

Desarrollador de Aplicaciones Web

Programación Web I

1er Cuatrimestre (2018)



Departamento de Ingeniería e Investigaciones Tecnológicas

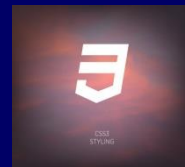


Comisión Miércoles Noche

**Ing. Rubén Goría
Alicia Rosenthal**

Comisión Viernes Mañana

**Ing. Gerardo Barbosa
Ing. Gabriel Panik**



1. Introducción
2. ¿Qué es CSS/ CSS3?
3. Modos de Implementación
4. Atributos de fuente, texto y fondo
5. Layout de un Sitio Web (esquema de distribución de elementos)
6. Nuevas propiedades en CSS3
7. Versiones y sus Compatibilidades



CSS (Cascading Style Sheets)

CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento

*Observemos estas tres páginas web...
igual contenido pero diseños totalmente distintos...
¡Es la magia de CSS!*

Introducción



The Beauty of CSS Design

*Una demostración de lo
que se puede lograr
visualmente usando
diseño basado en CSS-.
Seleccione cualquier
estilo de la lista para
cargarlo en esta página.*

*Baje el archivo de
ejemplo [html](#) y [css](#)*

Y De que se Trata Esto?

Hay claramente una necesidad de que CSS sea tomada seriamente por los artistas gráficos. El Jardin Zen intenta motivar, inspirar, y animar la participacion. Para empezar, vea algunos de los excitantes diseños de la lista. Seleccionando cualquiera de ellos cargara la hoja de estilos en esta misma pagina. El codigo se mantiene

El camino hacia La Luz

Ensuciando un camino oscuro y aburrido reposan las reliquias pasadas de selectores específicos para navegadores, DOMs incompatibles, y soporte incierto para CSS .

Hoy, tenemos que limpiar la mente de prácticas pasadas. La claridad de la Web ha sido alcanzada gracias a los esfuerzos infatigables de gente como W3C, WaSP y los grandes creadores de los navegadores.

El Jardin Zen css invita a usted a relajarse y meditar acerca de las lecciones importantes de los maestros. Comience a ver con claridad. Aprenda a usar (aun por ser) tecnicas honradas por el tiempo en una nueva y vigorosa forma. Conviertase en uno con la Web.

Introducción



CSSOCEANGARDEN

THE BEAUTY OF CSS DESIGN

A demonstration of what can be accomplished visually through CSS based design. Select any style sheet from the list to load it into this page.

Baje el archivo de ejemplo [html](#) y [css](#)

THE ROAD TO ENLIGHTENMENT

Ensuciando un camino oscuro y aburrido reposan las reliquias pasadas de selectores específicos para navegadores, DOMs incompatibles, y soporte incierto para CSS .

SELECT A DESIGN

UNDER THE SEA!

por [Eric Stoltz](#)

MAKE 'EM PROUD

por [Michael McAgthon](#) And
[Scotty Reifsnyder](#)

ORCHID BEAUTY

por [Kevin Addison](#)

OCEANSCAPE

por [Justin Gray](#)

CSS CO., LTD.

por [Benjamin Klemm](#)



Baje el archivo de ejemplo [html](#) y [css](#)

CSS * Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any stylesheet from the list to load it into this page.



* Select a Design

- [Under the Sea!](#)
por [Eric Stoltz](#)
- [Make 'em Proud](#)
por [Michael McAghon and Scotty Reifsnyder](#)
- [Orchid Beauty](#)
por [Kevin Addison](#)
- [Oceanscape](#)
por [Justin Gray](#)
- [CSS Co., Ltd.](#)
por [Benjamin Klemm](#)
- [Sakura](#)
por [Tatsuya Uchida](#)
- [Kyoto Forest](#)

* The Road to Enlightenment

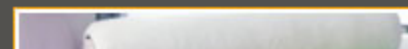
Ensuciando un camino oscuro y aburrido reposan las reliquias pasadas de selectores específicos para navegadores, [DOMs](#) incompatibles, y soporte incierto para [CSS](#).

Hoy, tenemos que limpiar la mente de prácticas pasadas. La claridad de la Web ha sido alcanzada gracias a los esfuerzos infatigables de gente como [W3C](#), [WaSP](#) y los grandes creadores de los navegadores.

El Jardín Zen css invita a usted a relajarse y meditar acerca de las lecciones importantes de los maestros. Comience a ver con claridad. Aprenda a usar (aun por ser) técnicas honradas por el tiempo en una nueva y vigorosa forma. Conviertase en uno con la Web.

* So What is This About

* Participation





Página Web

Estructura

- Párrafos
- Encabezados
- Listas
- Tablas
- Capas
- Etc.

Contenido

- Textos
- Imágenes
- Enlaces

Apariencia

- Colores
- Tipografías
- Alineación
- Fondos
- Tamaños
- Etc.

Comportamiento

- Efectos
- Validaciones
- Automatización

HTML

CSS

Javascript



Breve Historia

Cascading HTML
Style Sheet
(Håkon Wium Lie)

CHSS

SSP

Stream –based Style
Sheet Proposal
(Bert Bos)

CSS
(1995)

Cascading Style Sheets

CSS 1.0 (1996)

CSS 2.0 (1998)

CSS 2.1 (2009)

CSS 3.1 (2011)

W3C

<http://www.w3.org/TR/CSS/>

¿Qué es CSS?



CSS (Cascading Style Sheets)

Las hojas de estilo en cascada permiten separar la estructura de las páginas HTML de su apariencia. Es el diseño por sobre la estructura de la página web.

Ventajas:

- Poder separar el contenido del diseño o presentación
- Fácilmente modificable
- Rediseño más rápido y seguro
- Correctamente implementado reduce el peso de las páginas

Desventajas:

- Una mala implementación puede llevar a un costoso mantenimiento

Listado completo

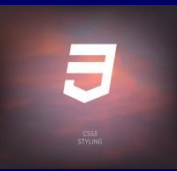
<http://www.w3schools.com/css/default.asp>

¿Qué es CSS3?



CSS (Cascading Style Sheets)

CSS3 constituye un nuevo paradigma, donde HTML se presenta junto con CSS y Javascript como un único instrumento integrado



Sintaxis básica

Las hojas de estilo en cascada se presentan como un conjunto de reglas que responden a la siguiente sintaxis:





Estilos En Línea

1. Como atributo de una etiqueta HTML con el modificador style

```
<html>
  <head> </head>

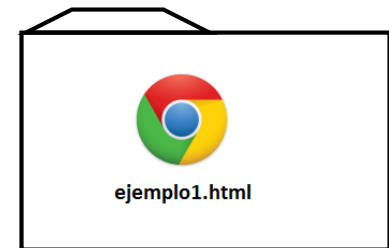
  <body>
    <p style="color:#000099"> Esto es un párrafo de color azul. </p>
    <p style="color:red; font-family:verdana"> Esto es un párrafo de
    color rojo y con tipo de fuente Verdana. </p>
  </body>
</html>
```



Estilos Embebidos

2. En la cabecera del Documento (HEAD)

```
<html>
  <head>
    <style>
      <!--
      H1 {text-decoration: underline; text-align:center}
      P {font-Family:arial,verdana; color: white; background-color: black}
      BODY {color:black;background-color:red; text-indent:1cm}
      // -->
    </style>
  </head>
  <body>
    <p> Texto al que se le aplica CSS por head </p>
    <h1> Así se visualiza la etiqueta H1 </h1>
  </body>
</html>
```





Archivos Externos

3. Como archivo externo .css

- ✓ Todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML conectan mediante la etiqueta `<link>`.
- ✓ Un archivo CSS es un archivo simple de texto, escrito normalmente en el bloc de notas, guardado con la extensión .css
- ✓ Los archivos CSS van dentro de la carpeta del sitio, por buenas costumbres, dentro de una carpeta css que los agrupa
- ✓ Una página Web puede enlazar tantos archivos CSS como sea necesario
- ✓ Un solo archivo CSS puede conectar con todas las páginas Web del sitio para definirle a todas los mismos formatos
- ✓ El mantenimiento del sitio Web se simplifica al máximo, ya que cambiando el css se modifican los estilos de todas las páginas del sitio



Archivos Externos

Se crea en un Editor de Textos, se guarda con extensión .css y se conecta al archivo html a través de la etiqueta link

En HTML5 los estilos por defecto son CSS, por lo tanto no necesitamos agregar ningún atributo en la etiqueta de apertura <style>



Archivos Externos

ATRIBUTOS DE LA ETIQUETA LINK

rel: indica el tipo de relación que tiene el recurso enlazado, para archivos CSS siempre se utiliza el valor ***stylesheet***

href: indica la URL del archivo CSS. Como lo estamos guardando dentro de nuestra PC, dentro de la carpeta “**ejercicio2**”, para realizar la conexión utilizamos el Path relativo “**../ejercicio2/css/estilo.css**”; donde **..** indica la ruta hacia el directorio padre **c:**

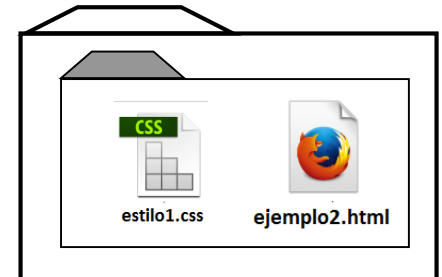
media: indica el medio en el que se van a aplicar los estilos del archivo CSS, en el ejemplo el monitor. A continuación se detalla la lista de medios posibles:



Archivos Externos

ENLACE DESDE EL HTML AL CSS

```
<html>
  <head>
    <link rel="stylesheet" href="estilos.css" >
  </head>
  <body>
    <h1>
      Aplicar un estilo CSS a través de un archivo externo
    </h1>
    <p> Texto al que se le aplica estilos </p>
  </body>
</html>
```

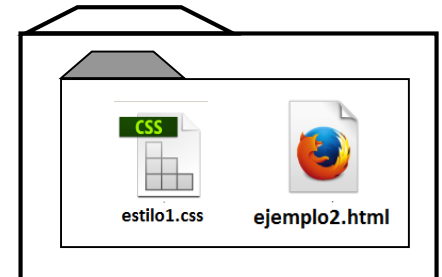


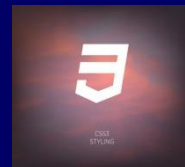


Archivos Externos

ENLACE DESDE EL HTML AL CSS

```
h1 {  
  color: blue; font-family: Algerian;  
  font-size: 1.5 em; line-height: 1.8 em;  
  text-align: center;  
}  
  
p {  
  color: red; font-family: Arial, Helvetica, sans-serif;  
}
```





Medios CSS

| Medio | Descripción |
|------------|---|
| all | Todos los medios definidos |
| braille | Dispositivos táctiles que emplean el sistema braille |
| embosed | Impresoras braille |
| handheld | Dispositivos de mano: móviles, PDA, etc. |
| print | Impresoras y navegadores en el modo "Vista Previa para Imprimir" |
| projection | Proyectores y dispositivos para presentaciones |
| screen | Pantallas de ordenador |
| speech | Sintetizadores para navegadores de voz utilizados por personas discapacitadas |
| tty | Dispositivos textuales limitados como teletipos y terminales de texto |
| tv | Televisores y dispositivos con resolución baja |



Compatibilidad con Navegadores

Algunas propiedades CSS3 se encuentran en estado experimental en algunos navegadores. Para aplicarla efectivamente a los documentos, es necesario declararlas con los correspondientes prefijos. Los prefijos para los navegadores más comunes son los siguientes:

- -moz- para Firefox. •
- -webkit- para Safari y Chrome. •
- -o- para Opera. •
- -khtml- para Konqueror. •
- -ms- para Internet Explorer. •
- -chrome- específico para Google Chrome.



Selectores CSS

Elementos individuales

`p {font-family: Arial, Verdana, sans-serif; font size: 12pt; font-style: italic;}`

Elementos agrupados

`h1, p, b {font-color: red;}`

Clases para elementos

`p.parrafoIndividual{font-color: blue; font-weight: bold;}`

Clases generales

`.especial {font-weight: bold; font size: 10pt; }`

Selectores ID

`#txtCodigo{font-weight: bold; font-family: Arial, Verdana;}`

Selectores Universal (se aplica a todas las etiquetas)

`* { margin: 0px; padding: 0px; }`



Implementación Sugerida

- Archivos de Hojas de Estilos separadas del HTML
- Implementadas por tipo de etiqueta HTML (reutilizando para otras etiquetas)

```
p {  
    background-color:#cc66cc;  
    font-family : Verdana;  
    color : blue;  
}
```

- Implementadas por Categorías (Clases) o en clases para una etiqueta

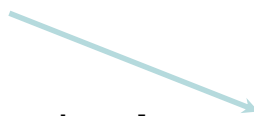
```
.especial  
{  
    background-color:#FFFFFF;  
    font-weight:normal;  
    font-family : Verdana;  
    color : blue;  
    font-size:10px;  
}
```



Combinación de selectores

- Combinando selectores se puede utilizar el mismo valor para el atributo **class** en diferentes elementos pero asignar diferentes estilos para cada tipo.

p.texto1 { font-size : 20px }



Solo se aplica el tamaño de fuente a la **clase .texto1** correspondiente a las **etiquetas p**. Cualquier otro elemento con clase .texto1 que no sea p no se verá afectado.

- La última versión de CSS ha incorporado el **Selector de Atributo**. El **Selector de Atributo** permite referenciar un elemento no solo por los atributos id y class sino también a través de **cualquier otro atributo**. Con la construcción palabra clave [atributo=valor], podemos referenciar un elemento que tiene un atributo particular con un valor específico. Ejemplo:

p[name] { font-size : 20px }

p[name="mitexto"] { font-size : 20px }



Utilización de comodines en selectores

p[name^="mi"] { font-size : 20px}

La regla con el selector ^= será asignada a todo elemento <p> que contiene un atributo name con un valor comenzado en “mi” (por ejemplo, “mitexto”, “micasa”).

p[name\$="mi"] { font-size : 20px}

La regla con el selector \$= será asignada a todo elemento <p> que contiene un atributo name con un valor finalizado en “mi” (por ejemplo “textomi”, “casami”).

p[name*="mi"] { font-size : 20px}

La regla con el selector *= será asignada a todo elemento <p> que contiene un atributo name con un valor que incluye el texto “mi” (en este caso, el texto podría también encontrarse en el medio, como en “textomicasa”).

La misma técnica puede ser utilizada con cualquier atributo y valor que necesitemos. Solo tiene que escribir los corchetes e insertar entre ellos el nombre del atributo y el valor que necesita para referenciar



Pseudo clases en CSS3

```
<div id="contenedor">  
    <p class="mitexto1">Mi texto1</p>  
    <p class="mitexto2">Mi texto2</p>  
    <p class="mitexto3">Mi texto3</p>  
    <p class="mitexto4">Mi texto4</p>  
</div>
```

El div “contenedor” incluye cuatro elementos `<p>` que, considerando la estructura HTML, son hermanos entre sí e hijos del mismo elemento `<div>`. Usando pseudo clases podemos aprovechar esta organización y referenciar un elemento específico

```
p:nth-child(2){ background: #999999; }
```

La pseudo clase `nth-child()` permite encontrar un hijo específico , indicando el número entre los paréntesis. De esta manera se puede acceder a cada elemento pudiendo aplicar distintas reglas.



Pseudo clases en CSS3

La **pseudo clase nth-child()** permite aplicar distintas reglas utilizando palabras clave: **odd** (índice impar); **even** (índice par); **first-child** (primer hijo); **last-child** (último hijo); **only-child** (único hijo).

Ejemplo:

```
p:nth-child(odd){
    background: #999999;
}
p:nth-child(even){
    background: #CCCCCC;
}
```



Nuevos selectores en CSS3

CSS3 también incorpora nuevos selectores que ayudan a llegar a elementos difíciles de referenciar utilizando otras técnicas.

- **Selector >** Este selector referencia al elemento de la derecha cuando tiene el elemento de la izquierda como padre. Por ejemplo, `div > p` referenciará cada elemento `<p>` que es hijo de un elemento `<div>`.
- **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 ~** 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.



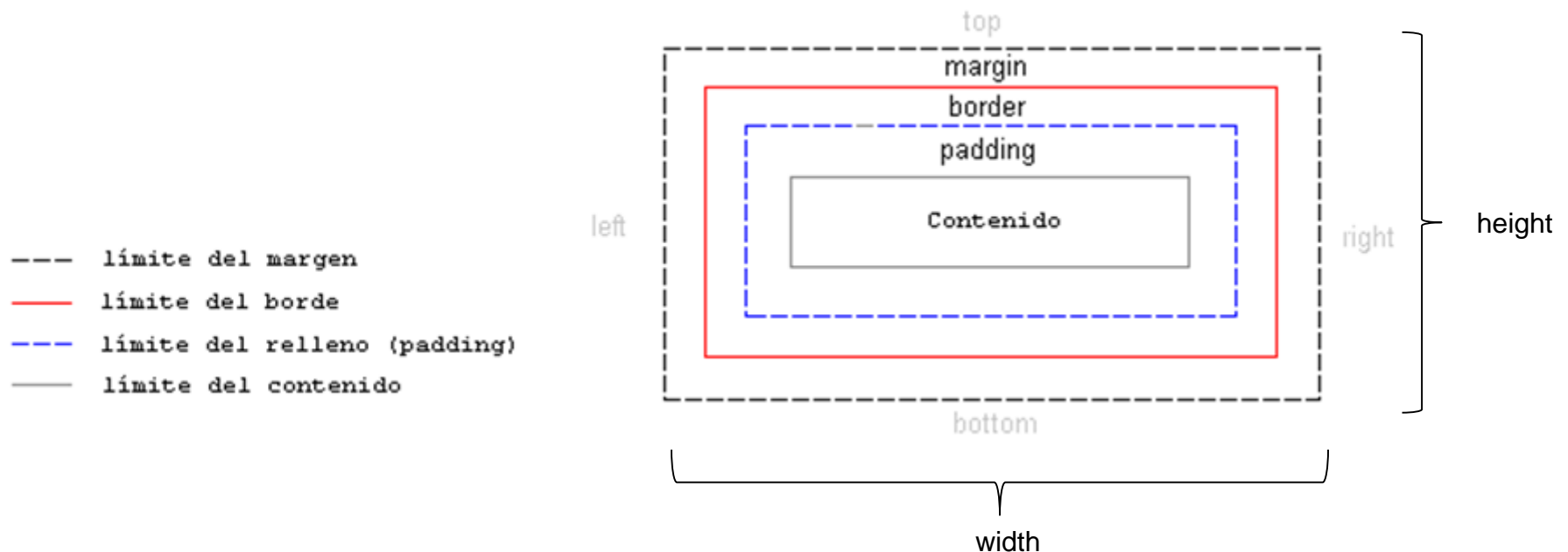
Un poco de historia...

- Inicialmente las **tablas** fueron la herramienta utilizada por desarrolladores para crear y organizar cajas de contenido en la pantalla. Este puede ser considerado el primer modelo de caja de la web.
- Cuando los sitios webs crecieron y se volvieron más y más complejos esta práctica comenzó a presentar serios **problemas relacionados con el tamaño y el mantenimiento del código** necesario para crearlos.
- La incorporación del elemento **<div>** permitió a CSS crear cajas en la pantalla, posicionarlas a un lado o a otro y darles un tamaño, color o borde específico entre otras características. Así surge lo que hoy conocemos como **Modelo de Caja Tradicional**.
- **HTML5** incorpora nuevos elementos que permiten identificar cada **sección del documento** y organizarlo para que sea extensible a distintos tipos de dispositivos de acceso a la WEB. Los mismos se incorporan al Modelo de Caja Tradicional.



Agrupador DIV

En HTML cada elemento es considerado como una caja. El margen (**margin**) es en realidad el espacio alrededor del elemento, el que se encuentra por fuera del borde de esa caja (el estilo **padding**, por otro lado, es el espacio alrededor del contenido del elemento pero dentro de sus bordes, como el espacio entre el texto y el borde de la caja virtual que contiene ese texto).



Propiedad MARGIN

MÁRGENES (Para el body)

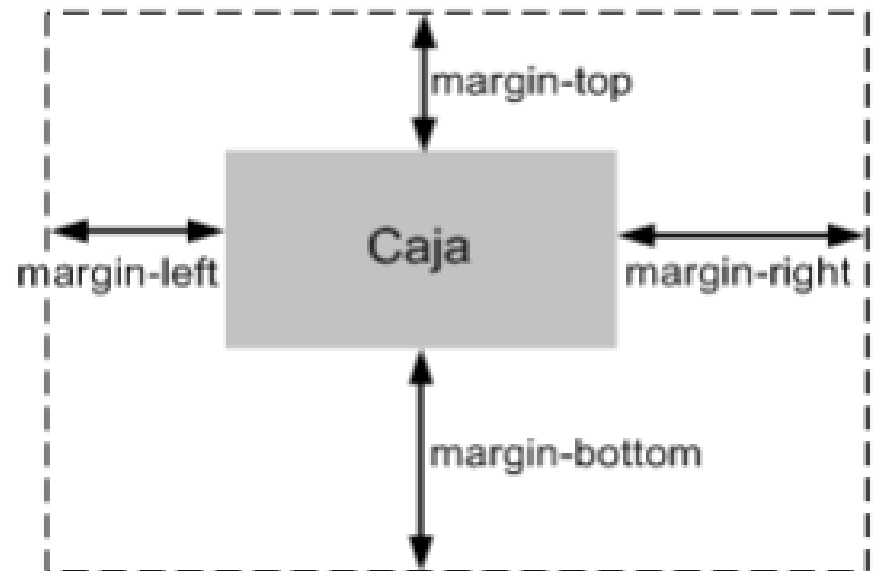
```
body{  
    margin-top:100 px;  
    margin -right: 40 px;  
    margin-bottom:10 px;  
    margin-left: 70 px;  
}
```

O bien

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

```
div {  
    width: 960px;  
    margin: 15px auto;  
}
```

15px auto asigna 15 pixeles al margen superior e inferior del elemento <div> que está afectando y declara como automático el tamaño de los márgenes de izquierda y derecha, serán calculados automáticamente de acuerdo al tamaño del cuerpo del documento y el elemento <div>





Propiedad Padding (relleno)

Es la distancia interior entre el borde y el contenido del elemento.

- Es parecido al atributo cellpadding de las tablas.
- El espacio insertado tiene el mismo fondo que el fondo del elemento-contenedor.
- Al igual que la propiedad margin presenta cuatro variantes (top – right – bottom – left) Se trabaja de igual modo que la propiedad margin.

```
#cabecera {  
    background: #FFFBB9;  
    border: 1px solid #999999;  
    padding: 20px;  
}
```



En la regla mostrada se otorga a `<header>` un fondo amarillo, un borde sólido de 1 pixel y un margen interior de 20 pixeles usando la propiedad padding.



Propiedad box-sizing

Incluye en padding y el borde en el width y height del elemento

- Los valores posibles son: **content-box** | **border-box** | **initial** | **inherit**
- border-box es el valor que permite la inclusión de los valores

Layout de un Sitio Web



Etiquetas correspondientes a cada elemento

Ejemplo de etiquetas de HTML5 que corresponden a las secciones de un documento (codificación). Se ha agregado un contenedor <div> y la propiedad id a las distintas secciones.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS, JavaScript">
    <title>Aquí ponemos el título del documento</title>
  </head>
  <body>
    <div id="agrupar">
      <header id="cabecera">
        <h1> Este es el título principal del sitio WEB</h1>
        <h2> Y este es el subtítulo </h2>
      </header>
      <nav id="menu">
        <ul>
          <li>Opción 1 del Menú de Navegación</li>
          <li>Opción 2 del Menú de Navegación</li>
          <li>Opción 1 del Menú de Navegación</li>
        </ul>
      </nav>
    </div>
  </body>
</html>
```

Layout de un Sitio Web



Etiquetas correspondientes a cada elemento

Ejemplo de etiquetas de HTML5 que corresponden a las secciones de un documento (continuación de la codificación)

```
<section id="seccion">
  <p> Esta es la sección principal </p>
  <article> Artículo 1 </article>
    <figure>
      
      <figcaption> Universidad Nacional
        de La Matanza </figcaption>
    </figure>
  <article> Artículo 2 </article>
  <article> Artículo 3 </article>
</section>
<aside id="columna">
  <p> Esta es la barra lateral </p>
</aside>
<footer id="pie">
  Derechos reservados &copy; 2016
</footer>
```

```
</div>
```

```
<body>
```

```
</html>
```



Propiedad Border

Sus opciones son:

border-width: thin, medium, thick

border-color: en hexadecimal, con su nombre o a través de la función `rgb(; ;)`

border-style: dotted, dashed, solid, double, groove, inset/outset

```
footer {  
    border-width: 1 px;  
    border-style: solid;  
    border-color: blue;  
}
```



En la regla mostrada se otorga a <footer> un borde sólido de 1 pixel de color azul.

o bien

```
footer {  
    border 1px solid blue ;  
}
```



Propiedades Width y Height

Ejemplo

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



Propiedad Float

Permite hacer flotar a la derecha o a la izquierda una caja con su contenido respecto del documento o una caja contendedora mayor

- Los valores a asumir son : **none | left | right | inherit**
- Es recomendable identificar un **width** y **height** a la caja flotante

Cuando se posiciona una caja de forma flotante :

1. La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante.
2. La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.

Layout de un sitio Web



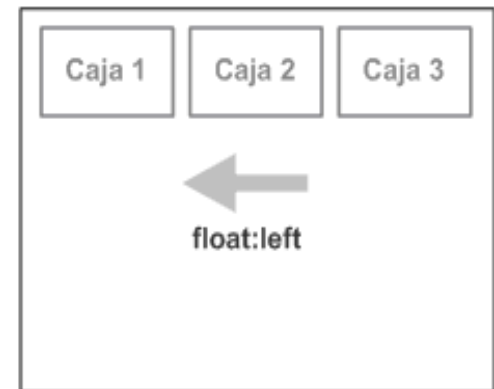
Propiedad Float

Elemento contenedor



Posicionamiento normal

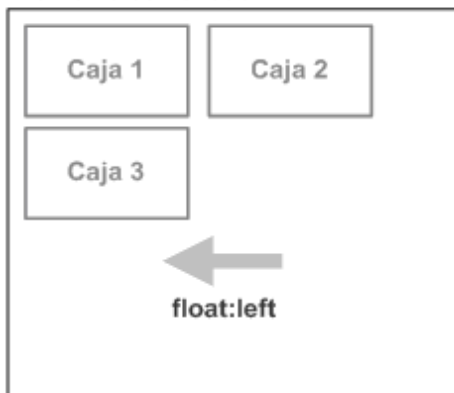
Elemento contenedor



Posicionamiento float de las 3 cajas

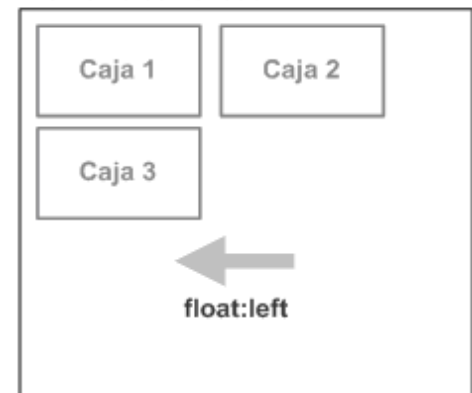
```
div {  
  float: left;  
  width: 300px;  
  margin: 20px;  
}
```

Elemento contenedor



Posicionamiento float de las 3 cajas

Elemento contenedor



Posicionamiento float de las 3 cajas

Propiedad Float

Float puede usarse para ubicar imágenes con texto:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.





Propiedad Clear

La especificación oficial de CSS explica este comportamiento como *"un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento flotante hacia la izquierda"*.

- Indica el lado del elemento HTML que no debe ser adyacente a ninguna caja posicionada de forma flotante.
- Los valores a asumir son: **none | left | right | both | inherit**

```
#pie {  
    clear: both;  
    text-align: center;  
    padding: 20px;  
    border-top: 2px solid #999999;  
}
```




Propiedad Position

Junto con FLOAT, el posicionamiento permite ubicar los elementos en el lugar exacto dentro de la página.

- Para ello se usa un **sistema de coordenadas**.
- Los valores posibles son : **static (defecto) | relative | absolute | fixed**

PARÁMETRO ABSOLUTE

- Se posiciona en forma absoluta y se combina con top, right, bottom y left.
- No deja espacio vacío luego de ser posicionado.

PARÁMETRO RELATIVE

- Se calcula desde la posición original del documento.
- Si se mueve el elemento deja el espacio vacío luego de ser posicionado.



Propiedad Position

Junto con FLOAT, el posicionamiento permite ubicar los elementos en el lugar exacto dentro de la página.

- Para ello se usa un **sistema de coordenadas**.
- Los valores posibles son : **static (defecto) | relative | absolute | fixed**

PARÁMETRO ABSOLUTE

- Se posiciona en forma absoluta y se combina con top, right, bottom y left.
- No deja espacio vacío luego de ser posicionado.



Posición absoluta



Propiedad Position

Junto con FLOAT, el posicionamiento permite ubicar los elementos en el lugar exacto dentro de la página.

- Los valores posibles son :
 - Dentro del Flujo: static (defecto) | relative
 - Fuera del Flujo: absolute | fixed
- Relative, Absolute y Fixed usan un **sistema de coordenadas** basados en top y left.



Propiedad Position

POSITION ABSOLUTE

PARÁMETRO ABSOLUTE

- Se posiciona en forma absoluta y se combina con top, right, bottom y left.
- No deja espacio vacío luego de ser posicionado.



Posición absoluta



Propiedad Position

POSITION ABSOLUTE

PARÁMETRO RELATIVE

- Se calcula desde la posición original del documento.
- Si se mueve el elemento deja el espacio vacío luego de ser posicionado.





Propiedad Z-index

Permite que los elementos se conviertan en capas y puedan superponerse.

- La propiedad z-index se acompaña de un número.
- El elemento de número mayor se superpone al de número menor.

Ejemplo de Z-index

```
#capasuperior{  
    position: absolute;  
    top: 100 px;  
    left: 100 px;  
    z-index: 3;  
}
```

```
#capamedia{  
    position: absolute;  
    top: 90 px;  
    left: 90 px;  
    z-index: 2;  
}  
#capainferior{  
    position: absolute;  
    top: 80 px;  
    left: 80 px;  
    z-index: 1;  
}
```



Propiedad OverFlow

Permite maquetar capas, recortando el contenido de una capa para que quepa según sus dimensiones.

- Los valores posibles son: **visible/hidden/scroll/auto**



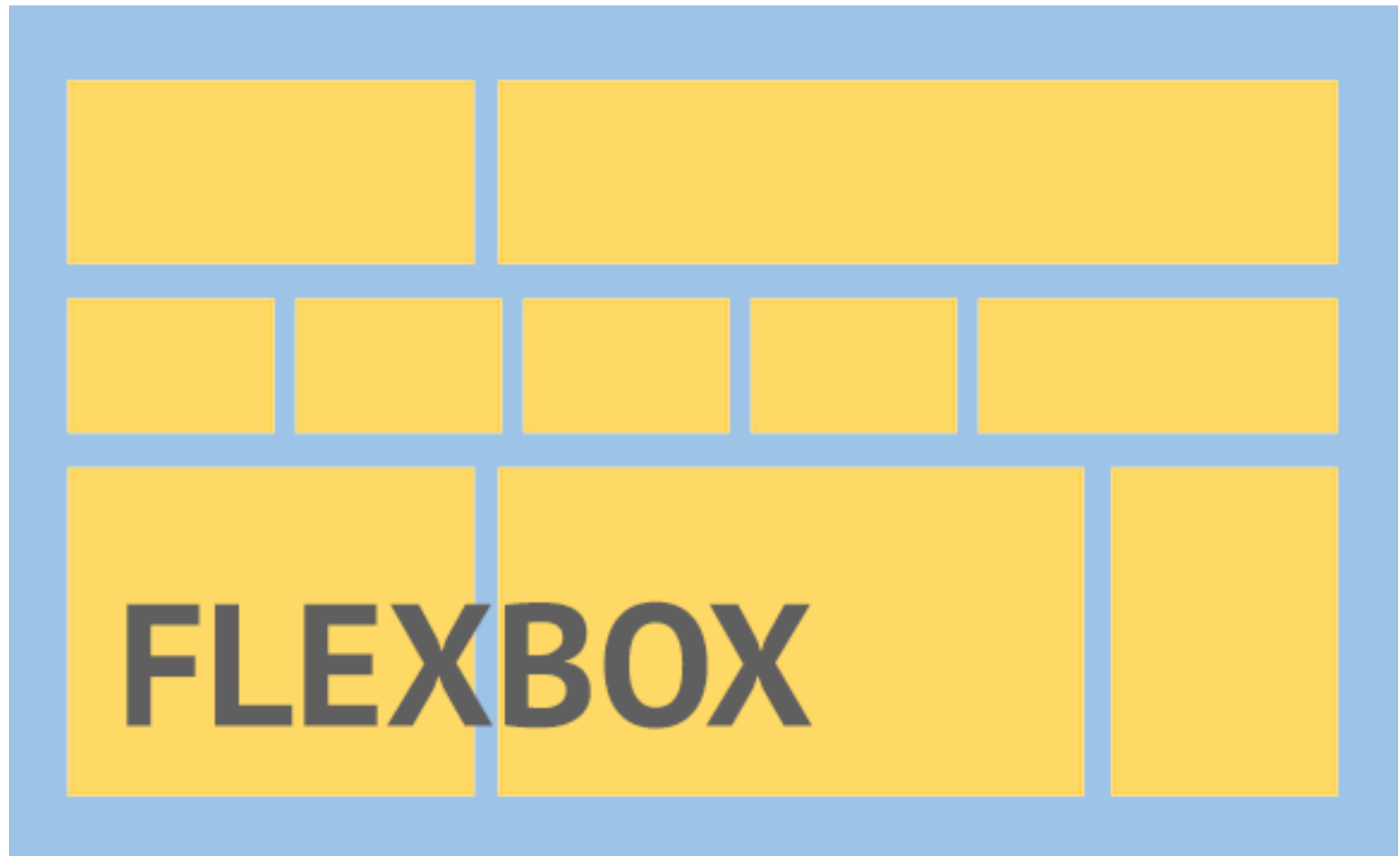
Propiedad Display

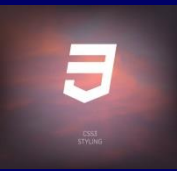
Algunos valores posibles:

- none: El elemento no se visualiza, y no ocupa el espacio
- inline: El elemento ocupa el espacio justo y necesario (omite height y width)
- block: El elemento ocupa todo el ancho disponible (valor por defecto)
- list-item: El elemento se visualiza como una item de una lista
- Inline-block: El elemento ocupa el espacio justo respetando height y width
- Flex: Modelo de Cajas Flexibles (Flex Box)
- Grid: Modelo de Cuadrículas



Maquetación

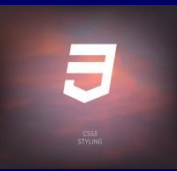




Maquetación

```
section {  
  display: -webkit-box;  
  display: -moz-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: box;  
  display: flex;  
}
```



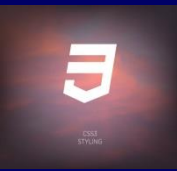


Maquetación

```
section {  
  display: -webkit-box;  
  display: -moz-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: box;  
  display: flex;  
  -webkit-justify-content: space-between;  
  -moz-justify-content: space-between;  
  -o-justify-content: space-between;  
  -ms-justify-content: space-between;  
  justify-content: space-between;  
}
```



Layout de un sitio Web

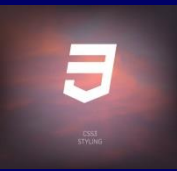


Maquetación

```
section {  
  -webkit-align-items: flex-end;  
  -o-align-items: flex-end;  
  -ms-align-items: flex-end;  
  -moz-align-items: flex-end;  
  align-items: flex-end;  
  display: -webkit-box;  
  display: -moz-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: box;  
  display: flex;  
  -webkit-justify-content: space-around;  
  -moz-justify-content: space-around;  
  -o-justify-content: space-around;  
  -ms-justify-content: space-around;  
  justify-content: space-around;  
}
```

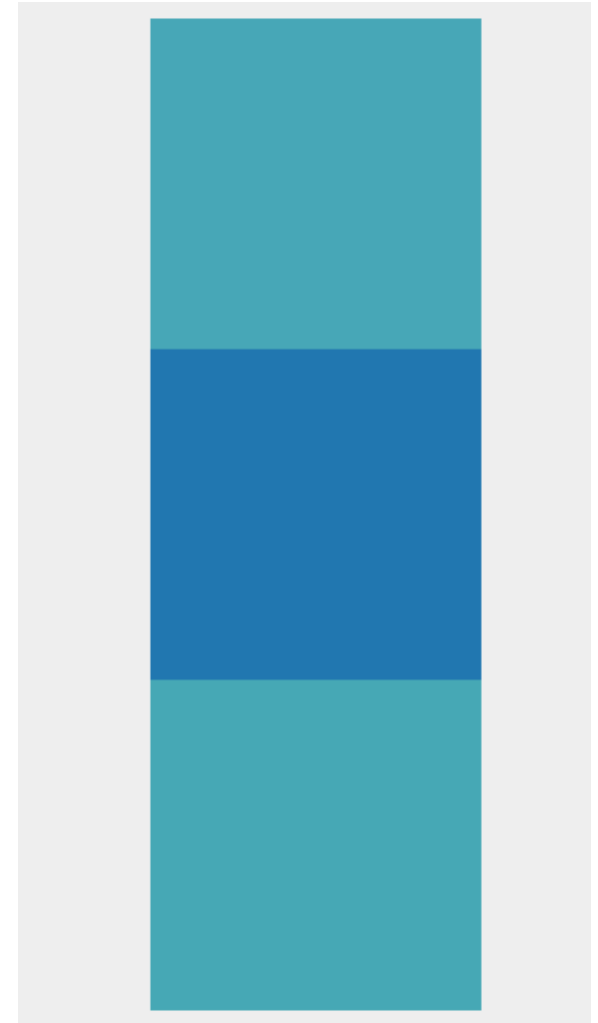


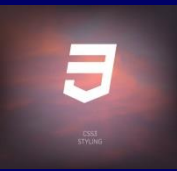
Layout de un sitio Web



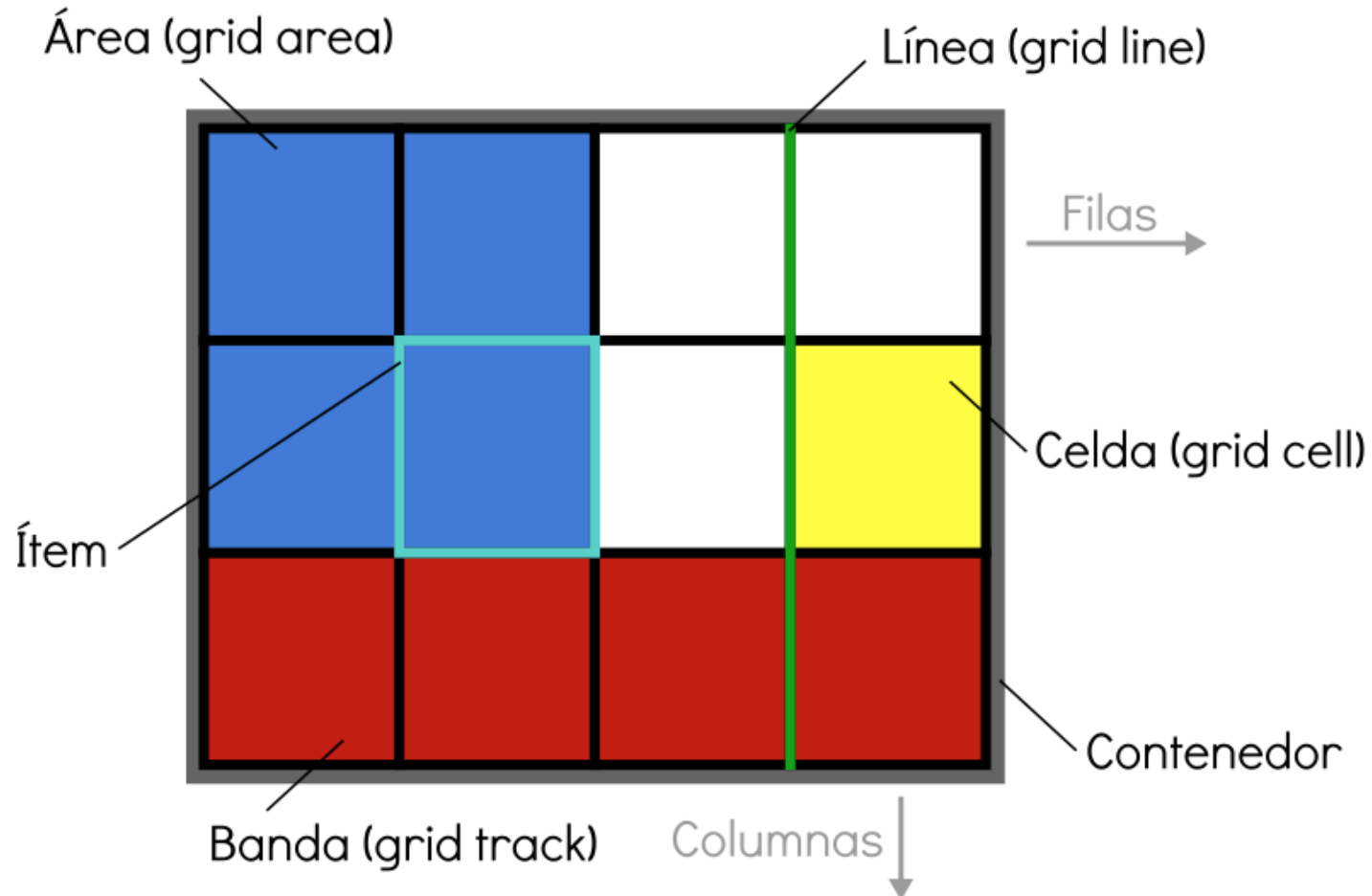
Maquetación

```
section {  
  -webkit-flex-wrap: wrap;  
  -o-flex-wrap: wrap;  
  -ms-flex-wrap: wrap;  
  -moz-flex-wrap: wrap;  
  flex-wrap: flex-end;  
  display: -webkit-box;  
  display: -moz-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: box;  
  display: flex;  
  -webkit-justify-content: space-around;  
  -moz-justify-content: space-around;  
  -o-justify-content: space-around;  
  -ms-justify-content: space-around;  
  justify-content: space-around;  
}
```





Maquetación





Maquetación

Contenedor: Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.

Ítem: Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).

Celda (grid cell): Cada uno de los cuadritos (unidad mínima) de la cuadrícula.

Area (grid area): Región o conjunto de celdas de la cuadrícula.

Banda (grid track): Banda horizontal o vertical de celdas de la cuadrícula.

Línea (grid line): Separador horizontal o vertical de las celdas de la cuadrícula.

Layout de un sitio Web



Maquetación

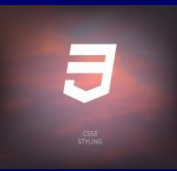
```
<div class="wrapper">
<div class="box a">A</div>
<div class="box b">B</div>
<div class="box c">C</div>
<div class="box d">D</div>
<div class="box e">E</div>
<div class="box f">F</div>
</div>
```



```
.wrapper {
display: grid;
grid-template-columns: 100px 100px 100px;
grid-template-rows: 80px 80px;
grid-gap: 10px;
background-color: #fff;
color: #444;
}
```

```
.box {
background-color: #444;
color: #fff;
border-radius: 5px;
padding: 20px;
font-size: 150%;
}
```


Layout de un sitio Web



Maquetación

```
<div class="wrapper">  
<div class="box a">A</div>  
<div class="box b">B</div>  
<div class="box c">C</div>  
<div class="box d">D</div>  
</div>
```



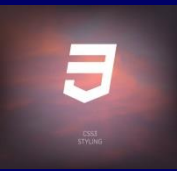
```
.wrapper {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: 100px 100px 100px;  
  background-color: #fff;  
}
```

```
.box {  
  background-color: #444;  
  padding: 20px;  
  border-radius: 5px;  
  font-size: 150%;  
}
```

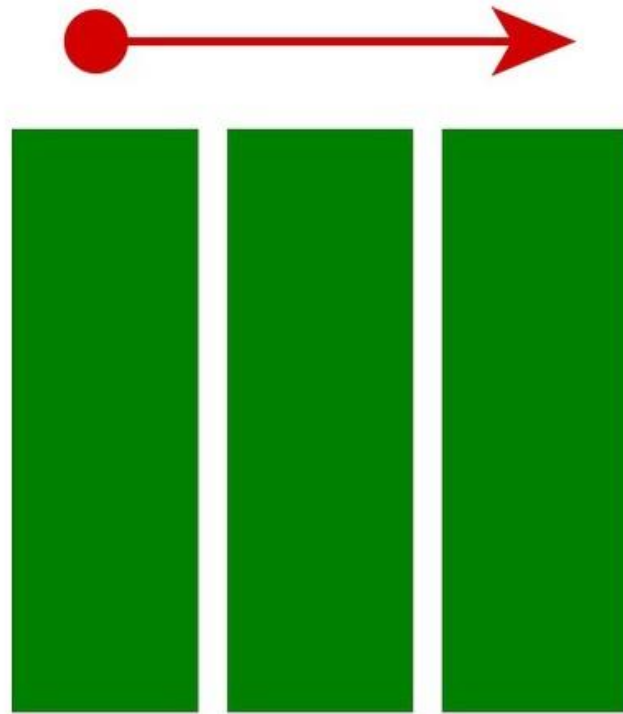
```
.a {  
  grid-column: 1 / 3;  
  grid-row: 1;  
}
```

```
.b {  
  grid-column: 3 ;  
  grid-row: 1 / 3;  
}  
.c {  
  grid-column: 1 ;  
  grid-row: 2 ;  
}  
.d {  
  grid-column: 2;  
  grid-row: 2;  
}
```

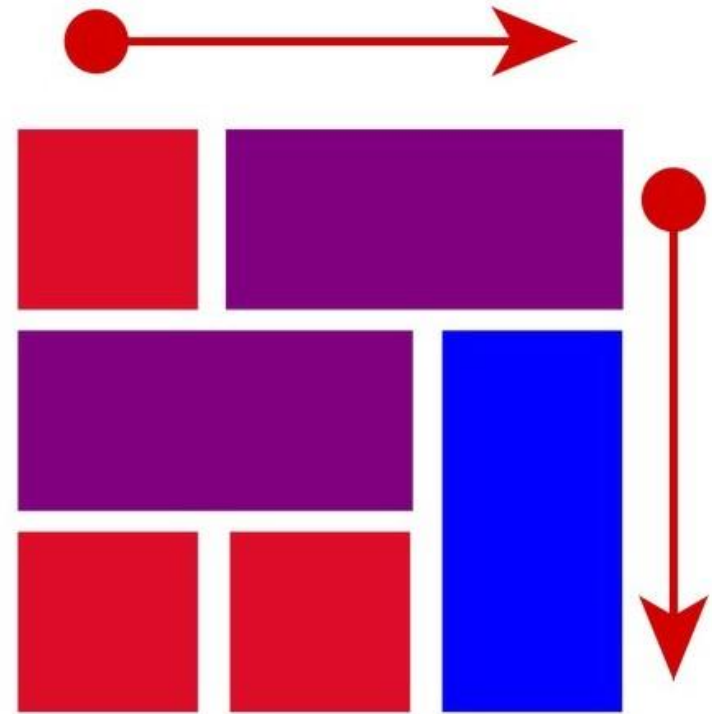
Layout de un sitio Web



Maquetación



Flexbox
One Dimensions



CSS Grids
Two Dimensions



Maquetación

CSS Grid es mejor para la construcción general

CSS Grid en diseños con filas y columnas.

Flexbox alineando el contenido dentro de los diferentes elementos de la web.

Flexbox funciona mejor en una dimensión.

Propiedades CSS3



CSS3 provee nuevas propiedades y funciones para crear efectos visuales y dinámicos que son parte esencial de la web actualmente.

- **border-radius**
- **box-shadow**
- **text-shadow**
- **@font-face**
- **linear-gradient**
- **radial-gradient**
- **rgba**
- **hsla**
- **outline**
- **border-image**
- **transform**
- **transition**

Importante: Algunas de las propiedades arriba mencionadas se encuentran en estado experimental en algunos navegadores. Para aplicarlas efectivamente es necesario declararla con los correspondientes prefijos (ver compatibilidades)



Propiedad box-shadow

Esta propiedad crea sombras para la caja formada por el elemento.

- Puede tomar cinco parámetros: el color, el desplazamiento horizontal, el desplazamiento vertical, el valor de difuminación, y la palabra clave inset para generar una sombra interna.
- Los desplazamientos pueden ser negativos, y el valor de difuminación y el valor inset son opcionales (por ejemplo, box-shadow: #000000 5px 5px 10px inset).

#principal {

```
-moz-box-shadow: rgb(150,150,150) 5px 5px 10px inset;  
-webkit-box-shadow: rgb(150,150,150) 5px 5px 10px inset;  
box-shadow: rgb(150,150,150) 5px 5px 10px inset;  
}
```



Este estilo mostrará una sombra interna alejada del borde de la caja por unos 5 píxeles y con un efecto de difuminación de 10 píxeles.



Propiedad box-shadow

- ***Importante: El color si es requerido en Safari para PC por eso es conveniente que siempre esté definido***

Ejemplo

```
#elemento {  
    box-shadow: 3px 4px 3px 3px #999;  
}
```

el contenido



Propiedad text-shadow

Esta propiedad es similar a box-shadow pero específica para textos.

- Toma cuatro parámetros: el color de la sombra, la distancia horizontal y vertical de la sombra con respecto al objeto y el radio de difuminación.

```
#titulo {  
    font: bold 36px verdana, sans-serif;  
    text-shadow: rgb(0,0,150) 3px 3px 5px;  
}
```



Este estilo muestra una sombra azul aplicada al título de nuestra plantilla con una distancia de 3 pixeles y un radio de difuminación de 5.



Propiedad text-shadow

- Offset Horizontal y Vertical (pueden ser positivos o negativos). Son datos requeridos
- Radio de difuminado y Color (campos opcionales).

Ejemplo

```
#elemento {  
    text-shadow: 2px 2px #999;  
}
```

Texto



Regla @font-face

Esta regla nos permite cargar y usar cualquier fuente que necesitemos.

- Primero, debemos declarar la fuente, proveer un nombre con la propiedad font-family y especificar el archivo con src (por ejemplo, @fontface{ font-family: MiFuente; src: url('font.ttf') }).
- Luego de esto, podremos asignar la fuente (en el ejemplo MiFuente) a cualquier elemento del documento.

```
#titulo {  
    font: bold 36px MiFuente, verdana, sans-serif;  
    text-shadow: rgb(0,0,150) 3px 3px 5px;  
}
```

```
@font-face {  
    font-family: 'MiFuente';  
    src: url('font.ttf');  
}
```



Regla @font-face

Ejemplo

```
@font-face {  
    font-family: googleFont;  
    src: url(fonts/google.ttf);  
}
```

Escribiendo con la fuente de Google®



Función linear-gradient

La función `linear-gradient(posición inicio, color inicial, color final)` puede ser aplicada a las propiedades `background` o `background-image` para generar un gradiente lineal.

- Los atributos indican el punto inicial y los colores usados para crear el gradiente.
- El primer valor puede ser especificado en pixeles, en porcentaje o usando las palabras clave `top`, `bottom`, `left` y `right`.
- El punto de inicio puede ser reemplazado por un ángulo para proveer una dirección específica para el gradiente (por ejemplo, `linear-gradient(top, #FFFFFF 50%, #006699 90%);`).

```
#principal {  
    background: -webkit-linear-gradient(top, #FFFFFF, #006699);  
    background: -moz-linear-gradient(top, #FFFFFF, #006699);  
}
```



Función linear-gradient

Ejemplo

```
#elemento {  
    background: linear-gradient(green, white);  
}
```





Función radial-gradient

La función radial-gradient(posición inicio, forma, color inicial, color final) puede ser aplicada a las propiedades background o background-image para generar un gradiente radial.

- La posición de inicio es el origen y puede ser declarado en pixeles, porcentaje o como una combinación de las palabras clave center, top, bottom, left y right.
- Existen dos valores para la forma: circle y ellipse, y puntos de terminación pueden ser declarados para cada color indicando la posición donde la transición comienza (por ejemplo, radial-gradient(center, circle, #FFFFFF 0%, #006699 200%);).

background: radial-gradient(center, circle, #FFFFFF 0%, #006699 200%);



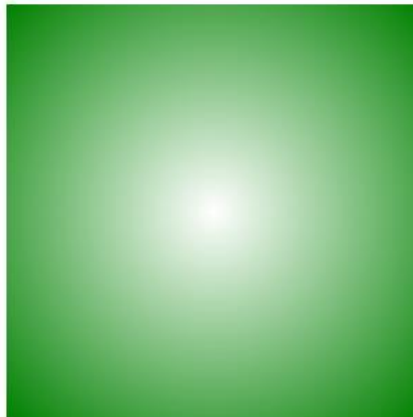
Función radial-gradient

Agrega un degradado radial de fondo en un elemento.

- Se debe indicar cuales son los dos o más colores entre los cuales se hará el degradado.

Ejemplo

```
#elemento {  
    background: radial-gradient(white, green);  
}
```





Función rgba

La función `rgba()` es una mejora de `rgb()`.

- Toma cuatro valores: el color rojo (0-255), el color verde (0-255), el color azul (0-255), y la opacidad (un valor entre 0 y 1).

```
#titulo {  
    font: bold 36px MiNuevaFuente, verdana, sans-serif;  
    text-shadow: rgba(0,0,0,0.5) 3px 3px 5px;  
}
```

Función hsla

La función `hsla()` es una mejora de `hsl()`.

- Puede tomar cuatro valores: el tono (un valor entre 0 y 360), la saturación (un porcentaje), la luminosidad (un porcentaje), y la opacidad (un valor entre 0 y 1).

```
#titulo {  
    font: bold 36px MiNuevaFuente, verdana, sans-serif;  
    text-shadow: rgba(0,0,0,0.5) 3px 3px 5px; color:  
    hsla(120, 100%, 50%, 0.5);  
}
```



Propiedad outline

Esta propiedad fue mejorada con la incorporación de otra propiedad llamada **outline-offset**. Ambas propiedades combinadas generan un segundo borde alejado del borde original del elemento (por ejemplo, `outline: 1px solid #000000; outline-offset: 10px;`).

```
#principal {  
    outline: 2px dashed #000099; outline-offset: 15px;  
}
```

La propiedad **outline** tiene similares características y usa los mismos parámetros que **border**. La propiedad **outline-offset** solo necesita un **valor en pixeles**.



Propiedad outline

Ejemplo: Se agrega un borde negro y un contorno rojo

```
#elemento {  
    border: 1px solid black;  
    outline: 1px dashed red;  
}
```





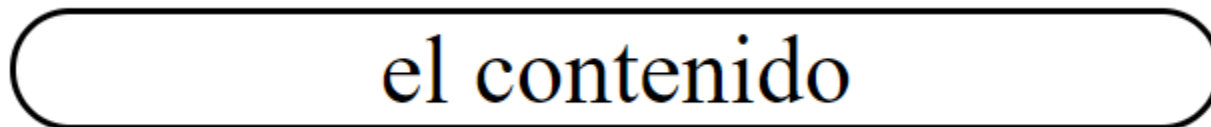
Propiedad border-radius

Esta propiedad crea un borde curvo al elemento.

- Se puede indicar un radio fijo o un porcentaje
- Los valores pueden ser selectivos por esquina, al igual que con el margen se puede definir 1, 2, 3 o 4 valores en el sentido de las agujas del reloj.

Ejemplo

```
#elemento {  
    border: 1px solid black;  
    border-radius: 25px;  
}
```





Propiedad border-image

Esta propiedad crea un borde con una imagen personalizada.

Necesita que el borde sea declarado previamente con las propiedades border o border-width, y toma al menos tres parámetros:

- la URL de la imagen,
- el tamaño de las piezas que serán tomadas de la imagen para construir el borde,
- una palabra clave (repeat – round o stretch) que especifica cómo esas piezas serán ubicadas alrededor del elemento (por ejemplo, border-image: url("file.png") 15 stretch;)

```
#principal {  
    -moz-border-image: url("diamonds.png") 29 stretch;  
    -webkit-border-image: url("diamonds.png") 29 stretch;  
    border-image: url("diamonds.png") 29 stretch;  
}
```



Propiedad border-image

Ejemplo

```
#elemento {  
    border-image:url("border.png") 30 30 round;  
}
```





Propiedad transform

Esta propiedad modifica la forma de un elemento. Utiliza cuatro funciones básicas: **scale** (escalar), **rotate** (rotar), **skew** (inclinarse), y **translate** (trasladar o mover).

- La función **scale** recibe solo un parámetro. Un valor negativo invierte el elemento, valores entre 0 y 1 reducen el elemento y valores mayores que 1 expanden el elemento (por ejemplo, ***transform: scale(1.5);***).
- La función **rotate** usa solo un parámetro expresado en grados para rotar el elemento (por ejemplo, ***transform: rotate(20deg);***).
- La función **skew** recibe dos valores, también en grados, para la transformación horizontal y vertical (por ejemplo, ***transform: skew(20deg, 20deg);***).
- La función **translate** mueve el objeto tantos píxeles como sean especificados por sus parámetros (por ejemplo, ***transform: translate(20px);***).



Propiedad transform

A veces podría resultar útil realizar sobre un elemento varias transformaciones al mismo tiempo. Para obtener una propiedad **transform** combinada, solo tenemos que separar cada función a aplicar con un espacio:

```
#principal {  
  -moz-transform: translateY(100px) rotate(45deg) scaleX(0.3);  
  -webkit-transform: translateY(100px) rotate(45deg) scaleX(0.3);  
}
```

Podemos aprovecharnos de la combinación de transformaciones y pseudo clases para convertir nuestra página en una aplicación dinámica:

```
#principal:hover{  
  -moz-transform: rotate(5deg);  
  -webkit-transform: rotate(5deg);  
}
```



Usando la vieja pseudo clase :hover, el resultado obtenido es que cada vez que el puntero del ratón pasa sobre esta caja, la propiedad transform rota la caja en 5 grados, y cuando el puntero se aleja la caja vuelve a rotar de regreso a su posición original



Propiedad transition

Esta propiedad puede ser aplicada para crear una transición entre dos estados de un elemento. Recibe hasta cuatro parámetros:

- la propiedad afectada,
- el tiempo que le tomará a la transición desde el comienzo hasta el final,
- una palabra clave para especificar cómo la transición será realizada (ease, linear, ease-in, ease-out, ease-in-out)
- un valor de retardo que determina el tiempo que la transición tardará en comenzar (por ejemplo, transition: color 2s linear 1s;).

```
#principal {  
    -moz-transition: -moz-transform 1s ease-in-out 0.5s;  
    -webkit-transition: -webkit-transform 1s ease-in-out 0.5s;  
}
```

Diseño Adaptativo



Hace algunos años, era sencillo definir el ancho del sitio



1995: 640 x 480px

2000: 800 x 600px

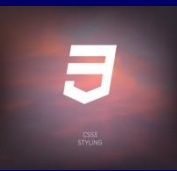


2005: 1024 x 768px

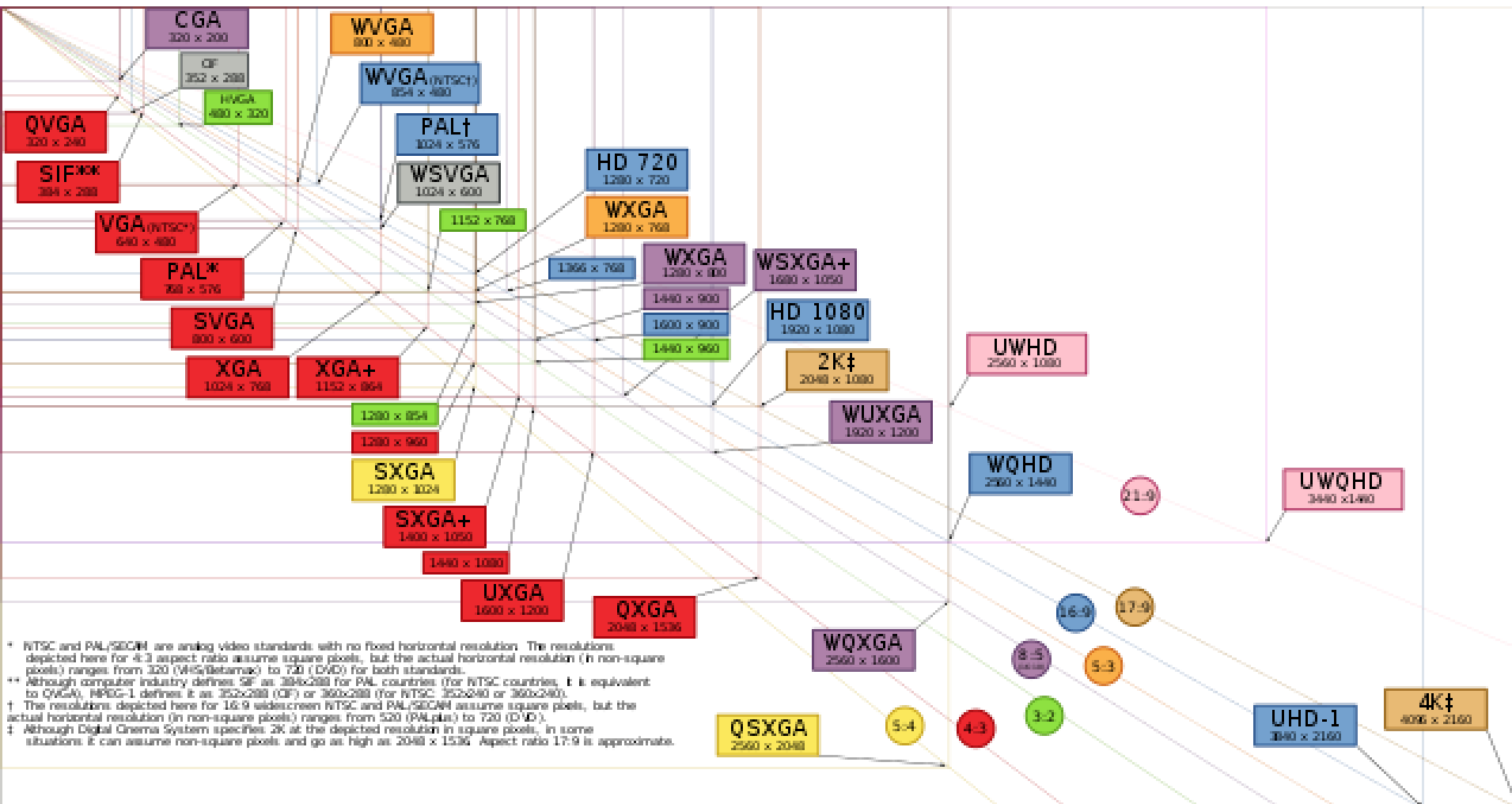


Ancho fijo centrado, o ancho líquido (en %)

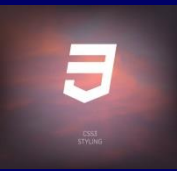
Diseño Adaptativo



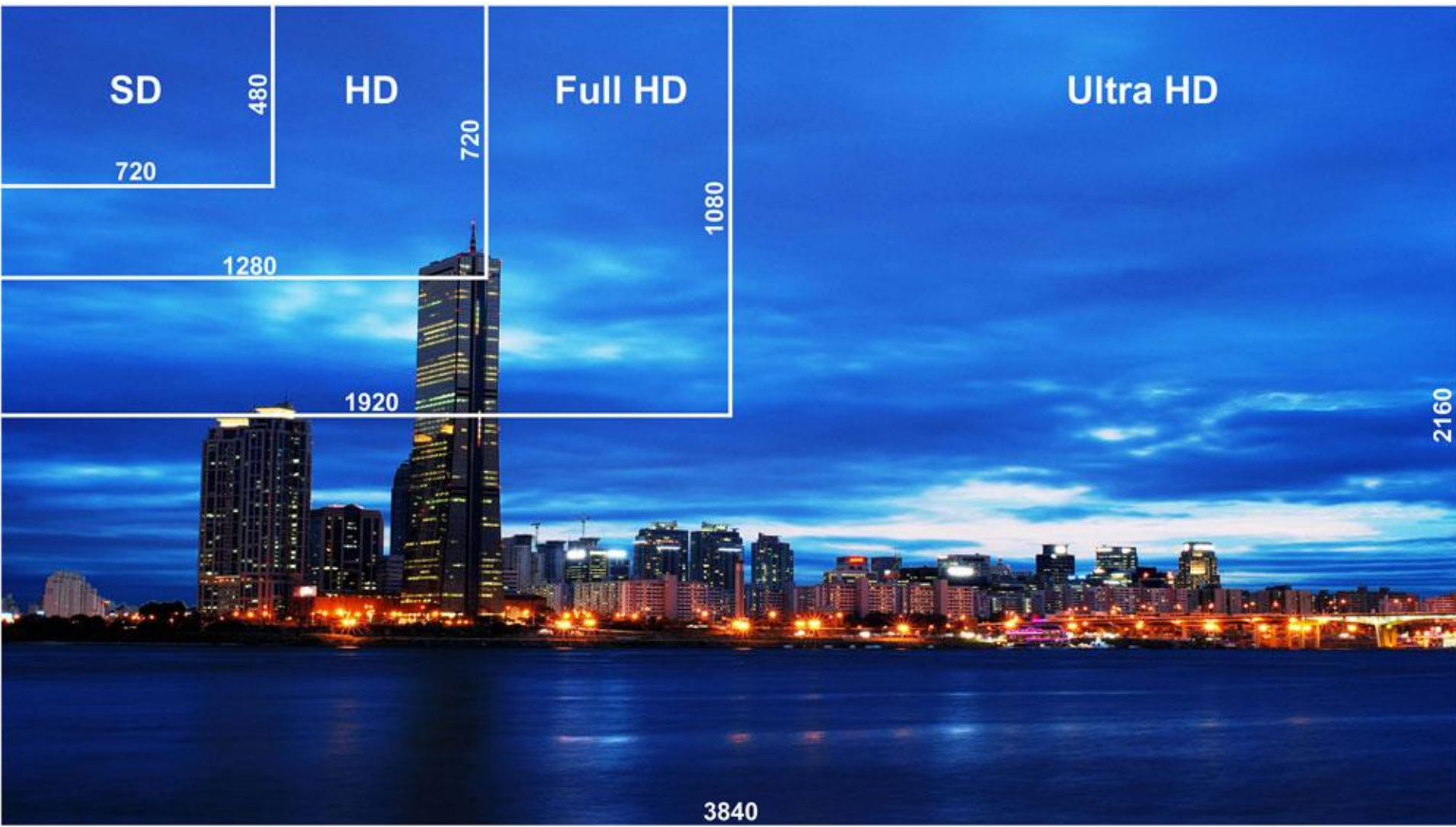
Los monitores crecieron.. crecieron las resoluciones



Diseño Adaptativo



Hoy son múltiples las posibilidades...



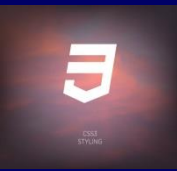
Diseño Adaptativo



Hoy son múltiples las posibilidades...



Diseño Adaptativo



Hoy son múltiples las posibilidades...





Problema: “miniaturización”

Textos ilegibles

Fotos irreconocibles

Links no pulsables

Zoom constante

Scroll ← horizontal →

Objetivo: Distinto tamaño = Distinto Diseño

The image displays four versions of the ISO website, illustrating how the layout adapts to different screen sizes. Each version is shown with a callout box indicating its device type and column count.

- 1 columna -celular-:** The mobile version shows a single column layout with a large header image and text, followed by a list of popular standards.
- 2 columnas -tableta chica-:** The small tablet version shows a two-column layout, with the header image and text on the left, and a list of popular standards on the right.
- 3 columnas -tableta grande-:** The large tablet version shows a three-column layout, with the header image and text on the left, and two columns of content on the right.
- 3 columnas (o más) -PC-:** The desktop version shows a three-column layout, with the header image and text on the left, and two columns of content on the right.

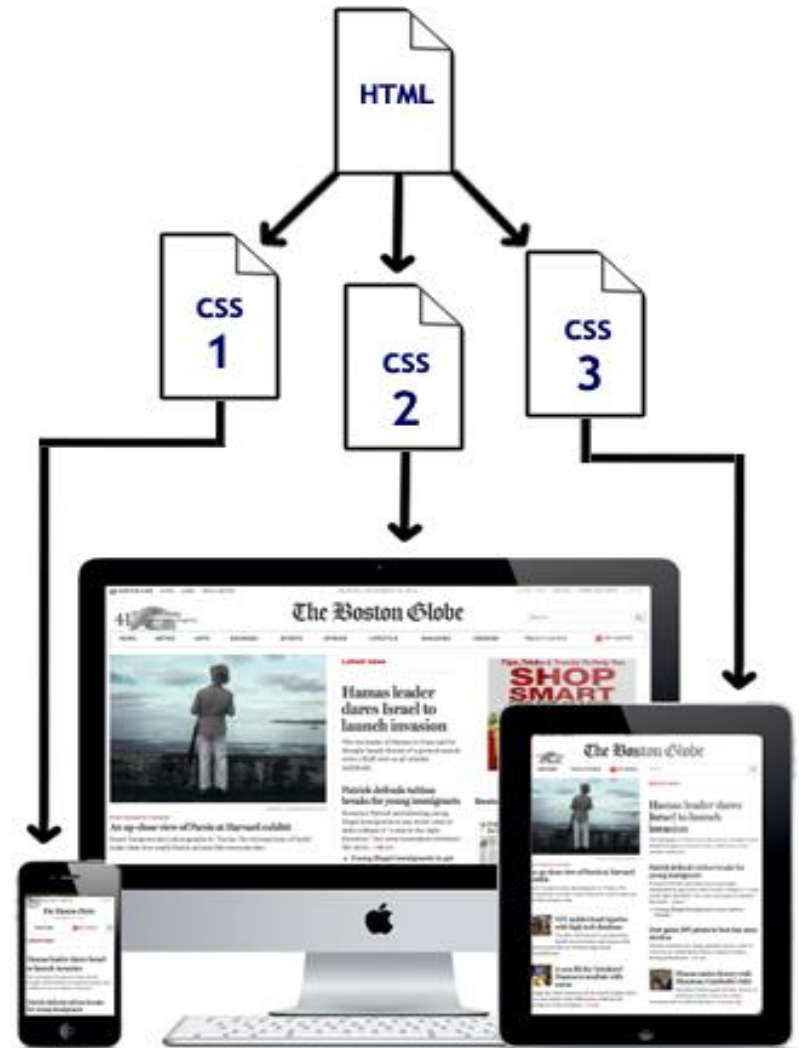


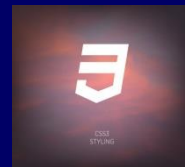
Guía Sugerida

- 1) Optimizar **layout** (cantidad de columnas)
- 2) Adaptar textos y fotos decorativas para garantizar **legibilidad**
- 3) Hacer **medios** flexibles (fotos, sliders, videos, mapas, tablas de datos)
- 4) Herramientas de **navegación** adaptadas

MEDIA QUERIES

Distintos **estilos**
dependiendo de
características
del dispositivo
(condiciones)





Medios CSS

| Medio | Descripción |
|------------|---|
| all | Todos los medios definidos |
| braille | Dispositivos táctiles que emplean el sistema braille |
| embosed | Impresoras braille |
| handheld | Dispositivos de mano: móviles, PDA, etc. |
| print | Impresoras y navegadores en el modo "Vista Previa para Imprimir" |
| projection | Proyectores y dispositivos para presentaciones |
| screen | Pantallas de ordenador |
| speech | Sintetizadores para navegadores de voz utilizados por personas discapacitadas |
| tty | Dispositivos textuales limitados como teletipos y terminales de texto |
| tv | Televisores y dispositivos con resolución baja |



Implementación de Media Queries

Esquema: **medio** and **(condición)**

Ejemplo en HTML:

```
<link rel="stylesheet" media="screen and (min-width:1024px)"  
href="grande.css">
```

Ejemplo en CSS:

```
@media screen and (min-width:1024px) {  
  
...  
}
```



Propiedades disponibles para condiciones

Width / Height / Device-width / Device-height

Orientation (portrait o landscape)

Aspect-ratio y Device-aspect-ratio

Resolution

Y “supuestamente” más... como Color,
Color-index y Monochrome

Nuevas Propiedades CSS3



Pero usaremos generalmente..

Width (con min-width y/o max-width)

Resolution y Aspect-ratio



Tamaños de Fuente

Porcentajes + Em's

```
body{ font-size:60%; }  
h1{ font-size:2em; }  
p{ font-size:1em; }
```

} Body en %

} Elementos en **em**

```
@media screen and (min-width:480px) {  
  body{font-size:80%;}  
}
```



VIEWPORT

Se utiliza para definir qué área de pantalla está disponible al renderizar una página, nivel de escalado y el zoom que debe mostrar inicialmente.

Disponemos de los siguientes parámetros en la etiqueta META

- *Width*: anchura virtual (emulada) de la pantalla o anchura del viewport.
- *Height*: altura virtual de la pantalla o anchura del viewport.
- *Initial-scale*: la escala inicial del documento.
- *Minimum-scale*: la escala mínima que se puede poner en el documento.
- *Maximum-scale*: la escala máxima configurable en el documento.
- *User-scalable*: si se permite o no al usuario hacer zoom.



VIEWPORT

```
<meta name="viewport" content="user-  
scalable=no, width=device-width, initial-scale=1">
```



BREAKPOINTS

Permite implementar CSS dependiendo de la media query alcanzada

```
/* Smartphones (portrait and landscape) ----- */  
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) { /*  
Styles */ }
```

```
/* Smartphones (landscape) ----- */  
@media only screen and (min-width : 321px) { /* Styles */ }
```

```
/* Smartphones (portrait) ----- */  
@media only screen and (max-width : 320px) { /* Styles */ }
```




BREAKPOINTS

Estrategias

1) Scaffolding: “acumula” estilos

Al menos 480px

Al menos 800px (también **aplica anterior**)

Al menos 1024px (también **aplica anteriores**)

2) Rangos: excluye (se usa para algo exclusivo de tamaños intermedios)

Al menos 320 y **como máximo** 480

Al menos 481 y **como máximo** 800 (no aplica anterior)



Compatibilidad con Navegadores

http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28Cascading_Style_Sheets%29

Desarrollador de Aplicaciones Web

Programación Web I

1er Cuatrimestre (2018)



Departamento de Ingeniería e Investigaciones Tecnológicas

Muchas gracias

Comisión Miércoles Noche

**Ing. Rubén Goría
Alicia Rosenthal**

Comisión Viernes Mañana

**Ing. Gerardo Barbosa
Ing. Gabriel Pañik**