Elementos

● Permiten organizar el contenido del documento o acciones que se desencadenan cuando el navegador interpreta el documento.

● La declaración del elemento consta generalmente de cuatro partes:

○ Etiqueta de inicio:

○ Cero o más atributos

○ Contenido: puede ser otros elementos o texto

○ Etiqueta de fin:

Atributos

● Son propiedades asociadas a los elementos. Tienen asociado valores y modifican o extienden las acciones de los elementos.

● Los pares atributo/valor se colocan antes del cierre del inicio del elemento.

● Puede haber cualquier número de atributos separados por espacios en blanco y no tienen un orden predeterminado.

● La sintaxis es: atributo=”valor”

Documento

● : informa al navegador que el contenido siguiente se trata de un documento HTML.

● : es el elemento raíz.

● : es la cabecera del documento y contiene metadata.

● : es el cuerpo del documento y donde se coloca el contenido que se quiere mostrar, como textos, tablas, listas, imágenes, etc.

Elemento head

● Es la cabecera del documento. Contiene metadatos como el título, el conjunto de caracteres, scripts y otra información.

● Esta información no se muestra directamente en el documento.

● Algunos elementos contenidos en incluyen

Comentarios

● Si se desea introducir algún comentario a un documento HTML la sintaxis es:

● Los comentarios serán ignorados por el navegador

Titulares, párrafos y texto

● <h1>, <h2>, <h3>... <h6>: Titulares. Sirven para

dividir el texto en párrafos. Se pueden definir

seis niveles de titulares.

● <p>: Párrafos.

● Acomodan el texto al tamaño de la ventana.

● Sucesiones de espacios, tabulaciones y saltos de

líneas, no se visualizan en el navegador.

● Si se quieren agregar espacios se debe agregar

la entidad “&nbsp;” para formatear el texto.

● Para introducir una nueva línea dentro de un

párrafo, se usa el elemento de línea <br>

● Para cambiar de tema se suele usar <hr/>

Formato de texto

● Los siguientes elementos permiten dar formato

al texto:

○ <b>negrita</b>

○ <strong>importante</strong>

○ <i>itálica o cursiva</i>

○ <em>enfatizado</em>

○ <mark>marcado</mark>

○ <small>pequeño</small>

○ <ins>insertado/nuevo</ins>

○ <del>eliminado/viejo</del>

○ <sub>subíndice</sub>

○ <sup>superíndice</sup>

Enlaces

● El elemento **<a>** (“anchor”, ancla) se utiliza para crear hipervínculos.

● El atributo **href** especifica el destino.

● El atributo target dice donde abrirlo

○ **\_self**: misma ventana/pestaña (default)

○ **\_blank**: nueva ventana/pestaña

○ **\_parent**: ventana/pestaña padre

○ **\_top**: ventana/pestaña raíz o tope

● Utilizando **“#” (hash),** se especifica el id del

elemento a navegar. Útil como marcador.

● <https://es.wikipedia.org/wiki/HTML#Elementos>

URL absoluta y relativa

● Uniform Resource Locator (Localizador Uniforme de Recursos).

● Una URL absoluta contiene toda la información necesaria para localizar un recurso. Ejemplo: <http://www.ulp.edu.ar/images/logo-ulp.png>

● A diferencia de las URL absolutas, las relativas no incluyen el dominio. Si la dirección comienza con una barra “/”, será relativa al dominio. Si no, será relativa a la página.

● carpeta/otra/archivo.html

● ../padre.html

Imágenes

● **¡Error! Nombre de archivo no especificado.**permite mostrar imágenes.

● Sus principales atributos son: ○ src: especifica la fuente de la imagen. ○ alt: es el texto alternativo. ○ width: ancho de la imagen ○ height: alto de la imagen

● Si bien permite establecer tamaño por atributo, es preferible usar CSS.

● Si colocamos un **¡Error! Nombre de archivo no especificado.**dentro un la imagen funcionará como un enlace.

Listas

● Las listas permiten organizar conjuntos de datos relacionados.

● Para listas ordenadas usamos <ol>.

● Para listas sin orden usamos <ul>.

● Para ambas, un elemento de la lista es <li>.

● También existe el elemento <dl> que se usa

para listas de descripciones.

● Ésta última utiliza elementos <dt> y <dd> para el término de la definición y su descripción

respectivamente.

Tablas

● Las tablas permiten organizar datos en filas y columnas.

● La etiqueta <table> define una tabla HTML.

● Cada fila se define con etiquetas <tr>.

● Los encabezados se definen con etiquetas <th>.

● Los datos se definen con <td>.

● Los atributos colspan y rowspan, permiten a una celda ocupar más de una columna y más de

una fila, respectivamente

● Por defecto, los encabezados van en negrita, el resto de las celdas en forma regular y todos los textos alineados a la izquierda.

Tablas (partes)

● El elemento <caption> permite colocar un título o leyenda a una tabla. Debe venir

inmediatamente luego de <table>.

● Los elementos <thead>, <tbody> y <tfoot> permiten especificar el encabezado, cuerpo y

pie de una tabla.

Sintaxis

● Cada conjunto de reglas CSS consta de dos partes: el selector y un bloque de declaraciones.

● El selector apunta al elemento HTML al que desea aplicar estilo.

● El bloque de declaración contiene una o más declaraciones separadas por punto y coma.

● Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos

puntos.

CSS en línea

● Se puede usar un estilo en línea para aplicar un estilo único a un solo elemento.

● Para usar estilos en línea, se agrega el atributo de style (estilo) al elemento relevante.

● El atributo de estilo puede contener cualquier propiedad CSS.

Selectores

● Los selectores CSS se utilizan para "buscar" (o seleccionar) los elementos HTML a los que se

desea aplicar estilo.

● Podemos dividir los selectores de CSS en cinco categorías:

○ Simples (elemento, id y class)

○ Combinación (basados en relaciones)

○ Pseudo-clase (por estado)

○ Pseudo-elementos (parte de un elemento)

○ Por atributos (basados en un atributo o valor de atributo)

● Elemento

p {

color: red;

}

● Id (#)

#parrafo1 {

text-align: center;

color: blue;

}

● Clase (.)

.centrado {

text-align: center;

}

● Universal (\*)

\* {

margin: 0;

}

Selectores

● Agrupamiento (,)

h1, h2, h3 {

text-decoration: underline;

}

● Pseudo-elementos (::)

p::first-line {

color: #ff0000;

font-variant: small-caps;

}

● Pseudo-clases (:)

p:hover {

color: red;

}

● Combinaciones

p.azul {

color: darkblue;

Selectores combinados

● Descendente ( ). e f. Selecciona todos los

elementos de tipo f que están dentro de e sin

importar la profundidad. No confundir con el

agrupamiento (,).

p a {

color: red;

}

<body>

<p>

Ingrese a <a href=”...”>enlace</a>. O también a

<strong><a href=”...”>otro enlace</a></strong>

</p> …

</body>

/\* Todos los elementos de tipo

"p" con atributo class="aviso" \*/

p.aviso { ... }

<p class=”aviso”>SI</p>

<p class=”otra”>NO</p>

/\* Todos los elementos "p" y todos los elementos

con atributo class="aviso" de la página \*/

p, .aviso { ... }

<h1 class=”aviso”>SI</h1>

<h2>NO</h2>

<p><b class=”aviso”>SI</b></p>

<p>SI</p>

p .aviso { ... }

<p>

<span class=”aviso”>

SI

</span>

</p>

<p>

NO

<span class=”otra”>

NO

</span>

</p>

Selectores combinados

● Hijo (>). e > f. Selecciona todos los elementos de tipo f que son hijos directos de e.

p > a {

color: red;

}

<body>

<p>

Ingrese a <a href=”...”>enlace</a>. O también a

<strong><a href=”...”>otro enlace</a></strong>

</p> …

</body>

Selectores combinados

● Adyacente (+). e + f. Selecciona todos los

elementos de tipo f que cumplan las dos

siguientes condiciones:

○ e y f deben ser hermanos

○ f debe aparecer inmediatamente después

de e en el código HTML

h1 + h2 {

color: red;

text-decoration: underline;

}

<body>

<h1>Titulo1</h1>

<h2>Subtítulo</h2> …

<h2>Otro subtítulo</h2> …

</body>

Selectores combinados

● Hermano general (~). e ~ f. Selecciona todos los elementos de tipo f que cumplan las dos siguientes condiciones:

○ e y f deben ser hermanos

○ f debe aparecer inmediatamente después

de e en el código HTML

h1 ~ h2 {

color: red;

text-decoration: underline;

}

<body>

<h1>Titulo1</h1>

<h2>Subtítulo</h2> …

<h2>Otro subtítulo</h2> …

</body>

● Cuando más de una regla se puede aplicar a un elemento hay un conflicto a resolver.

● Cascada: cuando dos reglas tienen la misma especificidad, se aplica la última.

● Especificidad: se aplica la que tenga el selector más específico.

● Herencia: algunos valores de propiedades se

heredan del elemento padre. Otros no.

● Origen: es posible que el usuario configure un CSS para anular el estilo del desarrollador.

Aunque esto está en desuso.

Existen 4 valores universales de propiedades para el control de la herencia. Todas las propiedades CSS aceptan estos valores.

● inherit: hereda el valor del padre.

● initial: en propiedades no heredadas, este valor

refuerza el comportamiento por defecto y es

necesario sólo para sobrescribir otra regla.

● unset: es inherit para propiedades heredadas e

initial para no heredadas. Uso especial con “all:”.

● revert: revierte el valor al que habría tenido si no se hubiesen hecho cambios

Especificidad

21

● Mide cuán específico es un selector. Más

específico, más prioridad.

● Se calcula de la siguiente manera:

○ a: 1 si la declaración de estilo es inline si

no, 0

○ b: cantidad de atributos ID en el selector

○ c: cantidad de clases, pseudo-clases y

otros atributos en el selector

○ d: cantidad de elementos y

pseudo-elementos en el selector

● Luego se ordena un número con la forma: a,b,c,d.

Modificador !important

● Podemos declarar reglas de estilos como

!important para que tomen precedencia sobre otras reglas de estilos.

● No usar a menos que sea absolutamente

necesario ya que cambia el orden de aplicación de los estilos.

body{

font-family: verdana, arial !important;

Fuente

● Hay 5 familias genéricas: serif, sans-serif, monospace, cursive y fantasy.

● font-family se usa para establecer una lista ordenada de fuentes que se usarán para

mostrar un elemento. Usar al final una genérica como respaldo.

● font-style usado para hacer cursiva.

● font-weight indica la intensidad (negrita).

● font-variant permite pequeñas mayúsculas.

● font-size define el tamaño de la fuente.

● font es la propiedad abreviada para setear style, variant, weight, size y family.

Texto

● text-align para alineación horizontal.

● text-decoration permite decorar el texto (enlaces).

● text-transformation transforma el texto a mayúsculas, minúsculas y “capitalize”.

● text-indent regula la sangría de la 1ª línea.

● letter-spacing y word-spacing definen el espacio entre letras y palabras respectivamente.

● line-height define la altura del renglón.

● white-space dice como tratar los espacios.

● text-shadow establece el efecto de sombra.

Colores

● Los colores pueden especificarse por nombre o su valor RGB, HEX, HSL, RGBA, HSLA.

● Algunos nombres de colores son red, green,blue, gold.

● RGB utiliza 3 valores para armar el color: red, green y blue. Con HEX se usa su valor hexadecimal. HSL permite establecer tono, saturación y luz. Todos son equivalentes.

● color es usado para colorear el texto de une elemento.

● background-color permite establecer el color de fondo de un elemento.

Fondo

● Además del color, se puede establecer una imagen o un degradado como fondo.

● opacity establece la opacidad. El canal alfa del color puede lograr efectos similares pero no es lo mismo.

● background-image permite establecer una imagen como fondo.

● background-repeat dice como se repite esa imagen.

● background-position ubica la imagen.

● background-attachment dice si la imagen queda fija o se mueve con el scroll.

Pseudo clases y elementos

● Pueden usarse en CSS como selectores, pero no existen en el código fuente HTML.

● Son "insertados" bajo ciertas condiciones para que el CSS pueda referenciarlos.

● Son abstracciones que permiten referirse a elementos que de otro modo resulta imposible.

● Las pseudo clases (:) pueden asociarse a estados, como “hover” y “visited”, entre otros.

● Los pseudo elementos (::) pueden ver como partes de un elemento, como “first-letter” y

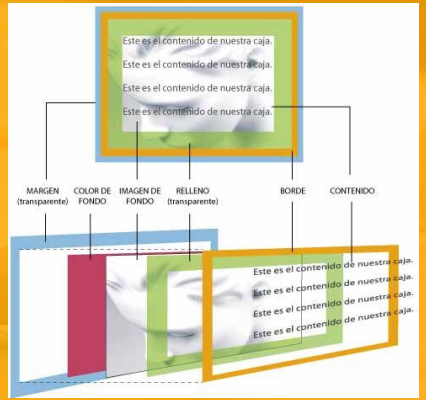
“first-line”.

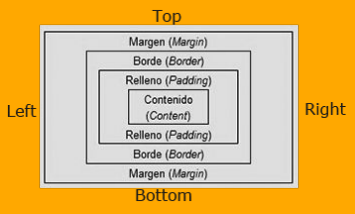
Modelo de Caja

● Todo en CSS tiene una caja alrededor.

● Controla altura, anchura, margen, etc.

● Se pueden ocultar, desplazar, fijarlas en una posición concreta dentro del documento, etc.





Bordes

● border-style especifica el tipo de borde.

● border-width especifica el ancho del borde. Se puede usar las unidades vistas (in px, cm, etc)

o thin, medium, o thick.

● border-color setea el color.

● Todas las propiedades se pueden setear para lados específicos, como border-top-color.

● Con border se puede setear el ancho, estilo (requerido) y el color de una sola vez.

● border-radius permite redondear los bordes.

Margen y Relleno

● margin crea espacio alrededor de los elementos, fuera de los bordes definidos.

● padding genera espacio alrededor del contenido de un elemento, dentro de los

bordes definidos.

● Posibilidad de especificar cada lado (-top, -bottom, -left y -right).

● La versión abreviada permite 4 valores (arriba, derecha, abajo e izquierda), 3 (arriba, abajo y

los costados), 2 (cada eje) y uno (todos igual).

Altura y anchura

● height (alto) y width (ancho) se utilizan para establecer altura y anchura de un elemento.

● No incluyen relleno, bordes y márgenes.

● min-width, min-height, max-width y max-height setean mínimos y máximos del

ancho y alto (ideal para responsivo).

● Con box-sizing:border-box se especifica que el padding y el border deben ser incluidos en el

total del ancho y alto del elemento.

Elementos de bloque y de línea (inline)

● Conforman la principal clasificación.

● Los elementos de bloque empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de su contenedor. Pueden contener otros elementos.

Ej.: p, h1..h6, ul, ol y div.

● Los elementos de línea sólo ocupan el espacio necesario para mostrar sus contenidos. Solo

pueden contener otros elementos en línea. Ej.: img, a, b y span.

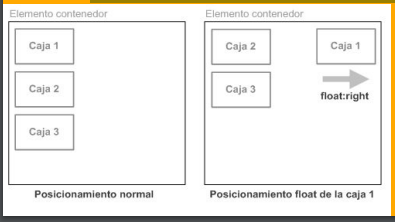
● display permite cambiar la forma de mostrar aunque no el tipo de elemento.

Float

● float permite sacar a un elemento del flujo normal del documento.

● El elemento puede ser posicionado a la izquierda o derecha de su misma línea o

contenedor tanto como sea posible.



Float

● Admite tres valores, right, left y none.

● Fuerza la propiedad display del elemento a block.

● Los elementos que se encuentran a continuación de una caja flotantes adaptan sus

contenidos para que fluyan (rodeen) alrededor del elemento.

● Los elementos anteriores al flotante no cambian su comportamiento.

● Debe aplicarse sobre elementos con un ancho definido explícitamente o implícitamente.

Clear

● La propiedad clear se utiliza conjuntamente con float para forzar a que un elemento no

fluya alrededor de otro elemento posicionado mediante float.

● Sus valores posibles son:

○ left: no permite flotantes a su izquierda

○ right: no permite flotantes a su derecha

○ both: no permite flotantes a ambos lados

○ none: permite otros flotantes a ambos lados

Position

● Especifica el tipo de posicionamiento.

● Sus posibles valores son:

○ static: valor por defecto.

○ relative: posicionado relativo al contenedor.

○ fixed: posicionado relativo al viewport\*.

○ absolute: posicionado relativo al ancestro

posicionado\*\* más cercano.

○ sticky: posicionado relativo a la posición de desplazamiento del usuario.

● Luego, los elementos posicionados\*\* se colocan utilizando las propiedades top,

bottom, left, y right.

Visualización

● CSS define cuatro propiedades para controlar la visualización.

● Display, visibility, overflow y z-index.

● Con visibility:hidden; se puede invisibilizar un elemento.

● El navegador crea la caja del elemento pero no la muestra. El resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.

● Los valores para la propiedad son: visible,

hidden y inherit.

z-index

● Cuando los elementos se colocan, pueden superponerse a otros elementos.

● z-index especifica el orden de apilamiento de un elemento (qué elemento debe colocarse

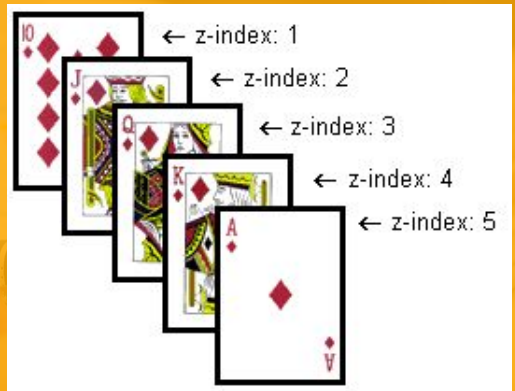
delante o detrás de los demás).

● Un elemento puede tener un orden de pila positivo o negativo.

● Solo funciona en elementos posicionados o elementos flex.

● Si dos elementos tienen el mismo z-index, el último elemento definido en el código HTML se

mostrará en la parte superior.



Overflow

● Controla qué sucede cuando los elementos sobrepasan el límite del contenedor.

● Normalmente cuando el contenedor tiene seteada las propiedades width y/o height.

● Sus posibles valores son:

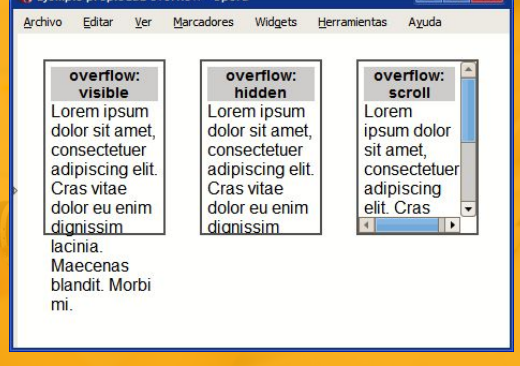
○ visible: desborde sin recortes, se renderiza fuera del contenedor. (Default).

○ hidden: desborde recortado y ocultado.

○ scroll: desborde recortado pero se muestran barras de desplazamiento.

○ auto: desborde recortado pero se muestran barras de desplazamiento solo si

es necesario



Flexbox

● El módulo de diseño de Flexbox facilita el diseño de una estructura de diseño flexible y receptiva sin usar flotación o posicionamiento.

● Para comenzar a usar el modelo Flexbox, primero debe definir un contenedor flexible estableciendo su propiedad display con el valor flex.

Contenedor Flexbox

● flex-direction: define en qué dirección el contenedor quiere apilar los elementos

flexibles (row, column y variantes -reverse).

● flex-wrap: especifica si ajusta los elementos flexibles (wrap, wrap-reverse y nowrap).

● flex-flow: (flex-direction + flex-wrap).

● justify-content: se utiliza para alinear los elementos flexibles (eje main).

● align-items: se utiliza para alinear los elementos flexibles (eje cross).

● align-content: similar al anterior pero alinea las líneas.

Ítems Flexbox

● Son los hijos directos de un contenedor flexbox.

● order: especifica el orden.

● flex-grow: especifica cuánto crecerá un elemento flexible en relación con el resto de

los elementos flexibles.

● flex-shrink: similar al anterior pero al achicar.

● flex-basis: especifica la longitud inicial.

● flex: flex-grow + flex-shrink + flex-basis,

● align-self: especifica la alineación del elemento

