

# CZZ



body { font: x-smal background: " color: black; margin: 0; padding: e;



 CSS es un lenguaje usado para definir la presentación de un documento
 estructurado escrito en HTML o XML.

Una hoja de estilo es un conjunto de normas de estilo que le dicen a un navegador cómo tiene que presentar un documento HTML a los usuarios.

Permite especificar cosas como el tamaño,
 el color, el tipo de fuente, el espaciado, la
 ubicación de texto e imágenes en la página,
 márgenes, etcétera



#### - CSS 1

- publicada en diciembre 1996, y abandonada en abril de 2008
- CSS 2

CSS

publicada como recomendación en mayo de

```
1998, y abandonada en abril de 2008

background:url("images/bk_body.png");
    font-family: Arial, Geneva, Verdana; 2.1
  目body {
9

    publicada como recomendación oficial el 7 de

10
                                       junio de 2011
11
12
    =#container{
                                 – CSS 3 -> Actual
13
      width:800px;
      margin:0 auto;
14
 15
                                    < style.css >
 16
```

## Reglas y hojas de estilo

 Una regla define algún aspecto del estilo de uno o varios elementos.

h1 { color: green; }

 Una hoja de estilo es un conjunto de reglas que se aplican a un documento HTML.

# Anatomía de una regla

```
h1 { color: green; }

SELECTOR DECLARACIÓN
```

- -El selector lo utilizamos para elegir a qué elementos se aplica el estilo.
- La declaración es la parte de la regla que especifica qué efecto tendrá ésta (es decir, el estilo)
- En el ejemplo, todos los elementos <h1> se verán afectados por la declaración (aparecerán en verde)

## Declaración

Una declaración de css esta compuesta de 2 elementos:

- Propiedades: Identificador que nos indica que propiedad vamos a afectar
- Valor: Indica el valor que vamos a modificar la propiedad seleccionada

color: Green;

Background-color: Red;

## Agrupando reglas y selectores

- Una de las metas de CSS era la brevedad
  - Facilita escribir la hoja de estilo "a mano"
  - Reduce el tiempo de carga
- Por ejemplo, sean las siguientes reglas:

```
h1 { font-weight: bold; }
h2 { font-weight: bold; }
h3 { font-weight: bold; }
```

Como las declaraciones son idénticas, se pueden agrupar:

```
h1, h2, h3 { font-weight: bold; }
```

## Agrupando reglas y selectores

 Un selector puede tener más de una declaración:

```
h1 { font-weight: bold; }
h1 { color: green; }
```

 Podemos agrupar las declaraciones en una lista separada por puntos y comas:

```
h1 { font-weight: bold;
    color: green; }
```

# Incluir la hoja de estilo en el documento

Para que la hoja de estilo afecte al documento HTML, hay
 que enlazar éste de alguna manera a la hoja de estilo

#### - Varias formas:

- Incrustar la hoja de estilo en el documento, con el elemento <style>
- Aplicar estilo a un elemento individual usando el atributo style
- Enlazar una hoja de estilo externa al documento, a través del elemento <link>

## Formas de usar CSS

 Un estilo en línea (inline) es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML

Hola mundo

## Atributo "style"

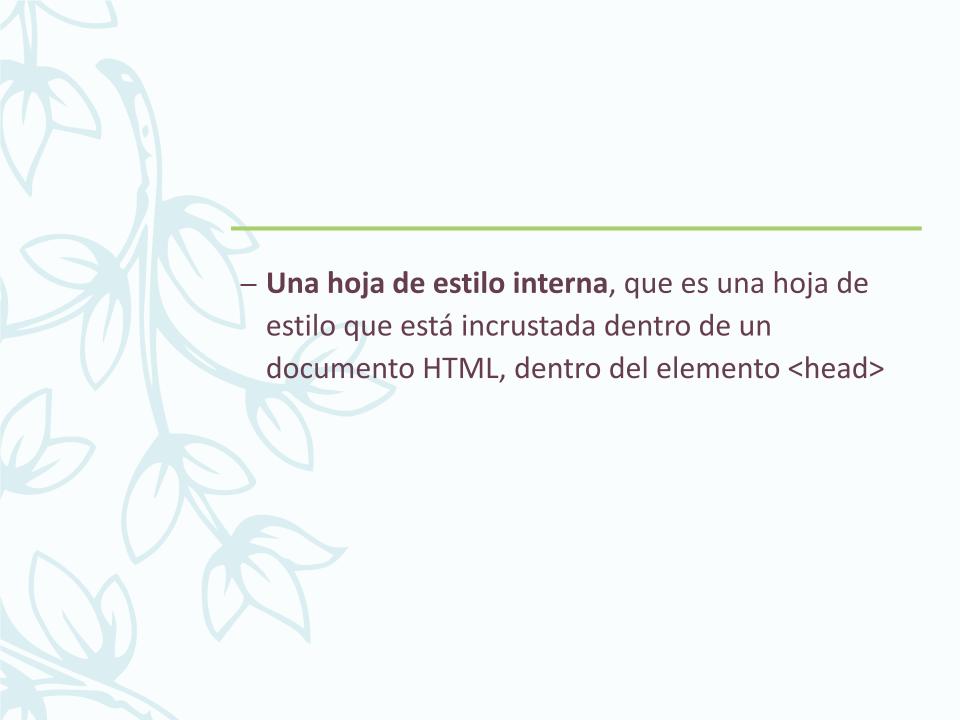
 Todos los elementos de HTML tienen un atributo style que permite añadir estilo para un elemento en particular.

<h1 style="color: blue;">Estilos en línea</h1>

- El estilo afecta sólo a ese elemento
- Un estilo local asignado con este atributo redefinirá cualquier otro estilo aplicado al documento

# Atributo "style"

- El uso de este atributo elimina la mayoría de las ventajas del uso de hojas de estilo:
  - No se puede reutilizar el estilo aplicado a un elemento (habría que repetirlo en todos los sitios)
  - Si luego se cambia el estilo, hay que buscar todos los atributos style y cambiar su valor
- Sólo tendría sentido para algo muy concreto y que no queramos cambiar la hoja de estilo sólo para eso



## Elemento "style"

Se puede meter el estilo en el propio documento HTML

```
<html>
<head>
  <title>...</title>
  <style type="text/css">
  <!--
    h1 { color: green; }
  </style>
</head>
<body>
```

Problema: sólo afecta a un documento HTML

## Hojas de estilo externas

- Una hoja de estilo externa, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web
- Una hoja de estilo externa no es más que un archivo de texto normal que contiene una serie de reglas
- Por convenio, llevan la extensión .css
- ¿Cómo se enlaza al documento HTML?
  - Mediante el elemento < link>

```
<! DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!--Google Webmaster-->
<meta name="google-site-verification"</pre>
content="2AUbZzsdsIgyaQMMml6c6GVpJ98NIXx3GbMorPJBDxM" />
<title>Albuquerque Web Design + Print Design - Roady Studios</title>
<link href="css/rs5.css" rel="stylesheet" type="text/css">
                                                          HTML
<!--Starting Google Analytics-->
<script type="text/javascript">
 var _gaq = _gaq || [];
 _gaq.push(['_setAccount', 'UA-11755188-1']);
 _gaq.push(['_trackPageview']);
 (function() {
   var ga = document.createElement('script'); ga.type =
'text/javascript'; ga.async = true;
   ga.src = ('https:' == document.location.protocol ? 'https://ssl' :
'http://www') + '.google-analytics.com/ga.js';
   var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
 3)();
</script>
</head>
```

```
body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-
  font-size: 100%:
  line-height: 1.5;
  color: #000;
  background-image: url('../images/bkgd.jpg');
#logo-header {
  width: 100%;
  text-align: center;
#logo-header img {
  max-width: 80%;
  height: auto:
#descrip-header {
  width: 100%;
  text-align: center;
.worktile img {
  margin: 10px;
  box-shadow: Opx Opx 7px #aaa;
  -moz-box-shadow: Opx Opx 7px #aga;
  -webkit-box-shadow: Opx Opx 7px #aaa;
}
```

## Elemento link>

```
<html>
<head>
<title>...</title>
k rel="stylesheet" type="text/css" href="miestilo.css"/>
</head>
<body>
<html>
</html>
```

# Ventajas de las hojas de estilo externas

#### Reutilización

- Poner toda la información de estilo en un sitio permite que sea referenciada por muchos documentos
- Facilita mantener un sitio Web grande consistentemente
  - Información de estilo corporativa
- Rendimiento
  - Una vez que se descarga la hoja de estilo la primera vez,
     el navegador ya la guarda en la caché

## Selectores

- Los selectores son los que nos indican a que elementos se le va aplicar el estilo definido.
- Los podemos agrupar en diferentes categorías:
  - Selectores Simples
  - Atributos
  - Pseudo-clases
  - Pseudo-elementos
  - Selectores múltiples
  - Combinadores

# Selectores Simples

- Los selectores de tipo o elemento se declaran colocando el nombre del elemento html a los que se le quiere aplicar el estilo.
  - TAG
  - -ID
  - CLASS

## Selectores simples

### Etiqueta

Si queremos elegir una etiqueta,
 simplemente escribimos su nombre.

– Por ejemplo, si queremos establecer las propiedades para los enlaces: a { ... }  Se puede especificar elementos HTML específicos (TAG's).

```
body
{
    background-image:url('../Fondo.jpg');
    background-repeat:no-repeat;
    background-attachment:fixed;
}
```

# Selectores simples

2. Elemento único (ID)

Podemos elegir un elemento único utilizando su atributo id.
 Para ello, empleamos el simbolo #:

# menu { ... }

El selector de id se utiliza para especificar un estilo para un único elemento

## Selectores simples

#### 3. Definir una clase

 Podemos definir una clase y hacer que muchos elementos la utilicen, escribiendo un punto antes del nombre.

Por ejemplo, si queremos destacar algo:

.importante{ ... }

Párrafo

#### El selector class

El selector class se utiliza para especificar un estilo para un grupo de elementos. A diferencia del selector de id, el selector class es la más utilizada en varios elementos

Esto le permite definir un estilo particular para muchos elementos HTML con la misma class

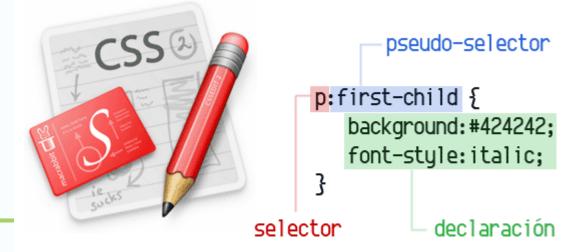
El selector de clase utiliza el atributo class HTML, y se define con un "."

```
.center
{
    color:blue;
    font-family:Arial;
    text-align:center;
}
```

Para especificar clases múltiples en un elemento solo hay que separarlas con un espacio en blanco.

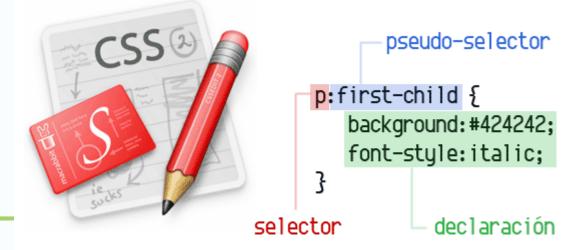
### Selector

```
Regla
Selector
               Propiedad
                            Valor
p{font-size:20px;color:red;text-align:center}
p{
  font-size:20px;
  color: red;
  text-align:center
```



La pseudo clase es una palabra clave precedida del simbolo dos puntos(:) que se agrega a los elementos solo cuando cumplen con cierto estado

- Unos ejemplos de pseudo clases son:
  - visited
  - hover
  - checked
  - enabled
  - disabled



Son muy similares a las pseudo clases solo que estos en lugar de indicar un estatus especial, nos permiten aplicar estilos a ciertas partes del documento. Estos se especifican con un doble dos puntos (::).

- Unos ejemplos de pseudo elementos son:
  - after
  - before
  - first-line
  - enabled

## Combinar tipos de selectores

Los selectores pueden combinarse

– Ejemplo:

```
p.importante { color: red; }
```

## Selectores contextuales

- No sólo debe coincidir el elemento, sino el contexto en el que aparece
- Supongamos que tenemos:

```
h1 { color: red; }
em { color: blue; }
```

– Podríamos añadir esta otra regla:

```
h1 em { color: blue; }
```

## Selectores múltiples

- Se utilizan para asignar una misma regla a múltiples elementos o selectores.
- Para utilizarlos simplemente hay que colocar comas "," entre los selectores.

```
p, li
     { font-size: 1.6em; }
h1, h2, h3, h4, h5, h6
     { font-family: helvetica, 'sans serif'; }
```

## Combinadores

- Se utilizan para combinar varios selectores,
   normalmente en base a su relación con otro
- Ejemplo: A y B, pueden ser cualquier selector.

	Combinador	Selección
	AB	Cualquier elemento que cumpla con <b>A</b> y <b>B</b> simultáneamente.
	АВ	Cualquier elemento que cumpla con <b>B</b> y ademas que sea descendiente de <b>A</b> (puede ser hijo, nieto, etc).
	A>B	Cualquier elemento que cumpla con <b>B</b> y que ademas sea hijo directo de <b>A</b> .
	A+B	Cualquier elemento que cumpla con <b>B</b> y que ademas sea el siguiente hermano de <b>A</b> (el siguiente hijo del mismo padre).
	A~B	Cualquier elemento que cumpla con <b>B</b> y que sea precedido de <b>A</b> . Ambos elementos deben tener el mismo padre, pero <b>B</b> no tiene que ser precedido inmediatamente por <b>A</b> .

## Ejemplos:

```
Agrupar
   #p-001,#p-002,#p-003{font-size:16px}
Anidado
   div {background-color:black;}
   div p {background-color:yellow;}
   div p span {color:blue}
   div p span.required {font-size:20px}
```

## Unidades y valores

- En CSS se manejan diferentes tipos de unidades dependiendo de la propiedad que se vaya afectar.
- Podemos resumirlos en los siguientes grupos:
  - Valores numéricos
  - Porcentajes
  - Colores
  - Coordenadas
  - Funciones

- Generalmente se usan para longitud y tamaño, se utilizan para definir layouts, tipografía, etc.
- Los podemos agrupar de dos formas:
  - Unidades absolutas
  - Unidades relativas

- Las unidades absolutas siempre van a ser del mismo tamaño sin que le afecten otras configuraciones.
  - px: Pixeles
  - mm, cm, in: Milímetros, centímetros o pulgadas
  - pt, pc: Puntos (1/72 de pulgada) o picas (12 puntos)
- Su uso no es tan común, excepto los pixeles

- Las unidades relativas son aquellas cuya influencia es relativa a elementos como el tamaño de fuente o el tamaño del viewport.
  - em: 1em es equivalente al tamaño actual de la fuente( mas especifico, al ancho de la letra mayúscula "M"). En base al tamaño de fuente default que manejan los navegador, podemos decir que 1em es igual a 16 pixeles.
  - ex, ch: Respectivamente equivalen a la altura de la letra minúscula "x" y al ancho del numero '0'.

- rem: Funciona igual que el em, excepto que siempre equivale al tamaño default de la fuente.
- vw, vh: Respectivamente son 1/100 del ancho del viewport y 1/100 del alto del viewport.
- De todos los anteriores el em es el que mas se suele utilizar.
- Son muy útiles ya que a como cambia el tamaño del viewport, conservando el layout de forma correcta.

## Porcentajes

- Estos especifican un tamaño en base a su contenedor padre.
- Este tipo de unidades hacen que los elementos se vayan adaptando al tamaño del layout, conservando el aspecto en dispositivos mas pequeños.

## Colores

- ☐ Existen diferentes formas de declarar colores en CSS.
  - Nombre
  - Hexadecimal
  - RGB
  - HSL
  - RGBA and HSLA

http://www.w3schools.com/cssref/css\_colors.asp

También se puede definir la opacidad de manera independiente



Debido a que un elemento puede ser afectado en diferentes reglas existe un orden en el que los estilos se aplica, a esto se le conoce como cascade.

La forma mas básica de cascade es por el lugar donde se declara la regla de estilo, siendo por orden de importancia:

El orden de prioridad es:

- Default del browser
- 2. Stylesheet externo
- 3. Stylesheet interno
- 4. Estilo inline

Siendo el mayor el prioritario.

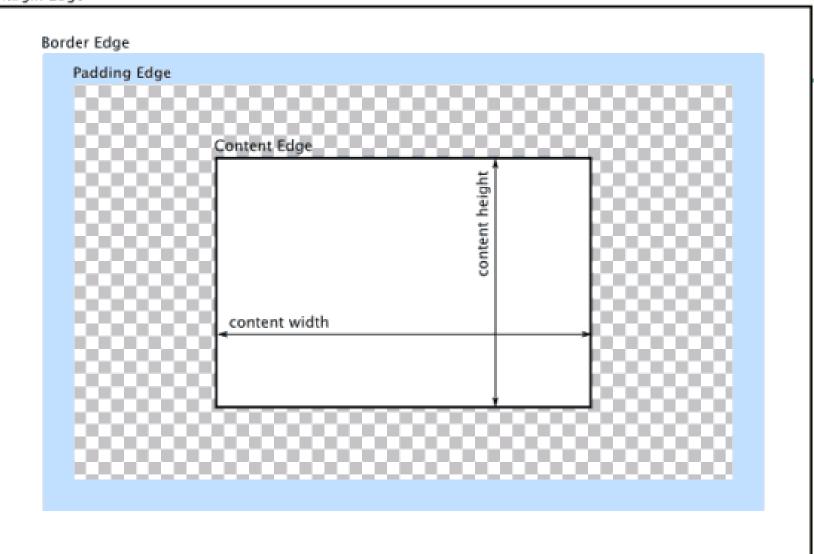
### **BOX MODEL**

- El box model es la base de los layouts en web, cada elemento puede ser representado como una caja rectangular con un contenido, borde, margen y un relleno o padding.
- width and height: Estas propiedades establecen el ancho y alto del content
- padding: Representa un margen interno entre el contenido y el borde de la caja
- border: El borde es una capa que se ubica entre el padding y el margin, por default es 0 lo que lo hace invisible
- margin: El margen representa un área que rodea la caja y es la que "empuja" las otras cajas en el layout

# ///

## **BOX MODEL**

Margin Edge

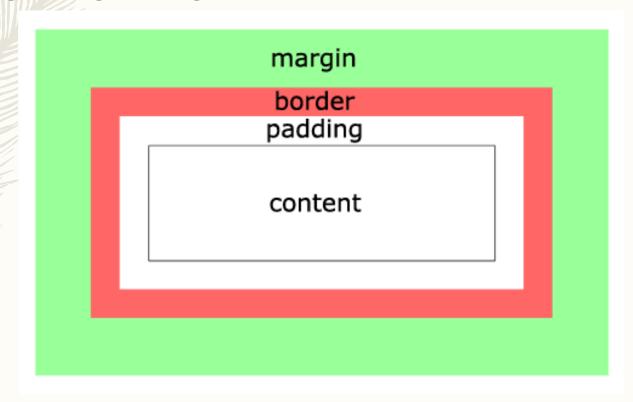


El width de un elemento es normalmente (no IE), SOLO EL CONTENT

En IE, el width incluye padding y border, si no se especifica un doctype

El background color solo afecta del border hacia adentro

ancho real= width + left padding + right padding + left border + right border + left margin + right margin



### Box Model: Overflow

- Cuanto tenemos un elemento con un tamaño fijo con contenido que no quepa en el se utiliza la propiedad overflow para indicar su comportamiento:
  - auto: Si el contenido excede las dimensiones de la caja, el contenido sobrante se oculta y se agregan barras de scroll
  - hidden: Similar a auto, solo que no aparecen los scroll, el contenido permanece oculto
  - visible: El contenido excedente se muestra fuera de la caja (comúnmente el default)

## Box Model: display

- A través de la propiedad display se puede especificar el comportamiento de la caja, por ejemplo block, inline e inlineblock.
- block: Se puede visualizar como si apiláramos cajas, el contenido de cada caja ocupa una línea separada.
- inline: Lo contrario al block, la caja sigue el flujo del contenido así que esta va aparecer rodeada de texto y otras cajas inline. Las propiedades ancho y alto no tienen efecto alguno en este tipo de cajas, cualquier cambio a padding, margin o border no tiene ninguna influencia en las cajas block
- inline-block: Combinación de los dos anteriores, sigue el flujo del texto sin crear nuevas líneas, pero su tamaño puede ser cambiado mediante width y height

## Box Model: box sizing

- Como vimos anteriorermente, el width y el height de una caja solo al content, esto se puede cambiar mediante la propiedad box-sizing.
  - content-box: Comportamiento visto anteriormente, el ancho y alto se calcula mediante la suma de width/height, padding y border
  - border-box: El width y height contiene content, padding y border

## Box Model: box sizing

#### Box Model is content-box

```
Margin 30

Border 5

Padding 15

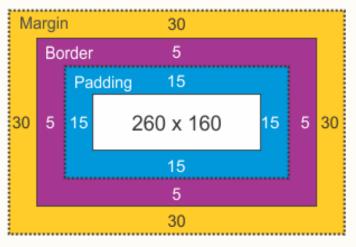
30 5 15 300 x 200 15 5 30

15 5

30
```

```
div{
    width: 300px;
    height: 200px;
    padding: 15px;
    border: 5px solid grey;
    margin: 30px;
    -moz-box-sizing: content-box;
    -webkit-box-sizing: content-box;
    box-sizing: content-box;
}
```

#### Box Model is border-box



```
div{
    width: 300px;
    height: 200px;
    padding: 15px;
    border: 5px solid grey;
    margin: 30px;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}
```

## Posicionamiento

- Mediante la propiedad position podemos cambiar por completo el comportamiento del flujo de elementos en el layout.
- Por ejemplo podemos colocar un elemento flotando en la pagina, o estático en una sola posición.



- Este tipo de posicionamiento le indica al navegador que coloque el elemento en el orden común que sigue el flujo del elemento.
- Se declara mediante el valor "position:static"
- Este es el comportamiento default de todos los elementos

#### Posicionamiento relativo

- Los elementos con el valor "position: relative" son posicionados en su ubicación relativa de acuerdo al flujo del layout, con la peculiaridad que su posición puede ser alterada mediante top, bottom, left y right.
- El contenido restante no ocupara el espacio vació que deje el elemento con posición relativa.

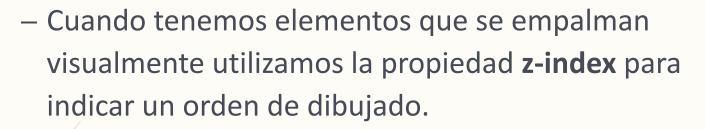
#### Posicionamiento absoluto

- Los elementos con el valor "position: absolute" son posicionados en forma relativa a su ancestro mas cercano, sino cuenta con un elemento ancestro se utiliza el body y el elemento avanza con el scroll de la pagina.
- Este tipo de posicionamiento no afecta el flujo de los demás elementos en el layout.

#### Posicionamiento fijo

- Los elementos con el valor "position: fixed" posiciona los elementos de manera fija en la ventana del navegador
- Este tipo de posicionamiento no influye en el flujo del documento ya que siempre están fijos en una sola posición.

#### z-index



 El valor default es 0 y como este numero vaya aumentando el elemento es dibujo al final.
 Tambien es posible asignar numeros negativos para reducir el orden del dibujado.

http://www.w3schools.com/cssref/playit.asp
?filename=playcss z-index&preval=1

- Propiedad float
- Cuando tenemos elementos que se empalman visualmente utilizamos la propiedad z-index para indicar un orden de dibujado.

## Ejemplos:

- http://attasi.com/
- http://attasi.com/labs/picsselz/
- http://lenticular.attasi.com/#
- http://photon.attasi.com/
- http://attasi.com/refresh/



http://www.w3.org/TR/CSS2/selector.html

http://www.w3.org/

https://developer.mozilla.org/es/

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

http://codepen.io/pen/

http://www.w3schools.com/cssref/default.asp