y Codificación**: Nunca se debe comenzar a prototipar sin haber definido primero los objetivos del cliente, las necesidades de los usuarios, los requisitos del proyecto y la arquitectura de información de la aplicación web.

Las **maquetas (mockups)**, permiten detallar el proceso interactivo de una o varias tareas

Los **prototipos funcionales o maquetas dinámicas**, normalmente en HTML, que simulan o tienen implementadas partes del sistema final a desarrollar como observamos en este ejemplo.

<http://wireframesketcher.com/>



Ejercicio. Wireframe (Esquema)

Arquitectura de la información.
Esquema de páginas (*Wireframe*)

Diseño Técnico: Maquetación

1. Etapas del diseño Web
 - a. Planificación de proyectos
 - b. Arquitectura de la información
 - c. Planos / Mapas del sitio
 - d. Esquemas
 - e. Alojamiento y publicación
2. Estructura y disposición
 - a. Maquetación con divs. Grids
 - b. Disposition.
 - i. Disposición fija
 - ii. Disposición fluida
3. Modelos de posicionamiento.
 - a. Posiciones relativas
 - b. Posiciones absolutas y fijas
4. Modos de display complejos
 - a. Table
 - b. Flex-Box
 - c. Grid
5. Interfaz de usuario (UI)
 - a. Elementos de navegación
 - i. GUI: Menús de Navegación
 - ii. GUI: Navegación con imágenes
 - b. Entrada de datos. CSS y formulario
6. RWD
 - a. Media queries.
 - b. Navegación responsive
 - c. Bootstrap 4
7. Mejoras gráficas
 - a. Fondos con Sprites y rollovers
 - b. CSS Dinámico
 - i. Transiciones,
 - ii. Transformaciones,
 - iii. Animaciones



Etapas del diseño Web

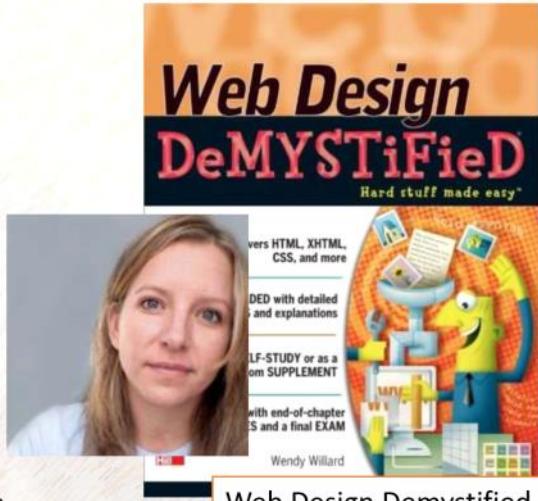
Lunes, 19 de febrero de 2018 11:37

1. Planificación de proyectos
2. Arquitectura de la información
3. Planos / Mapas del sitio
4. Esquemas
5. Alojamiento y publicación



Diseño Web: Tareas

- Gestión de proyectos
(*Project management*)
- Arquitectura de la información
(*Information architecture*)
- **Diseño gráfico**
(*Graphic Web Design*)
- **Codificación / programación**
(*Coding / programming*)
- Animación (*Animation*)
- Redacción / Edición
(*Copywriting / editing*)
- Pruebas (*Testing*)
- Posicionamiento en buscadores
(*Search engine optimization, SEO*)
- Mantenimiento (*Maintenance*)



Web Design Demystified.
Wendy Willard.
McGraw-Hill, 2011

Etapas del Diseño Web

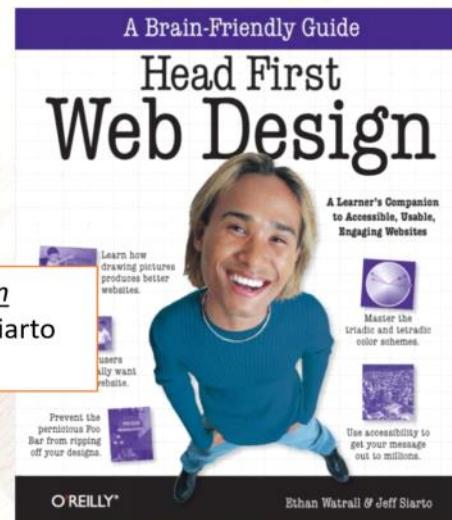


Otra aproximación

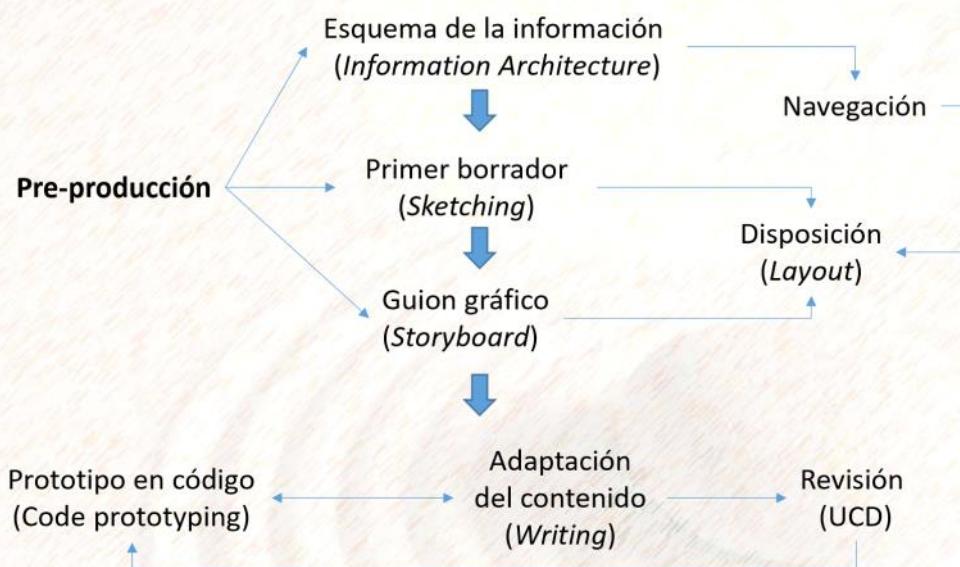
Desde una perspectiva muy práctica, plantea una mayor **integración del diseño gráfico**, incorporando en las fases de preproducción decisiones sobre el aspecto y el color, con el fin de conseguir que el sitio web sea una "metáfora visual" de lo que trata de representar.

Head First Web Design
Ethan Watrall & Jeff Siarto
O'Reilly Media, 2009

Las etapas descritas luego son bastantes similares a las de otros trabajos



Diseño / Implementación de una Web



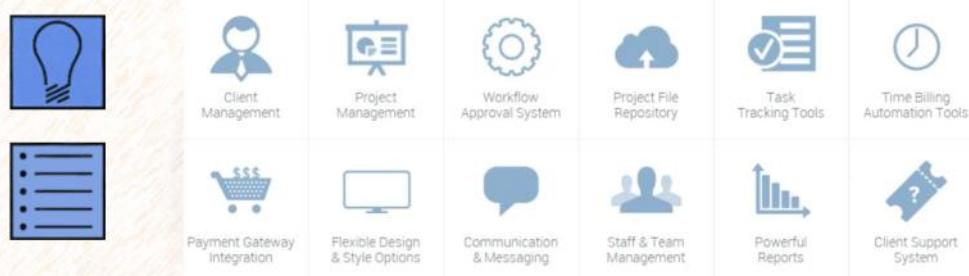
Head First Web Design
Ethan Watrall & Jeff Siarto
O'Reilly Media, 2009

Etapas del Diseño Web



Gestión de proyectos

- Evaluación de las **metas y objetivos** del proyecto
- Identificación de las posibles **áreas problemáticas**
- **Comunicación** con el cliente acerca del el proyecto
- Identificación del **público objetivo** del sitio web
- Recopilación del **contenido** del sitio web (→ Arquitectura Información)
- **Programación** del proyecto y sus etapas
- Mantenimiento de todas ellas en la **fecha prevista**
- Información al cliente sobre el **proceso** del proyecto y el mantenimiento del sitio



Arquitectura de la información

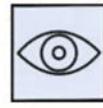
- Definición y etiquetado de las **áreas de contenido** del sitio web
- **Organización** del contenido
- Desarrollo de una **estructura** eficiente, centrado en el usuario, para el contenido del sitio web
- Documentación del **mapa del sitio** (*sitemap*)
- Comprobación de que la **navegación** cumple con los requisitos de los usuarios finales
- Creación de **esquema de páginas** (*wireframes*) y de **guiones gráficos** (*storyboards*) para seguir el progreso de un usuario a través del sitio web



Diseño gráfico

- Creación de diseños integrales o **maquetas** (*mockups*) para las secciones / páginas / pantallas clave del proyecto
- Traducción de esas maquetas a **prototipos** completamente funcionales (*working prototypes*).
- Desarrollo de **contenido gráfico complementario** según sea necesario
- Identificación de las **transiciones** entre secciones / pantallas / páginas
- **Codificación** de los diseños para que aparezca en los navegadores web

maqueta
(*mockup*)



Codificación / programación

- **Codificación de los diseños** para que aparezca en los navegadores web
- Diseño y mantenimiento de **bases de datos** apropiadas,
- Diseño y mantenimiento del conjunto de **sistemas necesarios** para la implementación del sitio web, incluyendo servidores, entornos de prueba, procedimientos de seguridad, redes...

Diseño grafico



Codificación

Completa interrelación

El diseñador, incluso si no es el que codifica, debería tener sólidos conocimientos de ese proceso

Animación / Multimedia

En algunos sitios Web, los elementos multimedia animados adquieren suficiente complejidad como para diferenciarse como una actividad independiente, que puede ser responsabilidad de un especialista en este área

- Diseño de las secciones / páginas / pantallas clave de la animación
- Desarrollo de la animación entre fotogramas / pantallas
- Codificación / programación de la animación



Redacción / Edición

- Colaboración con el arquitecto de información y / o diseñador para definir y etiquetar las áreas de contenido del sitio
- Desarrollo de los textos correspondientes al contenido del sitio
- Revisión y edición de los textos del sitio web para facilitar la lectura

Los textos deben ser creados específicamente para la web, respondiendo a las particulares características de este medio.

Pruebas

Control de calidad
(*quality assurance, QA*) → Objetivo: usabilidad

- Comprobación del funcionamiento de cada **enlace** tal como se espera de él.
- Comprobación de la funcionalidad de cada **formulario** de la web después de completarlo y probar su envío
- Prueba de la **capacidad de uso** de la página web para determinar hasta qué punto los usuarios pueden acceder correctamente al contenido
- Verificación de que todo el texto y los datos gráficos se **visualizan** como se esperaba en todos los navegadores y plataformas considerados como posible destino del sitio web
- Coordinación con el miembro del equipo responsable de **corregir** cada uno de los errores encontrados durante la prueba

Posicionamiento en buscadores (SEO)

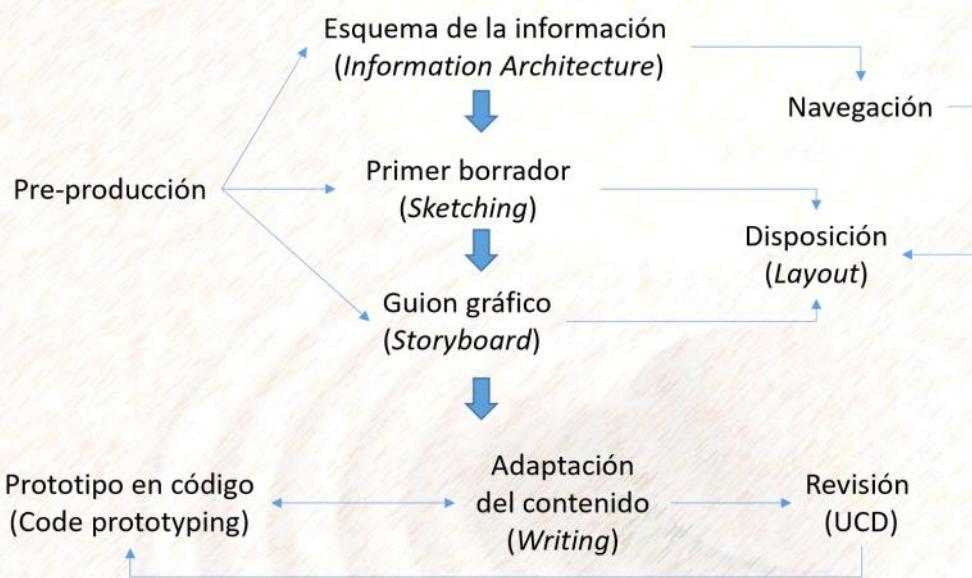
- Creación de **referencias cruzadas** (*cross-linking*) entre páginas para facilitar la navegación y aumentar la visibilidad
- Revisión y edición del contenido del sitio para añadir **palabras clave** y frases relevantes
- Incorporación de la web a los **motores de búsqueda**
- Comprobación de la lista del sitio en los motores de búsqueda principales
- La identificación de posibles cambios en el sitio que puede aumentar su **visibilidad y / o popularidad**
- Bloqueo de páginas privadas para evitar que sean registrados en los motores de búsqueda
- **Publicidad** con motores de búsqueda para aumentar la visibilidad

Mantenimiento

- Actualización de comunicados de prensa y noticias
- Incorporación de nuevos contenidos
- Otros trabajos de publicación
- Actualización de los listados del personal
- Mantenimiento actualizado del calendario

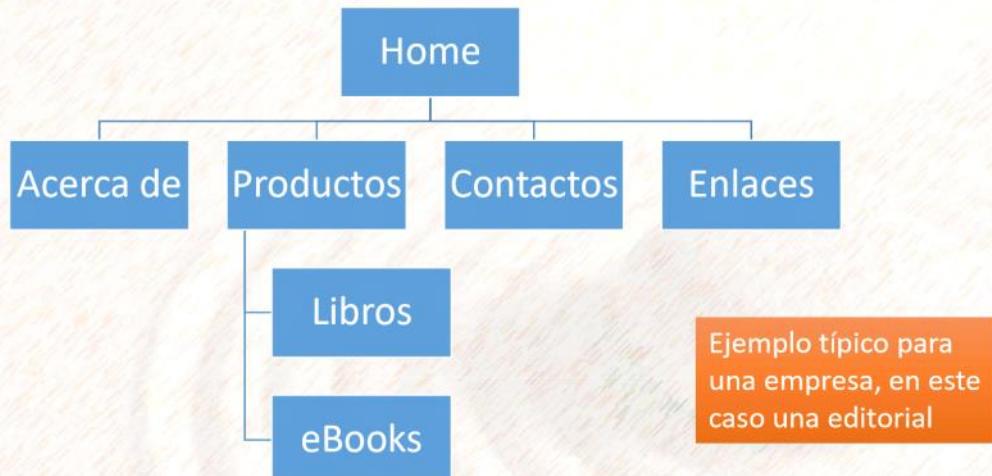
El proyecto inicial debe definir claramente las responsabilidades en relación con el mantenimiento del sitio

Diseño / Implementación de una Web



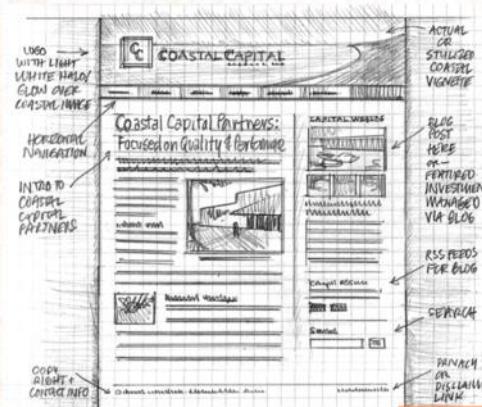
Esquema de la información

Organización jerárquica que refleja la estructura de la información que se quiera recoger en la Web



Primer borrador

Diseña la disposición básica (layout) del sitio Web: donde se sitian el contenido principal, el Sistema de navegación y los elementos adicionales?



Combinamos consideraciones graficas y técnicas: se piensa en bloques, como los que internamente utiliza HTML, y al mismo tiempo en los elementos semánticos de HTLM5.

La página web se divide en bloques que contienen menús (nav), artículos e imágenes (article), elementos adicionales (aside), además de una cabecera (header) y un pie de página (footer).

El resultado es una representación esquemática de la página (**wireframe**)

La hoja de cálculo como herramienta de diseño

A la hora de generar un borrador (sketch) con la disposición básica (layout) del sitio Web, Dan Rahmel sugiera el empleo de las hojas de cálculo como alternativa al tradicional dibujo

The image displays two wireframe sketches of website layouts, each featuring a header, sidebar, main content area, and footer.

BANNER Layout:

- Header:** Contains a search bar and navigation links for Home, Articles, News, and About Us.
- Left Sidebar:** Titled "Topics" with sections for First Menu, Second Menu, and Third Menu.
- Content Area:** Titled "Recent Articles" with links to Article 1, Article 2, and Article 3.
- Right Sidebar:** Titled "Topics" with sections for First Menu, Second Menu, and Third Menu.
- Footer:** Includes links for Home, Terms & Conditions, Advertise with us, Contact us, RSS, Site Map, and Copyright 2013, All Rights Reserved.

SMALL BANNER Layout:

- Header:** Contains a search bar and navigation links for Home, Articles, News, and About Us.
- Left Sidebar:** Titled "Topics" with sections for First Menu, Second Menu, and Third Menu.
- Content Area:** Contains placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sed pulvinar risus. Aenean condimentum, magna vitae sodales tincidunt, justo velit blandit sem, non interdum quam ligula ut tortor. Donec placerat hendrerit ligula, vel placerat turpis volutpat sit amet. Proin pretium tempus dictum. Nullam sollicitudin vulputate nibh, nec aliquet velit pharetra a. Sed mollis ultricies, nunc id lacinia fermentum, enim tellus. Fusce euismod viverra tortor, a dictum dolor lobortis et. Phasellus arcu tempor, pharetra vel consectetur vitae, fringilla in odio. Nulla facilisi. Morbi in neque eu tortor elementum fringilla sit amet a dolor. Cras neque turpis, lobortis rhoncus lobortis in, dapibus vitae mauris."
- Right Sidebar:** Titled "AD" with a section for Related Articles containing links to Article 1, Article 2, and Article 3.
- Footer:** Includes links for Home, Terms & Conditions, Advertise with us, Contact us, RSS, Site Map, and Copyright 2013, All Rights Reserved.

Guion gráfico: Maqueta

Diseño gráfico
(Designing)

- Elección del esquema de colores del sitio
- Elección del esquema de fuentes
- Creación de un *banner* para la cabecera
- Creación de una maqueta (mockup), con la disposición (layout) y los anteriores elementos.



Será necesario emplear un editor gráfico como Photoshop o GIMP

El resultado es la maqueta del sitio diseñado, algunos de cuyos elementos, como el logo o algunas imágenes, se usarán en la implementación final.

Desarrollo del código

Desarrollo del
sitio Web
(Coding)

El diseño se transforma en el código de una página web real, en HMTL and CSS.

El fichero **index.html / index.php ...** recoge la estructura de la página en HTML, con total independencia de su presentación. Se establecen los identificadores (class o id) de cada una de las áreas definidas en dicha estructura

La hoja de estilos **css** refleje el diseño elegido, asignando las características de presentación de cada elemento definido en el fichero anterior

En esta etapa se aplican las especificaciones técnicas de usabilidad, accesibilidad... genéricas o definidas para el proyecto

Pre-producción

En su conjunto esta fase implica la toma de las decisiones principales

Idealmente, intentamos conseguir que la web sea una metáfora visual de aquello de lo que trata

Decidimos como será la **navegación** en el sitio, siempre en función del esquema de la información establecido

Decidimos la **estructura** de los elementos como parte de esa metáfora

Decidimos como encajaremos esos elementos en el espacio disponible, eligiendo un tipo u otro de **disposición**

Lunes, 19 de febrero de 2018 11:37

Diseño, estructura y disposición

Estructura Web

- . Estructura y composición de páginas
- Divisiones (capas). Disposición y formato.
- Disposición de los elementos semánticos.

Disposición fija

- Tamaño de pantalla y resolución. Tipos de disposición.
- Disposición fija. Primer pantallazo. Optimización visual
- Proporciones en la Web. Rejillas fijas. Implementación: 960



Estructura y composición

La composición debe guiar al usuario visualmente.

El éxito de una buena composición de página consiste en una organización balanceada, clara y concisa de la información usando las herramientas básicas de diseño

- formato editorial,
- tipografía,
- color y
- uso de imágenes

El contraste es importante



Destacar las distintas áreas de información

Establecer áreas "verticales".



Páginas principales o portales de información

Diseño Web: Estructura y composición

Existen una serie de elementos que hay que distribuir en el espacio disponible

Los **encabezados** deben de presentar la identidad gráfica del sitio

El **contenido** principal y secundario

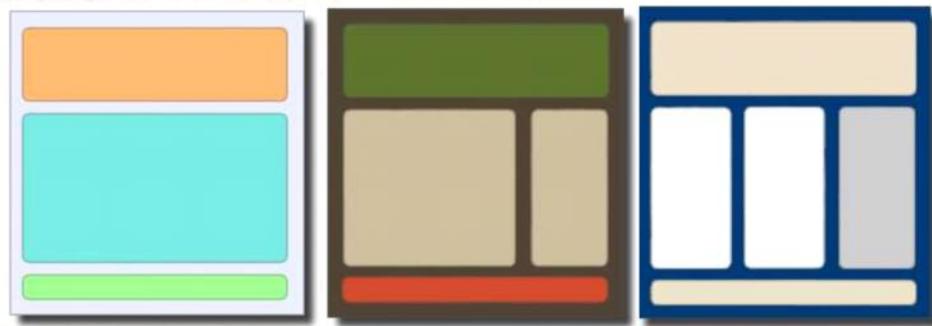
Los **pies de página** presentan por lo general el origen del sitio y metadatos acerca del sitio: información legal, derechos de autor, año de producción, contacto, etc

Consistencia interna



Usar plantillas

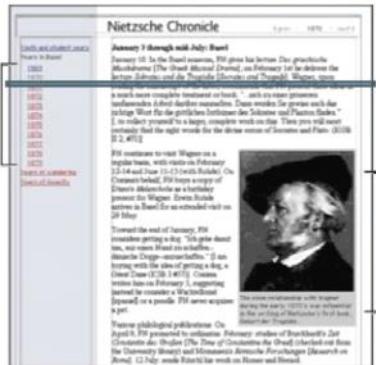
Modelos de estructura



Nos centraremos en la estructura posible, aunque realmente es su combinación con un color la que contribuye a construir la metáfora visual que buscamos. Más adelante veremos los colores

Implementar la estructura: tablas y divisiones

Mediante tablas o divisiones se pueden crear columnas para dividir la página y facilitar la distribución de información, manteniendo separadores de columnas (Gutters) y márgenes



Vemos como la tablas y divisiones permiten incorporar gráficas y pie de foto dentro de un área de texto largo

¿Tablas o divisiones?

Como indica la [W3C](#), las tablas como elemento para presentar **datos tabulados** no están obsoletas en HTML5, que solo ha cambiado la forma en que se definen algunas propiedades (usando CSS en HTML5).

La misma [W3C](#) explica claramente que usar tablas como **elemento de maquetación** no es lo adecuado, ya que para ello se dispone de las divisiones, incluidas sus nuevas variantes semánticas, y las hojas de estilo (CSS)

Además el uso de tablas presenta ciertas limitaciones reales:

- se logra un diseño muy "cuadrado"
- no es flexible a los distintos tamaños de pantallas (de 24", de 17", tablets de 7" y celulares de 4").
- Los sucesivos niveles de anidación generan un código complejo y difícil de modificar, lleno de td's y tr's



Maquetación con divs

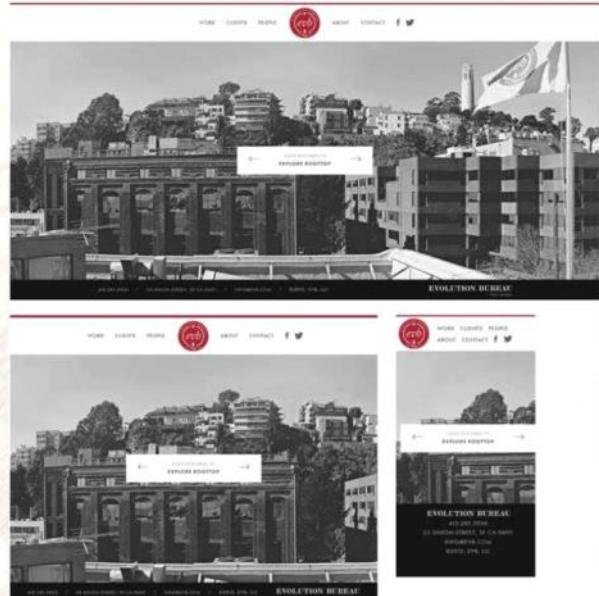
lunes, 5 de junio de 2017 9:07

Diseño Web: Maquetación con divisiones

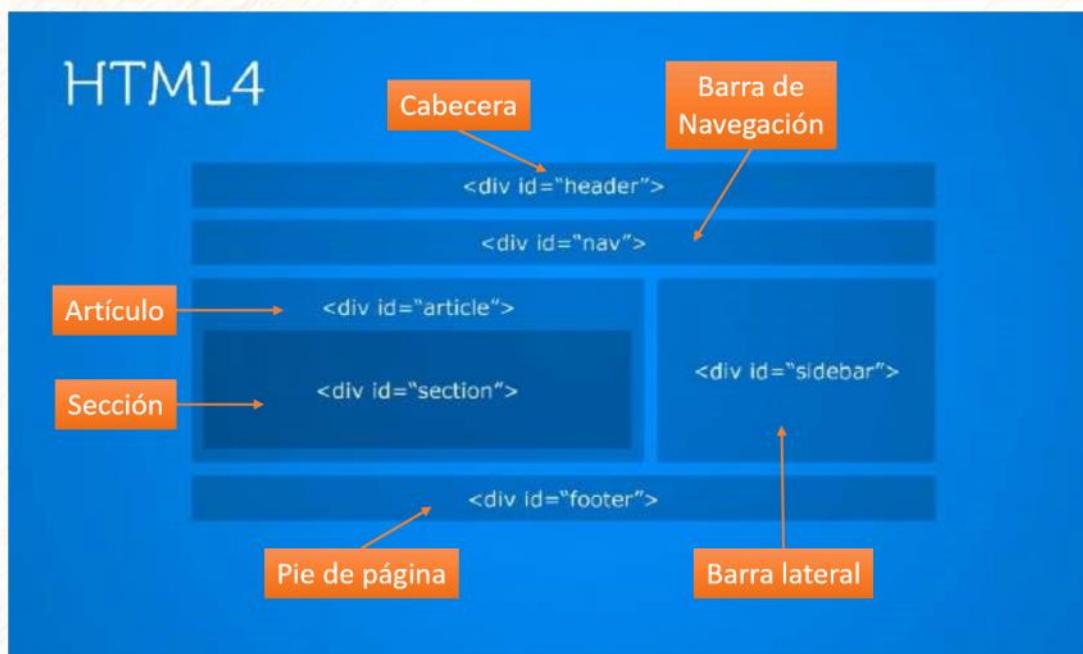


Espacios irregulares
o "no cuadrados"

Adaptación al tamaño de
pantalla



Divisiones y disposición de elementos



Divisiones o capas: la etiqueta div

<div> define una división o sección que agrupa diversos elementos de bloque (encabezados, párrafos, listas) e incluso imágenes.

Esta agrupación responde a varias razones

- a. Conceptualmente los elementos comparten algo que hace que tenga sentido agruparlos
- b. Hay un formato específico, recogido en un **estilo**. Habitualmente el atributo **class** recoge el nombre de la división para luego aplicar el estilo. (El único atributo específico de div, **align**, no es soportado en HTML5).
- c. Estos grupos se convierten en el eje central de la **distribución** de los elementos (layout) sustituyendo el uso de tablas como elemento de planificación de la página web

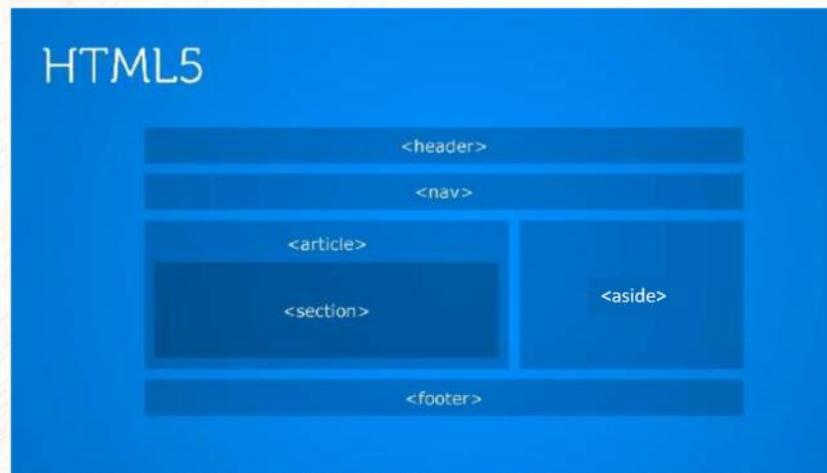
```
<div class="cap">
  <h3>Título del capítulo</h3>
  <p>Párrafos del capítulo.</p>
</div>
```

Divisiones semánticas



Nuevas
etiquetas, que
redefinen el
uso de <div>

header
nav
article / section
aside (elemento tangencial)
footer (meta información a pie de página)



Disposición de elementos (layout)

Estilos Divisiones (*div, header, nav, article, section, aside, footer*)

height ↗
width ↗ pixeles, igual que en otras etiquetas de bloques

float ↗ *left*
clear ↗ *right* *Float* indica si un elemento puede o no flotar en una dirección. *Clear* indica los lados de un elemento en los que no se pueden situar otros elementos flotantes.

position ↗ *absolute*: posición fija respecto al elemento contenedor (*parent*)
 → *relative*: desplazamiento de su posición natural
 → *fixed*: valor fijo en la ventana del navegador

top
right
bottom
left ↗ valores en pixeles correspondientes a la posición

Disposición de elementos (layout)

Estilos Divisiones (div, header, nav, article, section, aside, footer)

display	Especifica como debe aparecer un elemento	Box, block, flex, in-line, list-item, table
visibility	Elemento visible o no	
overflow	Especifica que ocurre si el contenido desborda (<i>overflows</i>) la caja de un elemento	
z-index	Establece el orden en que se apila (<i>stack</i>) un elemento	

Formato de las divisiones

Estilos

Divisiones (div, header, nav, article, section, aside, footer)

margin
(-top, -right, -bottom, or -left) → igual que en otras etiquetas de bloques

padding
(-top, -right, -bottom, or -left)

color
background-color → igual que en otras etiquetas de bloques

Ejemplo: HTML

```
<body>
  <div id="container">
    <div id="header">
      <h1>Main Title of Web Page (2)</h1>
    </div>
    <div id="menu">
      <h2>Menu</h2>
      <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JavaScript</li>
      </ul>
    </div>
    <div id="content">
      <p>Content goes here</p>
    </div>
    <div id="footer">
      <p>Copyright © W3Schools.com</p>
    </div>
  </div>
</body>
```

Main Title of Web Page

Menu

- HTML
- CSS
- JavaScript

Content goes here

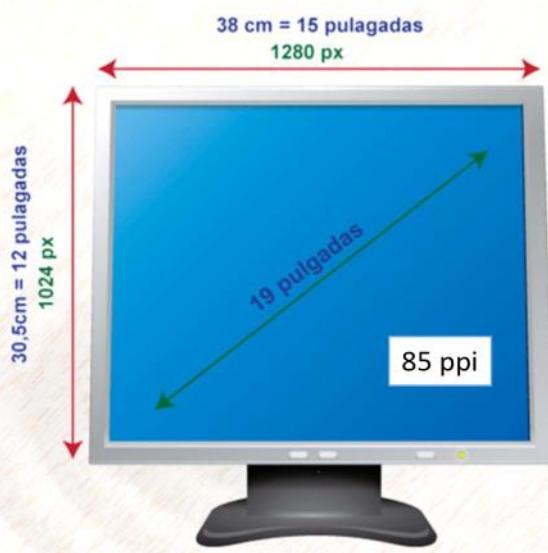
Copyright © ConMasFuturo.com

Tamaños de pantalla y resolución

La “resolución” se mide en pixeles y para cada dispositivo puede configurarse dentro de un rango, hasta un máximo que depende de su hardware.

Cualquier dispositivo tiene unas **dimensiones físicas**, que podemos expresar en cm, mm, pulgadas... de largo y de ancho

La **resolución** real corresponde a la **densidad de píxeles** (e.g. pixelex por pulgada, ppi) que se obtiene al dividir una longitud de la imagen en pixel por la correspondiente en pulgadas



AC3

Tamaños de pantalla y resolución

Los primeros PC estaban en unos 72 ppi

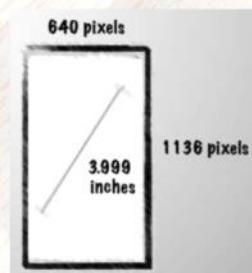


19" – 1280 x 1024 (85 ppi)
30" – 2560 x 1600



10"
1280x800
2048x1536

Una Tablet de gama alta
132 ppi



4" / 4.8"
640 x 1136
720 x 1280

HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

Resolución: espacio disponible

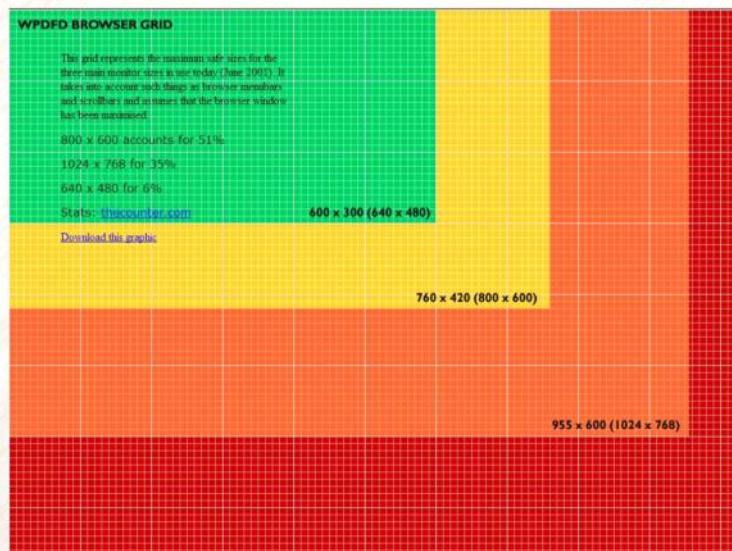
Dimensiones:

Limitaciones impuestas por el medio físico.
Tradicionalmente más habituales de los monitores

- 1. 1024 x 768
- 2. 800 x 600

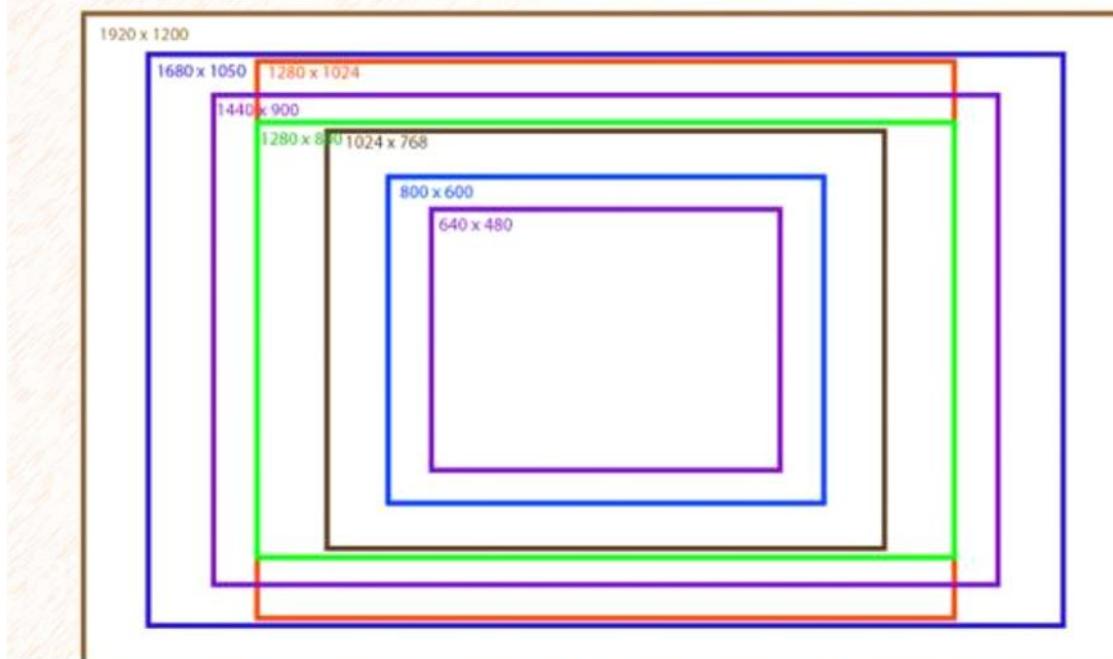
Y las tendencias actuales

- 1220 x 1024
- Tabletas, móviles



AC3

Resoluciones habituales



AC3

AC4

Tipos de disposición

- Disposición fija
(*Fixed width layout*)
- Disposición fluida
o “líquida”
(*Liquid layout*)
- Disposición
elástica
(*Elastic layout*)
- Disposición híbrida
(*Hybrid Layout*)

```
body {  
width: 90%;  
margin: 0 auto;}
```



```
body {  
width: 960px;  
margin: 0 auto;}
```

HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

Learning Web Design, Fourth Edition
Jennifer Niederst Robbins
O'Reilly Media 2012

Disposición fija

lunes, 5 de junio de 2017 9:10

Disposición fija

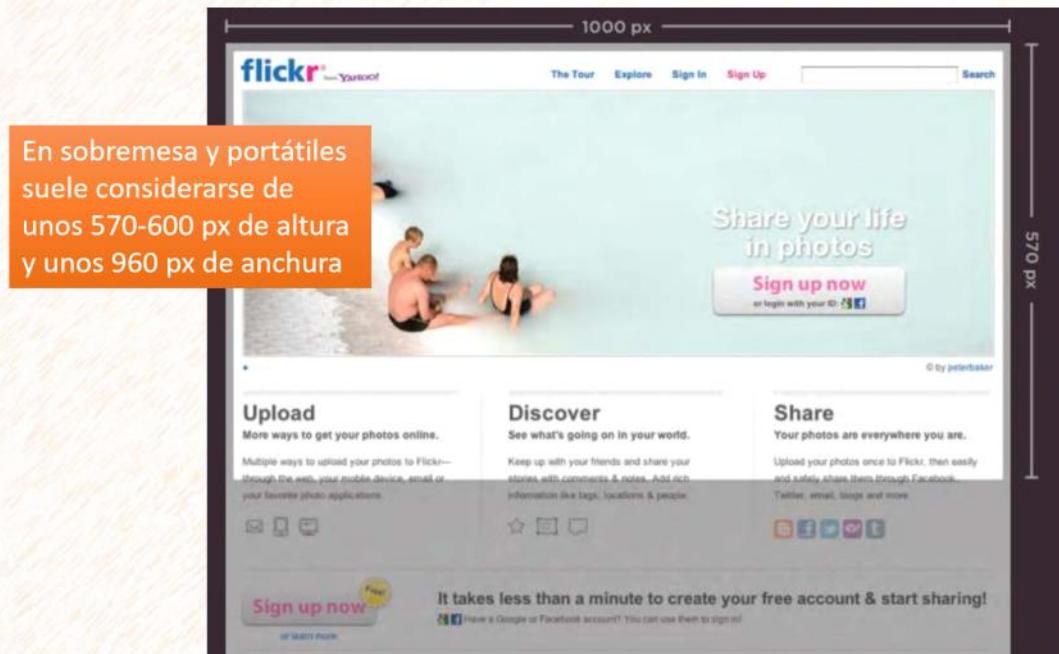


Ya en 2009, poco más del 2,7% de los usuarios navegaba con resoluciones inferiores a XVGA, que a su vez era la usada por el 40%

Se convirtió en lo habitual optimizar los diseños fijos para este ancho de pantalla, dando a las páginas unos **960 px de ancho** y dejando unos 65 px para la barra de scroll de la derecha del navegador

AC3

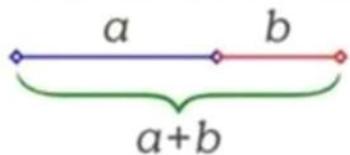
Primer pantallazo (*above the fold*)



HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

Optimización visual (1)

El número áureo

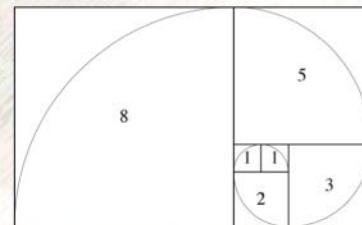


$$\frac{a+b}{a} = \frac{a}{b} = \varphi$$

Descubierto en la antigüedad y presente en numeroso ejemplos en la naturaleza (flores, cristales, conchas galaxias...) y en el arte

Euclides definió su valor diciendo, de forma resumida que dos partes de un segmento están en razón áurea si La relación entre el segmento mayor y la recta debe ser la misma que la que existe entre las dos partes del segmento

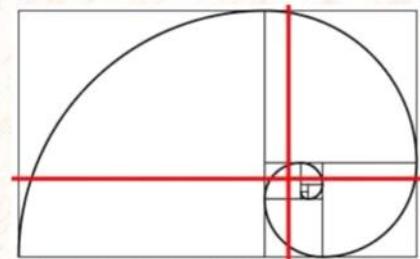
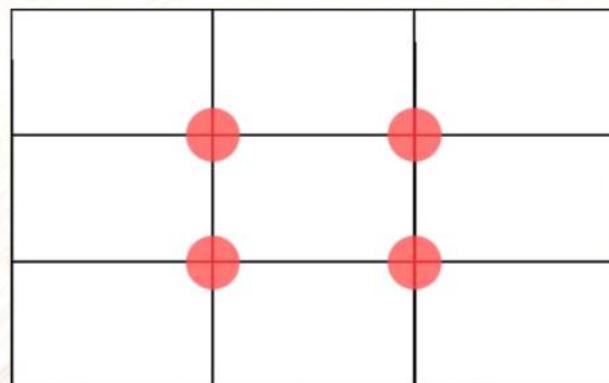
Fibonacci extendió el concepto a 2 números positivos y definió la famosa secuencia en la que la relación aurea aparece siempre al dividir un término por el anterior. Surge así un numero de infinitos decimales, el número ϕ (phi), 1,6180...



Optimización visual (2)

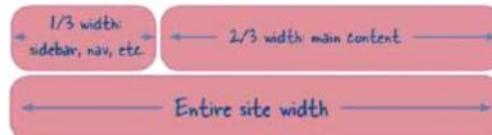
regla de los tercios

forma de composición para ordenar objetos dentro de la imagen: recomienda utilizar los puntos de intersección de estas líneas para distribuir los objetos de la escena



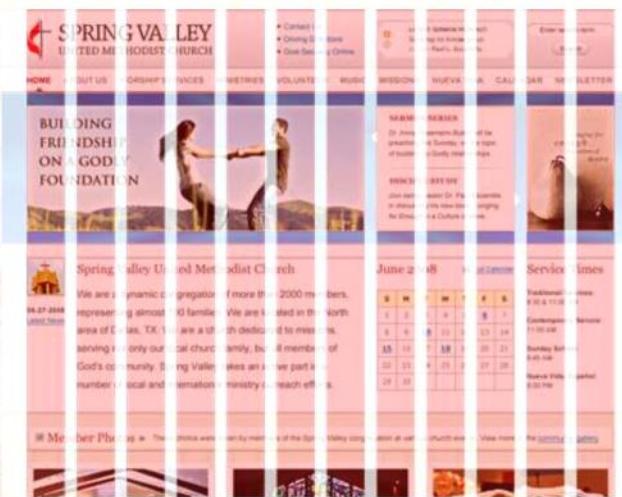
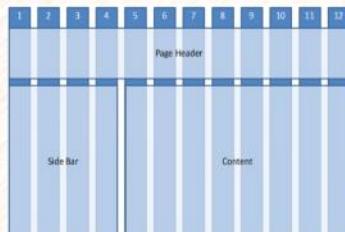
Cada punto deja 1/3 a un lado y 2/3 al otro, o el 33,3 % frente al 66.7 %, una versión simplificada del número aureo

Aplicación a la Web (1)



Aplicando proporciones a la Web (2)

Como ayuda para distribuir correctamente los elementos en alguna proporción se utiliza una rejilla (como una cuadrícula sólo en vertical)



AC3

Frameworks: rejillas fijas

960 Grid System

960.gs



Blueprint

www.blueprintcss.org



BlueTrip

bluetrip.org



AC3

Rejillas (grids) en 960



Rejilla de 12 columnas para disposiciones fijas de 960px

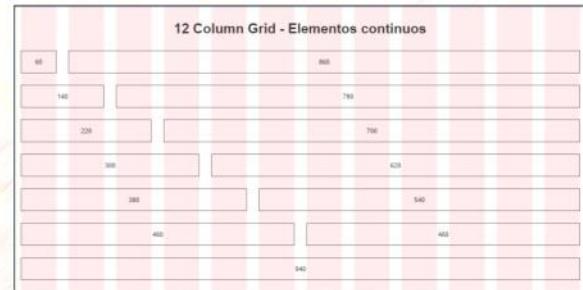
HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

Implementación con una rejilla (1)

El framework incluye varias clases contenedoras, para diversos números de columnas; e.g. container_12 para disponer de 12 columnas.

Se definen una serie de clases correspondientes a cada posible ancho en número de columnas

```
.container_12 .grid_1 {width: 60px}  
.container_12 .grid_2 {width: 160px}  
.....  
.container_12 .grid_12 {width: 940px}
```



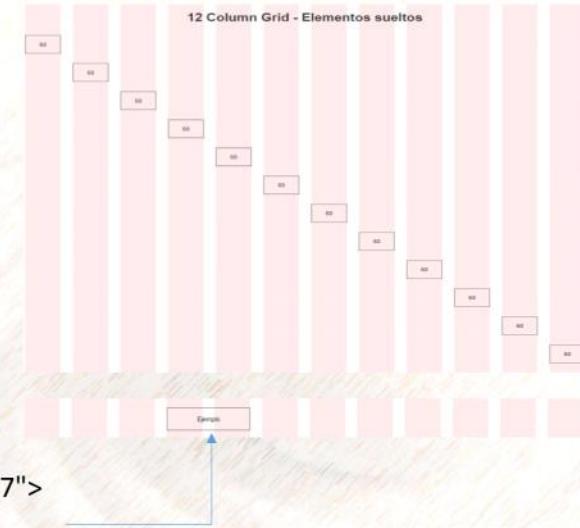
```
<div class="container_12">  
  <div class="grid_12">Some content</div>  
  <div class="grid_8">Some content</div>  
  <div class="grid_4">Some content</div>  
  <div class="grid_12">Some content</div>  
</div>
```

Implementación con una rejilla (1)

Otros estilos nos permiten crear de forma aislada elementos de cualquier ancho y posición , sin abarcar en una línea todo el ancho

```
.container_12 .prefix_1 {  
    padding-left: 80px;}  
.container_12 .prefix_2 {  
    padding-left: 160px;}  
....  
.container_12 .suffix_1 {  
    padding-right: 80px;}  
.container_12 .suffix_2 {  
    padding-right: 160px;}
```

```
<div class="grid_2 prefix_3 suffix_7">  
    <p>Ejemplo</p>  
</div>
```



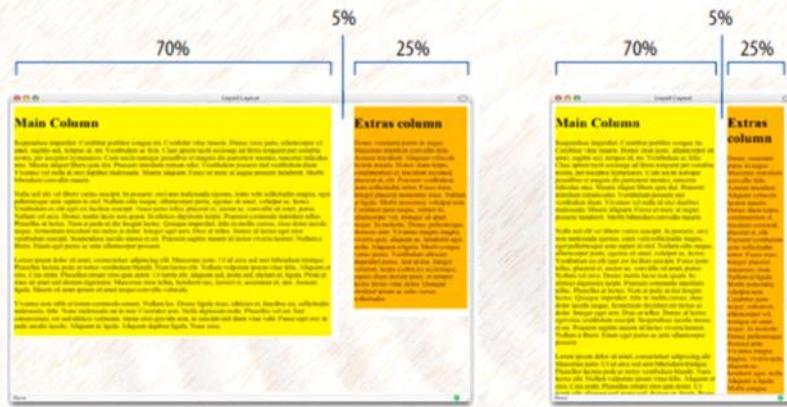
Disposición fluida

lunes, 5 de junio de 2017 11:30

AC3

Disposiciones fluidas

- Se respeta el flujo por defecto como esquema de posicionamiento de los componentes utilizando *float* y *clear* para las modificaciones adecuadas
- Se especifican las anchuras de los contenedores en porcentaje y se omiten cuando es posible, dejando que se aplique el valor por defecto *auto*
- Para evitar tamaños excesivos se utilizan las etiquetas *min-width* y *max-width*.



HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

Rejilla



Columnas de texto



- column-count** → número de columnas
- column-width** → ancho de las columnas (alternativa al anterior)
- column-gap** → separación entre las columnas
- column-rule** → borde lateral (filete) de las columnas): width style, color

Origen de las hojas de estilo

```
-moz-column-count: 3;
-webkit-column-count: 3;
-moz-column-width: 200px;
-webkit-column-width: 200px;
-moz-column-gap: 20px;
-webkit-column-gap: 20px;
```

El CSS es un lenguaje de estilos empleado para definir la presentación, el formato y la apariencia de un documento de marcado, sea html, xml, o cualquier otro. Commonmente se emplea para dar formato a los documentos en formato html que funcionan como aplicaciones web. También puede ser empleado en formatos xml, u otros tipos de documentos de marcado para la posterior generación de documentos.

Las hojas de estilos surgen de la necesidad de diseñar la información de tal manera que podemos separar el contenido de la presentación y, así, por una misma fuente de información, generalmente destinada a un lenguaje de marcado, ofrecer diferentes presentaciones en función de los dispositivos, servicios, contextos o aplicativos. Por lo que un mismo documento html, mediante diferentes hojas de estilos, puede ser presentado por pantalla, por impresora, por lectores de voz o por tabletas touch. Separamos el contenido de la

forma, composición, colores y fuentes. La especificación del CSS la mantiene el World Wide Web Consortium (W3C).

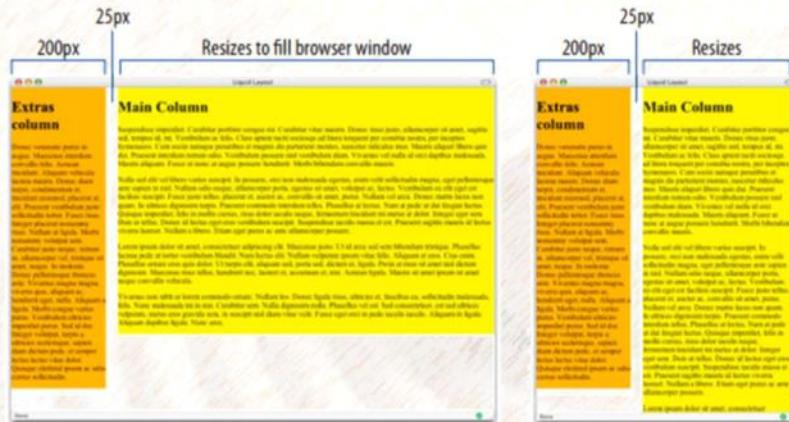
Las diferentes revisiones de las hojas de estilo tienen el origen en la primera especificación publicada por el W3C en diciembre de 1996 y que pretendía unificar la sintaxis y el modo de definir una hoja de estilos por los diferentes lenguajes derivados del SGML. A partir de esta primera especificación, en mayo de 1999, y como una extensión a la primera revisión, aparece la segunda, popularmente conocida como la especificación CSS2, y adoptada por la mayoría de los navegadores. Como características fundamentales que añade la revisión, se encuentra la posibilidad de definir posiciones de forma absoluta, relativa y fija, así como la profundidad de los elementos (z-index). cuando existen superposiciones, también el

soporte para formatos de voz y textos bidireccionales. De esta, aparece una revisión, la 2.1, que incorpora muchas de las mejoras hechas por los fabricantes de navegadores y que actualmente es respaldada y estandarizada.

La tercera revisión de la especificación del CSS por el W3C empieza en el 2005 y todavía está en proceso de definición. Pero esta vez, las diferentes implementaciones de los motores de renderizar los navegadores no están esperando a tener una especificación, sino que implementan ciertas cosas a su manera y, por lo tanto, muchas son utilizables en entornos de producción web. Es necesario, sin embargo, recordar que la especificación de que los diferentes implementaciones de los navegadores no es exacta, y de que, por lo tanto, cuando desdoblamos una hoja de estilos, el principio que tiene que regir es que funcione en todas partes, y no que funcione igual.

Disposiciones híbridas

- Mezcla de áreas de tamaño fijo y escalable
- Es frecuente que el primero corresponda a barras laterales (*side-bars*) menús etc.
- Como consecuencia de la mezcla los cálculos se hacen más complicados



Modelos de posicionamiento.

lunes, 19 de febrero de 2018 11:37

Conceptos previos.

Posicionamiento *static*

Variaciones en el flujo. Elementos flotantes. Float. Clear.

1. Modos de Posicionamiento. Esquemas.

1. Posicionamiento relativo
2. Posicionamiento absoluto

2. Aplicación del posicionamiento. Position.

1. Posiciones relativas
2. Posiciones absolutas y fijas
3. Superposiciones.



Posición de los elementos

Existen varios esquemas de posicionamiento

- Flujo normal
- Posicionamiento relativo
- Posicionamiento absoluto

Flujo normal

- cada elemento “bloque” inicia una nueva línea, colocándose inmediatamente debajo del elemento anterior
 - aunque indiquemos el ancho de un bloque el siguiente empezara debajo, incluso si tiene sitio para colocarse a continuación
 - es el esquema por defecto

Section 2

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit
in voluptate velit.

Posicionamiento relativo

- Un elemento se desplaza **a partir de la posición que le correspondería** en el modelo anterior en función de los valores de las propiedades *top*, *right*, *bottom* o *left*
- Este desplazamiento no afecta a la posición de los elementos circundantes, se mantienen en la posición que les corresponde en el flujo normal, como si el elemento desplazado siguiera en su posición original.

Lorem Ipsum

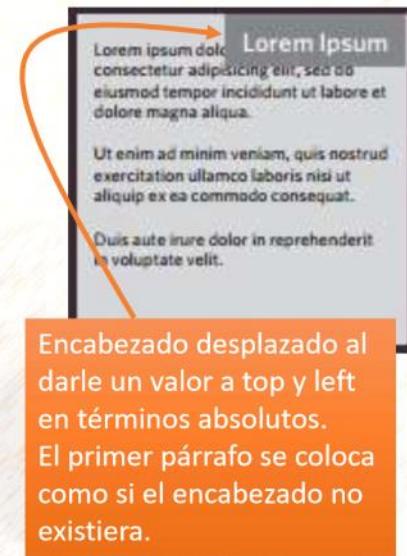
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea. Duis aute irure dolor in reprehenderit in voluptate velit.

Párrafo desplazado al darle un valor a top y left

Posicionamiento absoluto

- El elemento **se posiciona respecto a su contenedor**, completamente al margen del flujo normal
- El resto de los elementos no se ven afectados, sino que ignoran el espacio ocupado por aquel que se ha posicionado de forma absoluta, comportándose como si el elemento no existiera.
- Los elementos posicionados absolutamente se mueven cada vez que los usuarios desplazan la ventana del navegador (*scroll*)



Posición de los elementos

position

static: el esquemas de posicionamiento corresponde al flujo normal (valor por defecto)

relative: al usar este esquema se pueden indicar los desplazamientos necesarios a partir de posición natural de los elementos

absolute: este esquema permite indicar posiciones exactas respecto body o al elemento contenedor (parent) que haya sido marcado como no static

fixed: es una variación del esquema absoluto, en la que se indica el valor de la posición respecto a la ventana del navegador

top right bottom left

Si posición es “relative”, indican los valores de desplazamiento (offset) respecto a la posición natural

Si la posición es “absolute” (o “fixed”) indica los valores absolutos, medidos desde el borde indicado

Superposiciones

Al utilizar el posicionamiento relativo fijo o absoluto, los elementos se pueden superponer

Por defecto se respeta el orden en que se describen los elementos en el código HTML, siendo el último el que queda por encima

z-index

Permite indicar el orden de los elementos superpuestos, que irá desde los valores más altos o los valores más bajos.
Es el equivalente a administrar las capas en un programa de edición de 'gráficos'



Posición absoluta y porcentajes

```
.caja2 {  
    height: 50%;  
    width: 50%;  
    background-color: #CCF;  
    position: absolute;  
    left: 25%;  
    top: 23%;  
}
```

Aunque la posición se define como absoluta

Valores de tamaño
Valores de desplazamiento

Se pueden expresar en porcentaje

Se obtiene así una posición fija, pero adaptada a los diferentes tamaños que pueda tener la pantalla

Ejemplo



[./Curso_Web/HTML/14.Posiciones/Posiciones_div_abs_porcentaje.html](#)

Modos de presentación

Lunes, 19 de febrero de 2018 11:37

Conceptos previos.

Modos de display básicos Cambios de modos (línea/bloque).

Display *none* y Visibilidad;

1. Modelos de presentación alternativos. Etiqueta display.
2. *Display Tables*.
3. *Flex-Box*
4. *Display Grid*.



Modelos de display

domingo, 14 de mayo de 2017 13:17

Etiqueta display

display

Modelos de presentación alternativos.

inline
block
inline-block

run-in (block o inline según el contexto)
none

list-item
table

table-caption
table-column-group
table-header-group
table-footer-group
table-row-group
table-cell
table-column
table-row
inline-table

flex
list-item
grid

Display Table

miércoles, 10 de mayo de 2017 18:59

Display tipo tabla

Se puede utilizar para controlar el comportamiento de contenedores de diferente altura

```
<div id="contenedor">
  <div id="contenidos">
    <div id="columna1"></div>
    <div id="columna2"></div>
    <div id="columna3"></div>
  </div>
</div>
```

los elementos `<div>` que forman las columnas de la página en realidad se comportan como celdas de tabla, lo que permite que el navegador las muestre con la misma altura.

```
#contenedor {
  display: table;
}
#contenidos {
  display: table-row;
}
#columna1,
#columna2,
#columna3 {
  display: table-cell;
}
```



FlexBox

¿QUÉ ES FLEXBOX? → un **módulo completo de layout**

defino cómo se muestran los elementos y cómo se relacionan con el resto.

- CONTENEDOR
- ITEMS

El **contenedor** va a poder modificar las dimensiones y el orden de los **items** para acomodarlos según nuestras indicaciones.

¿Qué cosas podemos hacer con flexbox que antes no podíamos hacer de manera fácil?

- Alineación vertical
- Columnas de igual altura independientemente del contenido
- Cambiar al orden en el que se MUESTRAN los elementos sin que cambie el HTML

FlexBox



`display:flex`

define un “**contenedor flexible**”

`display:inline-flex`

El contenedor
sería inline
respecto al resto
de los elementos
de la pagina

convierte de forma automática a sus “hijos”
directos en “**elementos flexibles**”

sus tamaños y disposición se ajustará siempre
al espacio disponible en el contenedor

Contenedor
flexible

Items
“flexibles”

Aportaciones de FlexBox



Aplicaciones conocidas básicas:

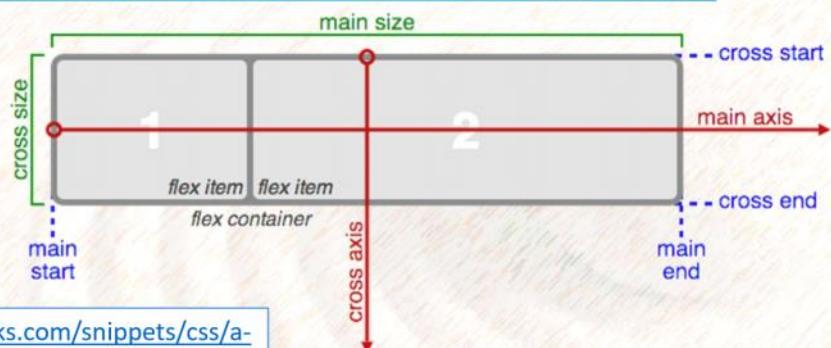
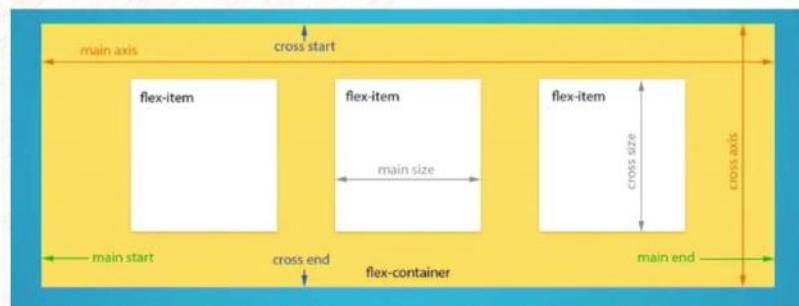
- Alineación vertical de elementos
- Columnas de igual altura
- Grid flexibles
- Cambiar orden de elementos
- Holy Grail Layout
- Sticky footer ...

Una alternativa
específicamente diseñada
para llevar a cabo estas
funcionalidades más
fácilmente.

Otras aplicaciones:

- Items tipo app
- Layouts en zig-zag
- Graficáis sencillas
- Inputs con iconos Q
- Headers con múltiples zonas
- Filtros de igual tamaño con número
desconocido de elementos

Conceptos de FlexBox



<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Codepen interactivo con un resumen de todas las propiedades que afectan tanto al flex-container como a los items:
<http://codepen.io/diana aceves/pen/a4becd2ed04b6e3974bc06d1fa12360b>

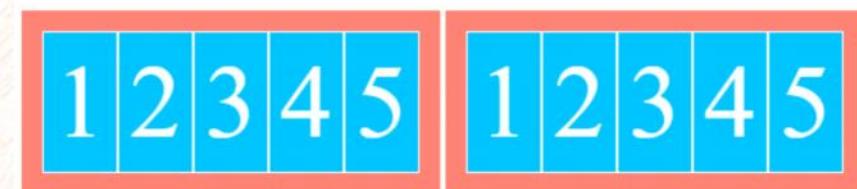


Contenedor flexibles

display:flex



display:inline-flex



Contenedor flexibles (1)

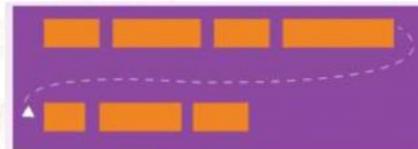


flex-direction



Row – row reverse – colum – colum reverse

flex-wrap



Nowrap – wrap – wrap-reverse

Ambos se pueden
resumir en **flex-flow**

Contenedor flexibles (2)



justify-content

flex-start – **flex-end** – **space-between** –
center – **space-around**

Alineación en el eje principal y reparto del espacio que queda

align-items

flex-start – **flex-end** – **center** – **baseline** – **stretch**:

Alineación en el eje secundario (o transversal) entre ellos

align-content

flex-start: - **flex-end**: - **center**: - **space-between**: - **space-around**: - **stretch**:

Alineación de FILAS respecto del eje transversal
(Importante!!! Solo con más de una línea)



Items flexibles

order <entero>



Row – row reverse – colum – colum reverse

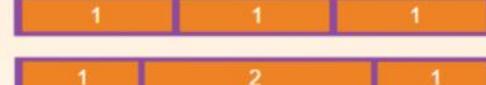
flex-grow

flex-shrink

flex-basis

Se pueden
resumir en **flex**

align-self



flex-grow (default 0): cuanto crecerá el cuando hay espacio
flex-shrink (reducción, default 1) cuanto decrecerá el elemento cuando no hay espacio
flex-basis (default = 'main-size'): tamaño inicial del elemento

<https://ed.team/blog/guia-definitiva-de-flexbox-2-flex-basis-flex-grow-flex-shrink>

Display Grid

domingo, 25 de febrero de 2018 12:56

Sistema de rejilla en dos dimensiones



Similitud con flex-box

Contenedor: `Display grid / Display inline-grid`



Los hijos directos del elemento pasan a ser `grid items`

Display ms-grid en Internet Explorer

display: -ms-grid;

Actualmente exclusivo de IE

Se define el número de filas y columnas indicando la relación de anchura / altura que se distribuirán entre ellas

Contenedor

**display: -ms-grid;
-ms-grid-rows: 1fr 2fr;
-ms-grid-columns: 1fr 1fr**

La segunda fila el doble de alta
Las 2 columnas igual de anchas

Elementos

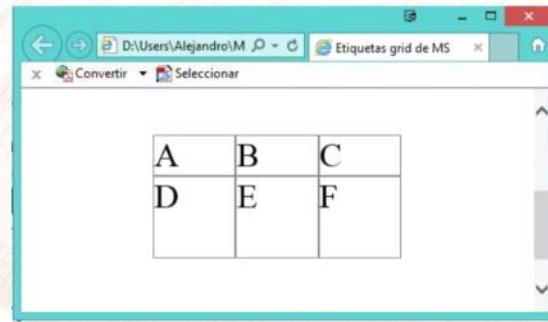
-ms-grid-row: 1; -ms-grid-column: 2;

En cada elemento se indica
que columna y fila ocupara (el
primero es por defecto 1 1)

Ejemplo

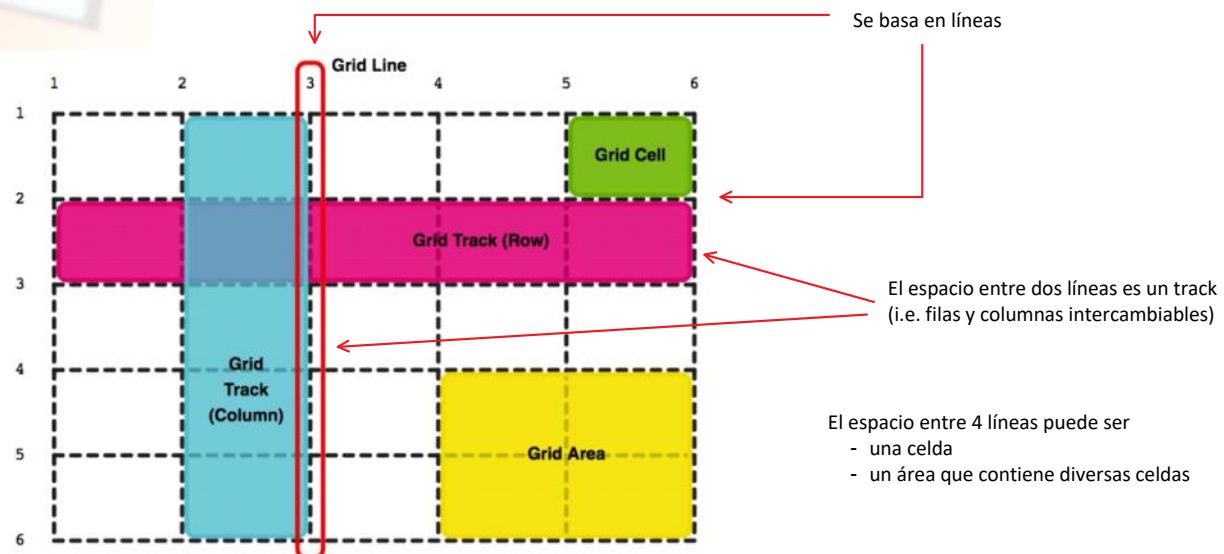
```
section#ejemplo2 > div { display: -ms-grid;
    -ms-grid-rows: 1fr 2fr 1fr;
    -ms-grid-columns: 1fr 1fr 1fr; }
section#ejemplo2 > div > div {border: 1px solid gray;}
section#ejemplo2 > div > div:nth-child(2) {-ms-grid-row: 1; -ms-grid-column: 2; }
section#ejemplo2 > div > div:nth-child(3) {-ms-grid-row: 1; -ms-grid-column: 3; }
section#ejemplo2 > div > div:nth-child(4) {-ms-grid-row: 2; -ms-grid-column: 1; }
section#ejemplo2 > div > div:nth-child(5) {-ms-grid-row: 2; -ms-grid-column: 2; }
section#ejemplo2 > div > div:nth-child(6) {-ms-grid-row: 2; -ms-grid-column: 3; }
```

```
<section id = "ejemplo2">
    <div>
        <div>A</div>
        <div>B</div>
        <div>C</div>
        <div>D</div>
        <div>E</div>
        <div>F</div>
    </div>
</section>
```



Conceptos básicos

domingo, 25 de febrero de 2018 13:02



Creamos el *grid* EXPLÍCITAMENTE.

Al contenedor GRID, le indico cómo deben ser las filas y las columnas:

Existen múltiples sintaxis y formas de hacerlo

```
.grid{  
  display: grid;  
  grid-template-columns: 30% 30% 30%;  
  grid-template-rows: 1fr 2fr 1fr  
}
```

Defino el ancho de cada columna y cada fila en cualquier unidad de medida

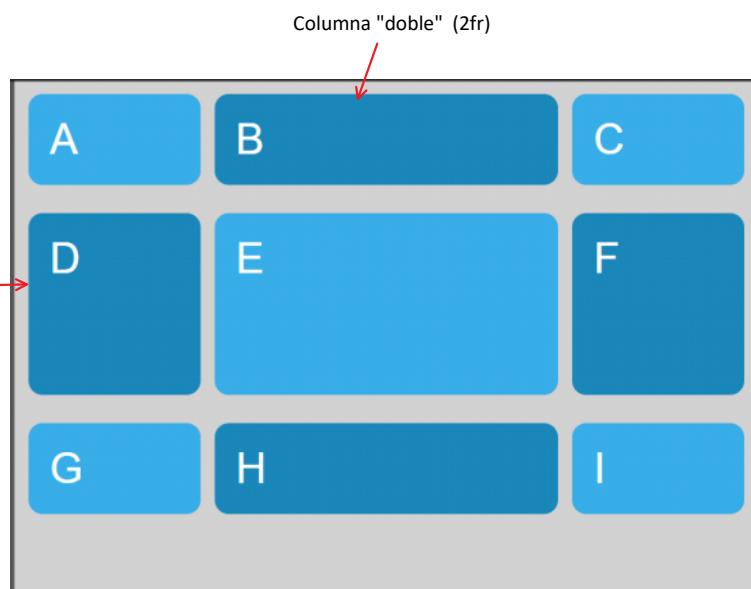
Lo más habitual es el uso de unidades de fracción (fr) que indica en qué proporción se reparte el espacio

Ejemplo

```
<div class="grid">  
  <div class="grid-item a">A</div>  
  <div class="grid-item b">B</div>  
  <div class="grid-item c">C</div>  
  <div class="grid-item d">D</div>  
  <div class="grid-item e">E</div>  
  <div class="grid-item f">F</div>  
  <div class="grid-item g">G</div>  
  <div class="grid-item g">H</div>  
  <div class="grid-item g">I</div>  
</div>
```

```
.grid{  
  display: grid;  
  grid-column-gap: 1rem;  
  grid-row-gap: 2rem;  
  grid-template-columns: 1fr 2fr 1fr;  
  grid-template-rows: 1fr 2fr 1fr  
}
```

La propiedad --gap permite definir la separación entre columnas



El resto del CSS se limita al aspecto, sin ninguna particularidad relativa a GRID

```
body{  
  font: 3rem sans-serif;  
  background-color: rgba(0,0,0,.2);  
  color: white;  
}  
  
.grid-item:nth-child(2n+1){background-color: #1A93BE;}
```

El resto del CSS se limita al aspecto, sin ninguna particularidad relativa a GRID

```
background-color: #ff0000; color: white; }  
.grid-item:nth-child(2n){ background-color: #1A83B5; }  
.grid-item:nth-child(2n+1){ background-color: #37A8E0; }  
.grid{ padding: 1rem; }  
.grid-item{ padding: 1.5rem; border-radius: 1rem; }
```

1. Conceptos de RWD
 - a. Media queries
 - b. Pantallas y *viewport*
 - c. Responsivo responsable. Performance
 - d. Navegación responsiva
2. *Frameworks*. Bootstrap 4



PUNTOS FUNDAMENTALES:

- LAYOUT FLEXIBLE : Unidades relativas para (casi) todo.
- MEDIA QUERIES
- ETIQUETA VIEWPORT <meta name="viewport" content="width=device-width, initial-scale=1 ">
- IMÁGENES: max-width, svg, srcset, picture...

Diseño web adaptable (Responsive Web Design, RWD)

Disposiciones fluidas o elásticas no son suficiente: hay que añadir otras técnicas con objeto de adaptar el sitio web al entorno del usuario.

Tiene su origen en el diseñador y autor norteamericano **Ethan Marcotte**, a partir de una serie de artículos en la publicación en línea *A List Apart*, y de su libro *Responsive Web Design* (2011).

intangibles con los que el diseño web debe negociar

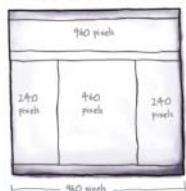
ventanas de ancho inconsistente
resoluciones de pantalla
preferencias de los usuarios
fuentes instaladas

mayor cantidad de dispositivos, modos de entrada y navegadores con la que nunca

Convertir en fluido

ancho del elemento
ancho del contenedor

ancho en %



LAYOUT FLEXIBLE: EM vs REM

EM: se refiere al tamaño de fuente establecido EN EL PROPIO SELECTOR y está afectado por la HERENCIA.

¿Cuándo EM?

- Cuando quiera que un valor (height, line-height, padding, margin...) crezca o disminuya proporcionalmente con un cambio de font-size en el elemento o en sus ancestros.
- En MEDIA QUERIES: Siempre se refiere al font-size del NAVEGADOR

<https://zellwk.com/blog/media-query-units/>

Diapositiva 116

AC2 CSS3 y Javascript Avanzado Jordi Collalt Puig Universitat Oberta de Catalunya, 2013 Algoritmos Generativos, 2013/2014

Media Types

Desde la versión 2.1 de los CSS, existe la posibilidad de definir estilos en función del uso de la hoja:

- para la pantalla (screen)
- para la impresión (print)
- para los lectores de voz (voice)

```
<link rel="stylesheet" media="screen">
  href="screen.css" />

<style>
  @media screen {
    .class { background: #ccc; }
  }
</style>
```

La mayoría de los navegadores y los dispositivos nunca se han adherido al espíritu de la especificación, dejando varios *media types* implementados imperfectamente, o completamente ignorados.

AC2

Media Queries



CSS3 permite incluso definir estilos específicos para diferentes tamaños de pantalla.

```
<link rel="stylesheet"
      media="screen and (max-device-width: 480px)"
      href="screen.css" />
```

La query contiene dos componentes:

1. Un *media type (screen)*,
2. la consulta entre paréntesis, conteniendo una característica a inspeccionar (*max-device-width*) seguida por el valor al que apuntamos (480px).

Definiremos así una hoja de estilos específica para dispositivos con una pantalla pequeña como los Smartphone

Diapositiva 117

AC2

CSS3 y Javascript Avanzado
Jordi Collell Puig
Universitat Oberta de Catalunya, 2013
Requerido - Cerrado, 29/03/2014

Nuevas Media Queries

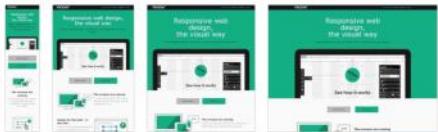
```
@media (min-resolution)
@media (-webkit-min-device-pixel-ratio)
```

Level 4 Media Queries

- @media (pointer) → detecta la precisión del dispositivo apuntador: *coarse*, *fine* o *none*
- @media (hover) → detecta si el dispositivo apuntador soporta el estado de *hover*
- @media (script) → detecta el soporte de JavaScript
- @media (luminosity) → detecta la luminosidad ambiental: *dim*, *normal* o *washed*.

Ejemplos

FRONT



International Polar Foundation



Ejemplos <http://mediaqueri.es/>

Directivas *import* y *media*: múltiples hojas de estilo

@import

- permite incorporar hojas de estilo (como link)

```
@import url("tables.css");
```

en este caso se pueden añadir *media queries* que filtraran la importación

```
@import url("shetland.css") screen and  
(max-device-width: 480px);
```

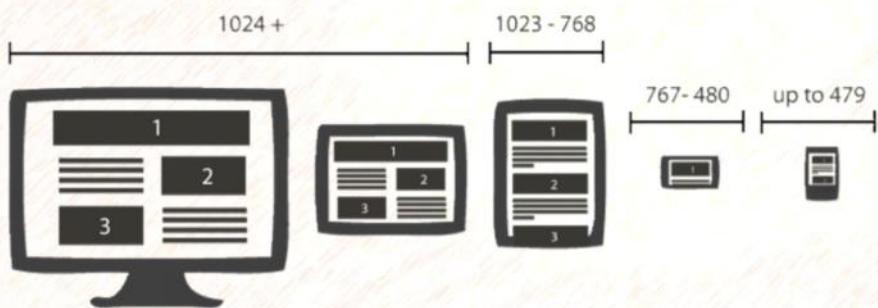
@media

- permite aplicar las *media queries* desde dentro de los estilos

```
@media screen and (max-device-width: 480px) {  
    .column {  
        float: none;  
    }  
}
```

Diapositiva 120

Tamaños de pantalla



Responsive Web Design



Tamaños de pantalla y mediaqueries (1)

escritorio / tablet horizontal: más de 1024px	
@media (min-width: 1024px)	Default [@media (min-width: 1024px)]
tablet vertical, mas de 768px (y menos de 1024px)	
@media (min-width: 768px)	@media (max-width: 1023px) and (min-width: 768px)
móvil horizontal, más de 480px (y menos de 768px)	
@media (min-width: 480px)	@media (max-width: 767px)
móvil vertical, menos de 480px	
Defatut [@media (max-width: 479px)]	@media (max-width: 480px)

Diseño bottom-up o mobile first

Diseño top-bottom

Tamaños de pantalla y mediaqueries (2)

escritorio grande: más de 1200px	
@media (min-width: 1200px)	Default [@media (min-width: 1200px)]
escritorio normal: mas de 1024px (y menos de 1200px)	
@media (min-width: 1024px)	@media (max-width: 1199px) and (min-width: 1024px)
tablet horizontal más de 768px (y menos de 1024px)	
@media (min-width: 768px)	@media (max-width: 1023px) and (min-width: 768px)
móvil horizontal / tablet vertical más de 480px (y menos de 768px)	
@media (min-width: 480px)	@media (max-width: 767px) and (min-width: 480px)
móvil vertical, menos de 480px	
Defatut [@media (max-width: 480px)]	@media (max-width: 480px)

Diseño bottom-up o mobile first

Diseño top-bottom

Viewport

Por defecto el viewport está definida a 960px: el navegador supone que este es el ancho de la web

Existe el metadata viewport en el head, solo válido para dispositivos móviles, que permite cambiar el valor por defecto y el comportamiento del navegador

```
<meta name="viewport" content="">
```

Content admite diversos parámetros con su valor, separados por comas

```
<meta name="viewport"  
content="width=device-width,  
initial-scale=1,  
maximum-scale=1">
```

width (min-width - max-width)
height (min-height' - max-height)
zoom
min-zoom
max-zoom
user-zoom
orientation
initial-scale
maximum-scale

Pruebas de Web Mobile

Emulador
es de
iPhone

<http://iphone4simulator.com/> (no escala)

<http://www.testiphone.com/> (no escala)

<http://iphone5csimulator.com/> (no escala)

<http://ipadpeek.com/> (no escala)

Emuladores
de iPhone /
Android /...

<http://www.mobilephoneemulator.com/> (escala) **

<http://mobiletest.me/> (no escala) **

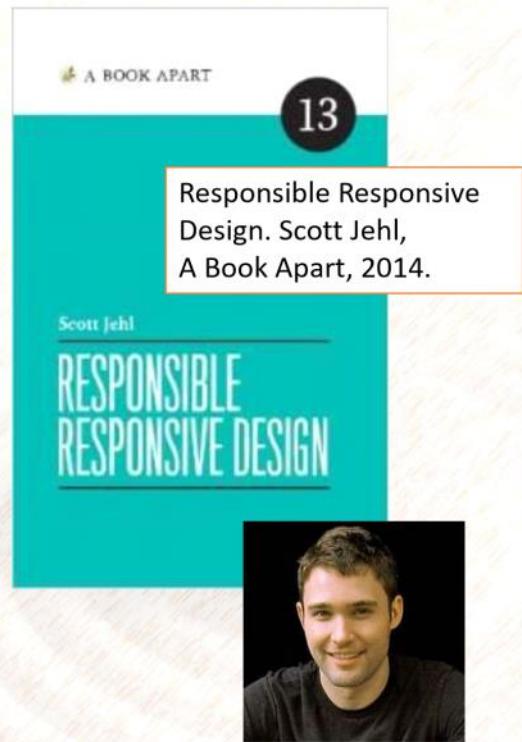
<http://transmog.net/iphone-simulator/> (no escala)



Diseño “Responsive” Responsable (1)

El diseño debe adecuarse a las características de los dispositivos móviles más allá del simple tamaño de la pantalla.

- **Usabilidad**
- **Accesibilidad**
- **Sostenibilidad**
(Sustainability)
- **Rendimiento**
(Performance:)



Diseño “Responsive” Responsable (2)

- Usabilidad** → El interfaz de usuario en un dispositivo móvil debe adecuarse a los requerimientos del usuario, teniendo en cuenta las especificidades del medio
- Accesibilidad** → Tener en cuenta estándares de accesibilidad y tecnologías *asistivas*
- Sostenibilidad
(Sustainability)** → El tiempo: la retro-compatibilidad como opción de futuro ligada a nuevas situaciones (e.g. Opera Mini parece un navegador antiguo, sin JS)
- Rendimiento
(Performance:)** → El rendimiento de cara a entornos móviles de bajo costo y/o en países emergentes como elemento crítico.

HTML5: Responsive navigation



WebCat, 13 de febrero de 2013. ([vimeo](#))
Javier Usobiaga - RWD: dealing with navigation

<https://vimeo.com/62146014>

“Patrones de navegación”

[Brad Frost: Responsive Navigation Patterns](#)

- Top Nav or “Do Nothing” Approach
- The Footer Anchor
- The Select Menu
- The Toggle
- The Left Nav Flyout
- The Footer Only
- The “Hide and Cry”

[Brad Frost: Complex Responsive Navigation Patterns](#)

- The Multi-Toggle
- The Ol’ Right to Left
- The ‘Skip the Sub-Nav’
- The Priority+
- Off-Canvas Flyout
- The Carousel+

<http://bradfrost.com/blog/web/responsive-nav-patterns/>

<http://bradfrost.com/blog/web/complex-navigation-patterns-for-responsive-design/>

Frameworks UI adaptables

Junes, 5 de junio de 2017 11:43

YUI

<http://yuilibrary.com/>



HTML and CSS: Designing and Building Web Sites
Jon Duckett
John Wiley & Sons. 2011

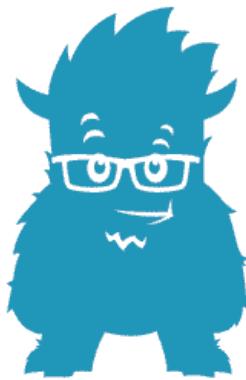
Skeleton

<http://www.getskeleton.com/>

Skeleton

Bootstrap

<http://getbootstrap.com/>



Foundation (ZURB)

<http://foundation.zurb.com/>

Material Design

<https://material.io/>



Mejoras gráficas

lunes, 19 de febrero de 2018 11:37

1. Fondos con Sprites y rollovers
2. CSS Dinámico
 - a. Transiciones
 - b. Transformaciones
 - c. Animaciones



Sprites

sprite → Conjuntos de imágenes, generalmente iconos, que se almacenan en un solo archivo



background-position

Permite extraer cada uno de los iconos



```
viv#icono1 {background-position: 0px 0px;}  
div#icono2 {background-position: 0px 64px;}  
div#icono3 {background-position: 128px 64px; }
```

Rollovers

botones que cambian
a una segunda forma

las diferentes posibilidades se
almacenan en un sprite

En respuesta al evento al
que ligamos el cambio de
botón (:hover, :active)
cambia background
position, de modo que en
cada caso se ve una parte
de la imagen, la adecuada
al evento producido



```
a.cms:hover{background-position: 0px -127px;}  
a.modules:hover{background-position: -160px -127px;}  
a.framework:hover{ background-position: -320px -127px; }
```

CSS Dinámico

Lunes, 19 de febrero de 2018 13:59

AC16

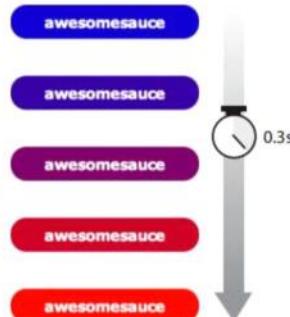
CSS Dinámico



- Transition
- Transformation
- Animation

CSS3 incorpora la posibilidad de efectos animados que hasta el momento sólo eran posibles con Flash o JavaScript

Ejemplo de cómo cambia un elemento de la página html al aplicarle una transición



Learning Web Design, Fourth Edition
Jennifer Niederst Robbins
O'Reilly Media, 2012

AC17

transition

permite suavizar los cambios (*tweening*), creando automáticamente el resto de los pasos que se encuentran implícitos en un cambio de estado (e.g. cambio de color, cambio de fuente...)

Desarrollado inicialmente para webkit por el equipo responsable de Safari; Actualmente es un borrador en el W3C, soportado por la mayoría de los navegadores, siempre que se utilicen sus correspondientes prefijos

- webkit-transition...
- mozilla-transition...
- op-transition...
- ms-transition...

El proceso de transición
suele aparecer ligado a una
pseudoclase como :hover,
:focus, or :active

```
a.smooth {  
    transition----}  
a.smooth:hover, a.smooth:focus {  
    background-color: red;  
}
```

transition: propiedades

Propiedades implicadas

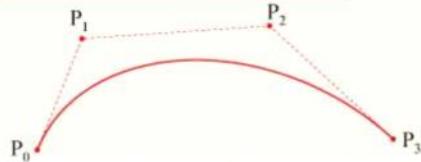
transition-property

propiedad sobre la que se creara la transición (e. g. una transformación)
Con el valor “all” se aplica a cualquier cambio

transition-duration

tiempo en segundos entre la posición final y la inicial

transition-timing-function



forma de la transición, en base a una curva Bézier: *ease*, *linear*, *ease-in*, *ease-out* o *ease-in-out*; *cubic-bezier(#,#,#,#)*, *steps(#, start/end)*, *step-start*, *step-end*

transition-delay

retardo en segundos hasta que comienza la transición

transition: atajo (shorthand)

transition

en su forma de atajo, la propiedad puede recibir los cuatro parámetros sucesivamente

transition: <propiedad> <tempo>s <función> <retraso>s;

a.smooth {
 ...
 background-color: mediumblue;
 -webkit-transition: background-color 0.3s ease-in-out 0.2s;
 -moz-transition: background-color 0.3s ease-in-out 0.2s;
 -o-transition: background-color 0.3s ease-in-out 0.2s;
 -ms-transition: background-color 0.3s ease-in-out 0.2s;
 transition: background-color 0.3s ease-in-out 0.2s;
}
a.smooth:hover, a.smooth:focus {
 background-color: red;
}

Ejemplo

Ejemplo



transition: ejemplo

```
#trans {  
    display: block;  
    width: 500px;  
    margin: 50px auto;  
    padding: 15px;  
    text-align: center;  
    border: 1px solid #999999;  
    background: rgb(221,221,221);  
    box-shadow: 0px 0px 0px 0px black;  
    -moz-transition: all 1s ease-in-out 0.5s;  
    -webkit-transition: all 1s ease-in-out 0.5s;  
    transition: all 1s ease-in-out 0.5s;  
}  
#trans:hover {  
    background: rgb(255, 106, 0);  
    border: 3px solid red;  
    border-radius: 20px;  
    box-shadow: 10px 10px 2px 2px black;  
}
```

AC20

transform

transform puede operar cuatro transformaciones básicas en un elemento:

- Escalar: scaleX(x) , scaleY(y), scale(x,y)
- rotar rotate(grados: deg)
El centro de rotación se puede modificar con transform-origin:
percentage | length | left| center| right| top| bottom
- inclinar skewX(deg) skewY(deg) skew(deg, [deg])
- trasladar o mover translateX(x), translateY(y), translate(x,y)

```
#principal:hover{  
-moz-transform: rotate(5deg);  
-webkit-transform: rotate(5deg);  
}
```

Se pueden aplicar a la vez diversas transformaciones

AC21

transform + transition

Una vez creada cualquier transformación, se le pueden aplicar transiciones para crear un proceso de cambio más suave

```
#principal {  
    display: block;  
    width: 500px;  
    margin: 50px auto;  
    padding: 15px;  
    text-align: center;  
    border: 1px solid #999999;  
    background: #DDDDDD;  
    -moz-transition: -moz-transform 1s ease-in-out 0.5s;  
    -webkit-transition: -webkit-transform 1s ease-in-out 0.5s;  
}  
#principal:hover{  
    -moz-transform: rotate(5deg);  
    -webkit-transform: rotate(5deg);  
}
```

AC22

Transformaciones en 3D

perspective

Valor numérico asignado a un contenedor para indicar la distancia al eje Z de los elementos contenidos.
Valores razonables entre 300 y 1500

perspective-origin

Permite modificar el origen de la perspectiva

backface-visibility

Permite hacer visible el reverso

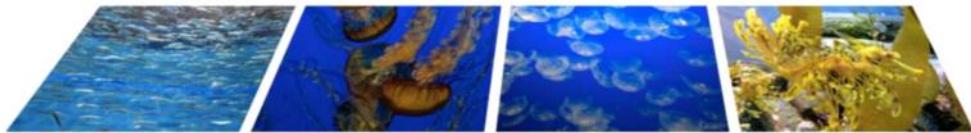
Establecida la perspectiva se aplica una transformación con alguna de las funciones 3D disponibles

- translate3d
- translateZ
- scale3d
- scaleZ
- rotate3d
- rotateZ
- matrix3d.

Transformaciones en 3D: ejemplo

```
ul {  
width: 1000px;  
height: 100px;  
list-style-type: none;  
padding: 0;  
margin: 0;  
-webkit-perspective:  
600;  
-moz-perspective: 600;  
perspective: 600;  
}
```

```
li {  
float: left;  
margin-right: 10px;  
-webkit-transform:  
rotateX(45deg);  
-moz-transform: rotateX(45deg);  
transform: rotateX(45deg);  
}
```



AC24

Animaciones: keyframe animations

@keyframes

Definimos la serie de etapas que constituyen la secuencia de animación

La posición que ocupa cada etapa se define en términos de % respecto a la duración completa de la serie

```
@keyframes animation-name {  
    keyframe { property: value; }  
    keyframe { property: value; }  
}
```

```
@keyframes colors {  
    0% { background-color: red; }  
    20% { background-color: orange; }  
    40% { background-color: yellow; }  
    60% { background-color: green; }  
    80% { background-color: blue; }  
    100% { background-color: purple; }  
}
```



Animaciones: keyframe animations

animation-name	→ nombre de la secuencia de animación
animation-duration	→ tiempo en segundos
animation-timing-function	→ forma de aceleración
animation-iteration-count	→ número de veces que se repite
animation-direction	→ posible dirección adelante y/o atrás
animation-play-state	→ estado reproduciéndose o en pausa
animation-delay	→ tiempo hasta que se inicia
animation-fill-mode	→ límites en los que puede ser aplicada

Animaciones complejas

viernes, 12 de mayo de 2017 13:20

AC27

Animaciones: ejemplos



Pure CSS3 At-At Walker
by Anthony Catzadilla
(www.anthonycalzadilla.com/css3-ATAT/)



CSS3-Man
by Anthony Catzadilla
(www.optimum7.com/css3-man/)



Star Wars Intro recreation in CSS3
by Guillermo Esteves
(www.gesteves.com/experiments/starwars.html)

Pure CSS3 At-At Walker by Anthony Catzadilla
(www.anthonycalzadilla.com/css3-ATAT/)

CSS3-Man by Anthony Catzadilla
(<http://www.optimum7.com/css3-man/animation.html>)

Star Wars Intro recreation in CSS3 by Guillermo Esteves
(www.gesteves.com/experiments/starwars.html)

Animaciones: entornos

Existían herramientas gráficas que facilitan la creación de animaciones CSS generando todo el código (HTML y CSS) necesario

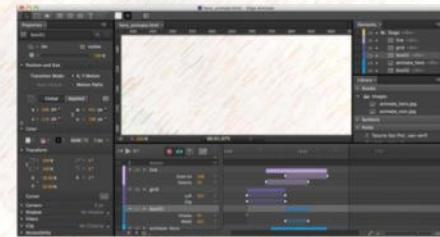
Sencha Animator

www.sencha.com/products/animator/



Adobe Edge

html.adobe.com/edge/animate/



Actualmente se integran en
plataformas más completas

Sencha www.sencha.com/products/

SUPPORT FORUM DOCS RESOURCES BLOG CONTACT US SEARCH

Sencha Platform
Comprehensive platform for web application development – deliver data-intensive, cross-platform web applications that drive competitive advantage for your business.

Education & Support Test
Develop Design Implementation & Customization Services

Enterprise Workflow Integration

Rapidly design, develop, and test cross-platform web applications

The Sencha Web Application Development platform simplifies the challenges of managing the software development lifecycle of web applications. Now you can seamlessly design, develop, and test data-intensive web applications and deliver the right user experience, on the right screen, at the right time.

Sencha Platform for Web Application Development

The Sencha portfolio of products and services forms an integrated, modular platform for designing, developing, and testing your cross-platform web applications. Sencha products can be deployed separately or together to form an end-to-end solution. With the Sencha Platform you can prototype, develop, theme, debug, and test your web applications on any device running any browser.

PLATFORM COMPONENTS

- Ext JS
- Sencha Test
- GXT
- Cmd
- Plugins
- Architect
- Themer
- Inspector

Adobe Animate <http://www.adobe.com/es/products/animate.html>

The screenshot shows the official product page for Adobe Animate CC. At the top, there's a navigation bar with links for 'Creatividad y diseño', 'Marketing y analítica', 'Documentos y firmas electrónicas', 'Soluciones empresariales', 'Asistencia técnica', and a 'Iniciar sesión' button. Below the navigation is a secondary menu with 'ADOBE ANIMATE CC' as the active tab, followed by 'Novedades', 'Aprendizaje y asistencia', 'Prueba gratuita', 'Más productos', and a prominent blue 'Comprar ahora' button. The main content area features a large image of a character from a video game or animation. Text overlay on the image reads: 'La nueva era de la animación: cualquier plataforma, cualquier dispositivo y cualquier estilo'. Below this, a descriptive paragraph states: 'Diseña animaciones interactivas con herramientas de dibujo vanguardistas, publicalas en diferentes plataformas, como Flash/Adobe AIR, HTML5 Canvas, WebGL o incluso para plataformas personalizadas, y llega a usuarios de televisión o de prácticamente cualquier ordenador de sobremesa o dispositivo móvil.' A link '¿Eres nuevo en Animate? Adquiere las destrezas básicas >' is also present. At the bottom of the main content area is a blue horizontal bar with the text 'Consigue Animate como parte de un plan de aplicación única de Creative Cloud por solo 60,49 €/mes IVA incluido.' and another 'Comprar ahora' button.

Animaciones impresionantes

Descubre cómo el conjunto de herramientas de animación vectorial líder puede ayudarte a trabajar más rápido y a crear un contenido fantástico en todas las plataformas, desde los dispositivos móviles hasta las pantallas de ultra alta resolución.