

U2. Uso de estilos. Parte III

Diseño de Interfaces Web

Contenidos

- Display y visibility
- Validación CSS y HTML5
- Reglas @
- Diseño web responsive
 - Diseño fluido vs adaptativo
 - Media queries
 - Diseño en móviles: Viewport
 - Diseño de grid fluido
 - Imágenes
- Frameworks y plantillas

Display

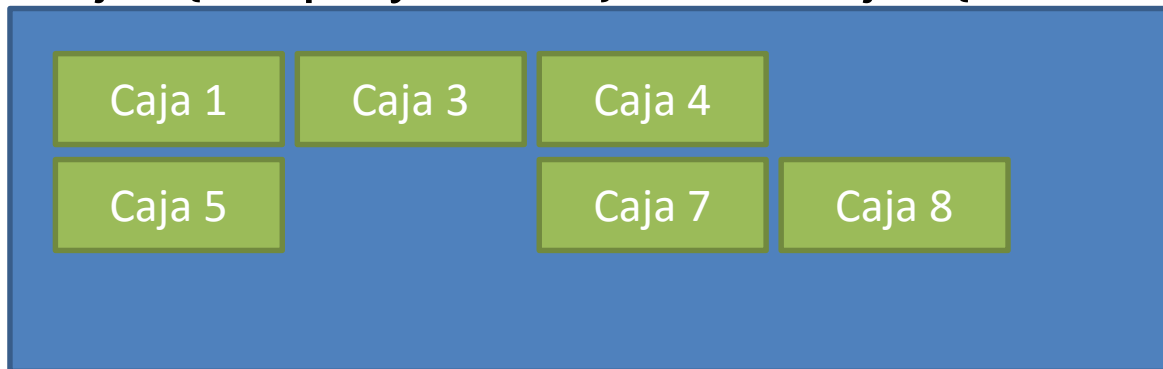
- La propiedad [display](#) modifica la manera en la que se muestra un elemento. Sus valores más utilizados son: **inline** | **block** | **none**
- Además display cuenta con otros muchos valores posibles interesantes:
 - **inline-block**
Permite que los elementos sean inline pero en conjunto los trata como un bloque.
 - **table y table-cell**
Permite que el elemento contenedor se comporte como una tabla.

Visibility

- Visibility permite hacer invisible un elemento. Sus valores posibles son: visible o hidden
- La diferencia entre visibility:hidden y display:none es que en el primer caso se crea ese elemento pero no se muestra.

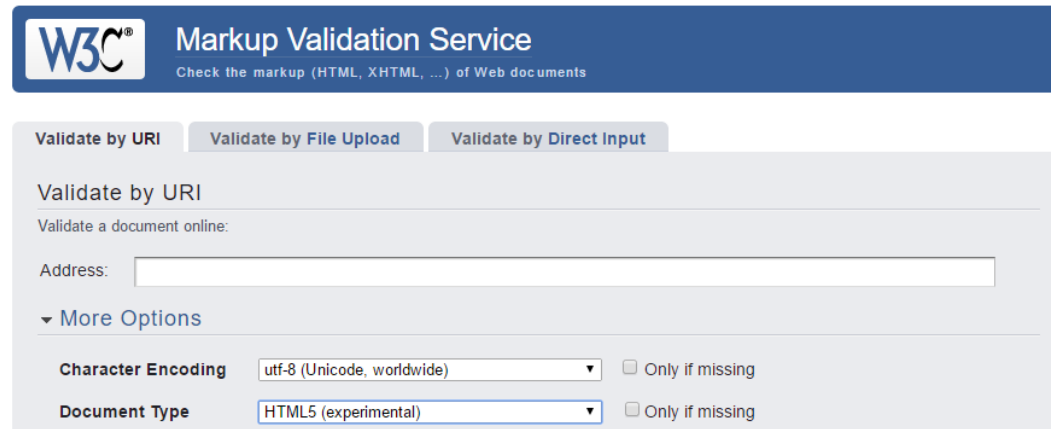
Caja2{ display:none }

Caja6{ visibility:hidden }



Validación CSS y HTML5

- Validador HTML5: <https://validator.w3.org/>

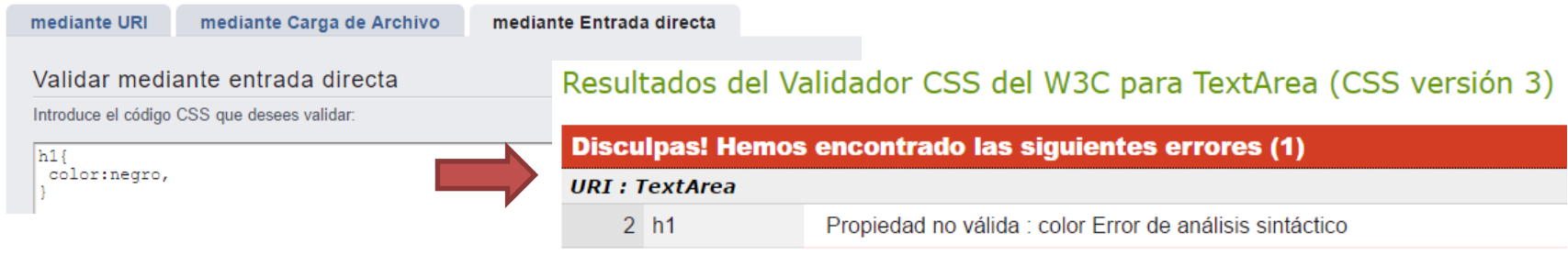


The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" tab is selected. Under this tab, there's a section "Validate by URI" with the text "Validate a document online:". Below this is a text input field labeled "Address:". Further down, there's a section "More Options" with two dropdown menus: "Character Encoding" (set to "utf-8 (Unicode, worldwide)") and "Document Type" (set to "HTML5 (experimental)"). There are also checkboxes for "Only if missing" next to each dropdown.

- Validador CSS: <https://jigsaw.w3.org/css-validator/validator>



The screenshot shows the W3C CSS Validation Service interface. At the top, there's a blue header with the W3C logo and the text "CSS Validation Service" and "Verifica Hojas de Estilo en Cascada (CSS) y documentos (X)HTML con hojas de estilo". Below this, there are three tabs: "mediante URI", "mediante Carga de Archivo", and "mediante Entrada directa". The "mediante Entrada directa" tab is selected.



The screenshot shows the W3C CSS Validation Service interface with the "mediante Entrada directa" tab selected. The text "Validar mediante entrada directa" is visible, followed by "Introduce el código CSS que desees validar:". Below this is a text area containing the CSS code:

```
h1{
  color:negro,
}
```

. A red arrow points from this text area to the results section. The results section has a green header "Resultados del Validador CSS del W3C para TextArea (CSS versión 3)". Below this is a red banner with the text "Disculpas! Hemos encontrado las siguientes errores (1)". Underneath, there's a table with the following content:

URI : TextArea	
2	h1 Propiedad no válida : color Error de análisis sintáctico

Reglas @

- Además de las reglas CSS, podemos añadir reglas @:

Regla	Descripción
@import	Agrega estilos CSS externos
<u>@media</u>	Aplica las reglas CSS al tipo de dispositivo que se indica
<u>@font-face</u>	Especifica una fuente externa para que el usuario la descargue
@charset	Especifica el charset
@page	Establece las dimensiones, orientación y márgenes del documento.
@supports	Especifica condiciones que se deben cumplir para usar una hoja de estilos.

Diseño web responsive

Técnica de diseño y desarrollo web que permite que un sitio web se vea adecuadamente en todo tipo de dispositivo.



Diseño responsive vs líquido o fluido

Diseño responsive o adaptativo



- Los contenidos se muestran alineados en un grid distinto (cambia el número de columnas) en función de la resolución

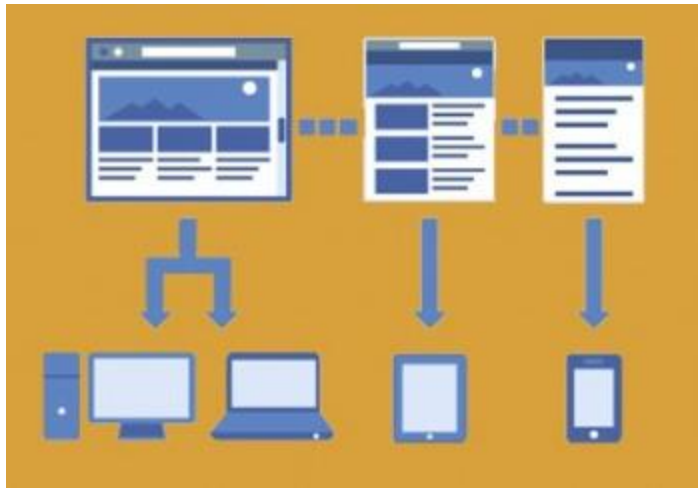
Diseño líquido o fluido



- Los contenidos se ajustan horizontalmente a la medida de la pantalla

Diseño web responsive

- La adecuación de la web a cada dispositivo, no sólo es una cuestión de tamaño, también de usabilidad y funcionalidad.



Diseño web responsive

- Para conseguir la adaptación de la web empleamos las herramientas siguientes:
 - **Media queries:** nos permiten asociar un estilo distinto en función del tipo de dispositivo y su resolución.
 - **Diseño líquido:** consiste en un diseño de grid flexible que se adapta a la resolución disponible (sin scrolls horizontales).
 - Contenidos e imágenes **multiresolución**.

Media queries

- Consisten en la consulta de información sobre el medio de visualización de una web: [tipo](#) y/o resolución.
- Esta consulta devolverá un booleano y en función de su valor, se aplicará o no el estilo especificado.

Opción a: `<link rel="stylesheet" media="screen and (width:780px)" href="estils780.css" />`

Opción b: `@media screen and (width:780px) {...}`

- Utilidad: proporcionan variantes al diseño base.

Media queries

```
<style>
  /*Estilo base que aplica siempre*/
  p {color:blue;}
  ...
  @media(min-width:1200px){
    ... /*Aplica a partir de 1200px*/
  }
  @media(min-width:992px) and (max-width:1199px){
    .../*Aplica entre 992 y 1199px*/
  }
</style>
```

- Otro ejemplo: <http://www.genbetadev.com/desarrollo-web/introduccion-a-las-reglas-media-de-css3>

Media queries: criterios consulta

Criterio	Tipo de valor	Descripción
Width (min/max)	número	Anchura de la zona de visualización del navegador.
Height (min/max)	número	Altura de la zona de visualización del navegador.
device-width	número	Anchura física del dispositivo
device-height	número	Altura física del dispositivo
orientation	<i>Landscape</i> <i>portrait</i>	Orientación del dispositivo apaisada o vertical
aspect-ratio	Ratio	Coeficiente ancho/alto.
color-index	Número	Número de colores de la tabla de colores.
monochrome	Número	Per examinar el número de niveles de gris en dispositivos monocromo.
resolution	DPI	Resolución de la pantalla expresada en DPI.

- Más información: https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Media queries: operadores

Comas (OR)	<p>Ej: ancho mínimo de 700px o en horizontal</p> <p>@media (min-width: 700px), (orientation: landscape) { ... }</p>
AND	<p>Ej: ancho mínimo y en horizontal</p> <p>@media (min-width: 700px) and (orientation: landscape) { ... }</p>
NOT (niega toda la query)	<p>Ej: no se aplica a cualquier tipo de dispositivo monocolor</p> <p>@media not all and (monochrome) { ... }</p>
ONLY (para versiones no compatibles con media query)	<p>Ej: proporciona una variante para los user-agent antiguos que toman la versión para pantalla en color.</p> <p><link rel="stylesheet" media="only screen and (color)" href="example.css" /></p>

Diseño en móviles: viewport

- Los navegadores de dispositivos móviles cuando entran en un sitio web escalan la web para que se muestre completamente en la pantalla.

iPhone vertical

- Dimensión real: 320px
- Dimensión emulada: 980px



SOLUCIÓN

Alterar el **viewport**
(área visible que el navegador del móvil emula que tiene)

Ejemplo: http://www.w3schools.com/css/css_rwd_viewport.asp

Diseño en móviles: viewport

```
<head>
```

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0" />
```

```
...
```

```
</head>
```

Otras consideraciones:

No todos los navegadores de dispositivos móviles resuelven igual el problema del zoom. En algunos casos tendremos que incluir `maximum-scale=1` para que siga sin hacer zoom en posición landscape (horizontal).

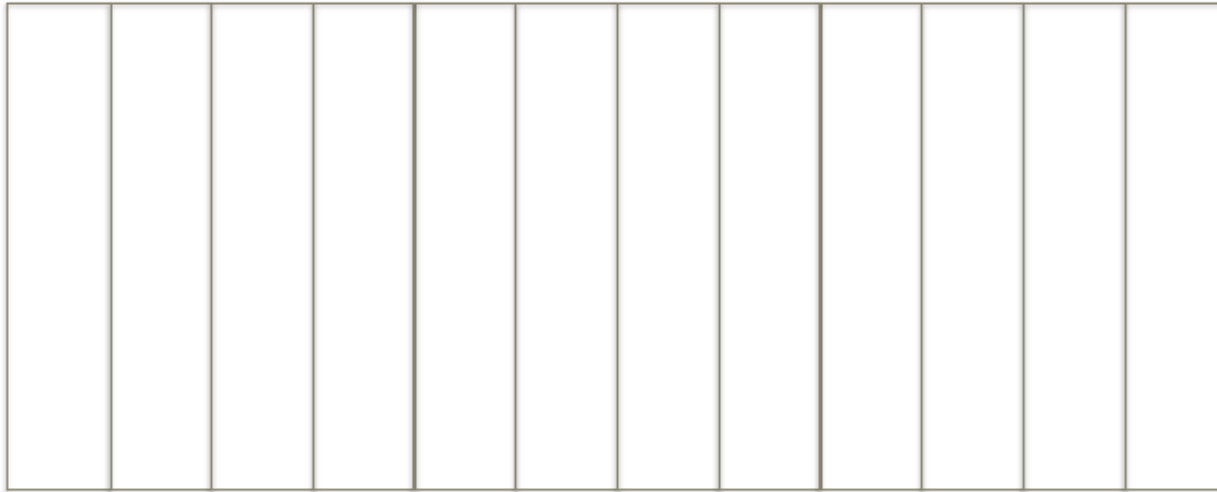
- Más info:

[https://developer.mozilla.org/es/docs/M%C3%B3vil/Viewport meta tag](https://developer.mozilla.org/es/docs/M%C3%B3vil/Viewport_meta_tag)

Diseño grid fluido

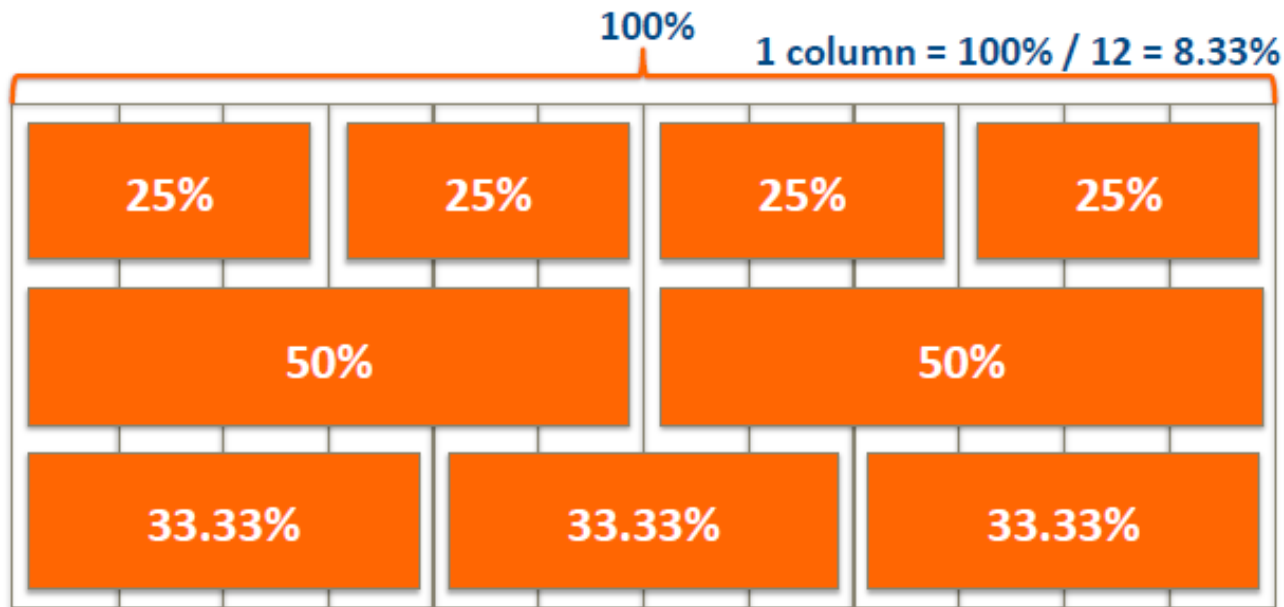
- Emplear un grid facilita el diseño del layout de la web

factors of 12: 1, 2, 3, 4, 6, 12



Diseño grid fluido

- Emplear un grid facilita el diseño del layout de la web



Ejemplo:

http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_styles

Grids con display **flex**: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Imágenes

- Podemos emplear distintas opciones
 - Emplear ancho adaptativo:
 - width:100%, la imagen aumentará o disminuirá
 - max-width:100%, la imagen sólo disminuirá (no superará su tamaño original)
 - Emplear diferentes imágenes:

```
body {background-image: url('img_small.jpg');}

@media (min-width: 400px) {
  body {background-image: url('img_large.jpg');}
}
```

Frameworks y plantillas

- El uso de frameworks de desarrollo y de plantillas son una alternativa para el desarrollo rápido de aplicaciones web.
- Los framework son un conjunto de utilidades que permiten un desarrollo de aplicaciones más rápido. Algunos de los más empleados son:
 - Bootstrap (getbootstrap.com),
 - Angular (angularjs.org),
 - Foundation ([oundation.zurb.com](https://foundation.zurb.com)),
 - Leaf (getleaf.com)
 - Materialize (materializecss.com)
- Más información: https://es.wikipedia.org/wiki/Framework_de_CSS