Unidad 2 – Uso de estilos.

2.1. Introducción.

CSS Cascading Style Sheets, hojas de estilo en cascada. Permiten separa en el desarrollo de un sitio web el diseño (aspecto o apariencia) de los contenidos (información). Desarrollo y mantenimientos más eficientes.

- Definir las estructuras HTML <body>
- Definir la apariencia HTML <head>
- ¿Y los contenidos?...

Es un estándar de W3C.

- CSS1 recomendación del W3C, año 1996.
- CSS2 recomendación del W3C, año 1998.
- CSS3 recomendación del W3C, año 2011.

Reglas de estilos que se aplican a aquellos elementos o partes en función de un determinado selector.

2.2. Selectores basados en etiquetas, en clases y en identificadores.

2.2.1. Basados en etiquetas:

selector { atributo : valor };

Separando varios selectores por comas aplicamos el mismo estilo.

2.2.2. Basados en clases:

selector.nombre-clase { atributo : valor };

.nombre-clase { atributo : valor }; (Más genérica)

2.2.3. Basados en identificadores:

selector#nombre-identificador { atributo : valor};

#nombre-identificador { atributo : valor }; (Más genérica)

Identificador es distinto de clase. Identificador para objetos específicos, clase para elementos genéricos.

2.3. Agrupamiento y anidamiento.

2.3.1. Agrupamiento (,).

selector1, selector2 { atributo1 : valor1; atributo2 : valor2 };

2.3.2. Anidamiento común ( ).

selector1 selector2 { atributo1 : valor1; atributo2 : valor2 };

2.3.3. Anidamiento de selectores hijos (>).

selector1 > selector2 { atributo1 : valor1; atributo2 : valor2 };

2.3.4. Anidamiento de selectores adyacentes (+).

```
selector1 + selector2 { atributo1 : valor1; atributo2 : valor2 };
```

2.4. Buenas prácticas al escribir CSS.

- Selectores en minúsculas, nunca empezando con caracteres especiales o numéricos.
- Nombre específico y claro, descriptivo.
- Nombres de clases o identificadores no deben incluir características.
- Nombres con visión semántica más que estructural.
- Separar las palabras mediante guiones o mayúsculas, legibilidad.
- No hacer uso excesivo de clases, dejarlo para las partes relevantes de la estructura.
- Documentar apropiadamente para el futuro u otros /* comentario */
- Agrupar las reglas según su selector siempre que sea posible.
- Es aconsejable definir al principio los selectores de etiquetas.
- Estructurar visualmente los atributos.

2.5. Atributos. Modelo de cajas.

- Atributos de posición.
- Atributos de márgenes (borde externo a la caja).
- Atributos de relleno (espacio interno para separar márgenes del contenido).
- Atributos de bordes (definen las líneas gráficas alrededor de la caja).

2.6. Colores de fondo, textos, enlaces, listas, tablas, visibilidad, imágenes.

- Atributos de fuentes.
- Atributos de párrafos.
- Atributos de tablas.
- Atributos de visibilidad.
- Atributos de listas.
- Atributos de enlaces.

2.7. Unidades de medida

- pulgadas 1” = 2,54 cm
- centímetros 1/100 parte del metro Sistema Internacional de Unidades (SI)
- milímetros 1/1000 parte del metro Sistema Internacional de Unidades (SI)
- puntos 1pt = 1/72” (ratio)
- picas 1pc = 12 pt
- em es igual al valor calculado de la letra mayúscula "M" de la propiedad de tamaño de fuente del elemento en el que se utiliza.
- rem es igual al valor calculado de la letra mayúscula "M" del tamaño de fuente en el elemento raíz (html).

Unidades rem y em se calculan en valores de pixel por el navegador, basados en tamaños de fuentes en tu diseño.

- Unidades em están basadas en el tamaño de la fuente del elemento en que se encuentran.
- Unidades rem están basadas en el tamaño de la fuente del elemento html.
- Unidades em pueden ser influenciadas por la herencia del tamaño de la fuente de cualquier elemento padre.
- Unidades rem pueden ser influenciadas por la herencia del tamaño de la fuente de la configuración de la fuente del navegador.
- Usa unidades em para tamaño que deberían escalarse dependiendo del tamaño de la fuente de un elemento además de la raíz.
- Usa unidades rem para tamaño que no necesitan unidades em, y que deberían escalar dependiendo de la configuración del tamaño de la fuente del navegador.
- Usa unidades rem, a menos que estés seguro que necesitas unidades em, incluyendo en tamaños de fuente.
- Usa unidades rem en Media Queries
- No uses em o rem en anchuras de maquetados multi-columna- usa % mejor.
- No uses em o rem si escalar inevitablemente causará que se desacomode un elemento del maquetado.

2.8. Superposición y precedencia de estilos.

Profundidad:

En determinadas circunstancias, varios elementos puedes superponerse unos a otros. La profundidad y por tanto el orden de superposición puede controlarse con la propiedad:

z-index: [numero]

Valor entero más alto supone estar más arriba en la pila de superposición y por tanto por encima de los valores inferiores. Están permitidos los valores negativos, aunque habitualmente suele tomarse 0 como el nivel más bajo.

Se relaciona con la propiedad o atributo que determina la posición de una caja respecto a las demás:

position : [static] [absolute] [relative] [fixed] [inherit]

Orden de precedencia según la especificidad.

1. CSS en línea, atributo de estilo HTML anula reglas de hoja de estilo.
2. Un selector más específico es prioritario sobre uno menos específico.
3. A igualdad de prioridad de las reglas anteriores, reglas posteriores anulan a las anteriores.
4. Una regla con !important siempre tiene prioridad.

2.9. Crear y vincular hojas de estilo

- style -> Reglas integradas en el propio HTML.

  e.g.: <p style = “atributo : valor;”>Minions ipsum tempor magna gelatooo.</p>

- <style> -> Reglas incrustadas dentro del fichero, en la cabecera.

  e.g.: <head>

  <style type=”text/css” media=”screen” title=”Mi-estilo”> atributo : valor;

  </style></head>

  * type -> el valor “text/css” es por defecto en HTML5.
  * media -> dispositivo [all] [print] [screen] [speech] en HTML5 (los demás tipos de dispositivos anteriores están en desuso).
  * title -> nombre del estilo e.g.: Mi-estilo.

- <link> -> Reglas externas, enlazadas en la cabecera.

  e.g.: <head>

  <link rel=”stylesheet” type=”text/css” media=”screen” title=”Mi-estilo” /> </head>

  * type -> igual que en el anterior.
  * media -> igual que en el anterior.
  * title -> nombre del estilo, enlazando varias hojas de estilo con el mismo nombre de estilo se logra una hoja de estilo múltiple, combinación de las distintas hojas de estilo.
  * rel -> relación entre documentos
    - [stylesheet] define un estilo persistente (el que se aplica si están activas las hojas de estilo) o preferido (es persistente y además tiene title, sólo puede se puede especificar uno).
    - [alternate stylesheet] define un estilo alternativo.

- @import -> Importar reglas, en la cabecera.

  e.g.: <head>

  <style> @import url(“estilo.css”) screen; </style>

  </head>

  * No soportado por todos los agentes de usuario, antiguamente.
  * No permite hojas de estilo preferentes / alternativas / preferidas.
  * Permite elegir según medio o características (Media Queries).

2.10. Selectores avanzados.

2.10.1. Combinators

Ver más información y ejemplos: https://www.w3schools.com/css/css_combinators.asp

Descendiente	Hijo	Hermano adyacente	Hermano general
( ) espacio	(>)	(+)	(~)

2.10.2. Pseudo clases (:)

Ver más información y ejemplos: https://www.w3schools.com/css/css_pseudo_classes.asp

Sintaxis:

selector : pseudo-clase { propiedad: valor; }

selector.clase : pseudo-clase { }

Pseudo clases con el selector <a>.

:link | :visited | :hover | :active

(en ese orden para que sea efectiva la definición)

Pseudo clases con el selector .

:hover

Pseudo clases de hijos de un elemento.

:first-child

:last-child

Pseudo clases de elementos en un determinado idioma.

:lang

2.10.3. Pseudo elementos (::).

Ver más información y ejemplos: https://www.w3schools.com/css/css_pseudo_elements.asp

Sintaxis:

selector :: pseudo-elemento { propiedad: valor; }

selector.clase :: pseudo-elemento { }

Nota:

:: first-line vs *: first-line* (Ejemplo de notación)

Los dos puntos dobles reemplazaron la notación de dos puntos para los pseudo-elementos en CSS3. Este fue un intento del W3C para distinguir entre pseudo-clases y pseudo-elementos. Para compatibilidad con versiones anteriores, la sintaxis de un único dos puntos es aceptable para pseudo-elementos CSS2 y CSS1.

Pseudo elementos relevantes:

:: first-line

:: first-letter

:: before

:: after

:: selection

2.10.4. Selectores de atributo

Ver más información y ejemplos: https://www.w3schools.com/css/css_pseudo_elements.asp

Usados para dar estilo a los formularios sin clases ni id.

Sintaxis:

- [atributo] -> atributo específico
- [atributo = “valor”] -> atributo y valor específico.
- [atributo ~= “valor”] -> atributo que contenga un valor.
- [atributo |= “valor”] -> atributo que sea valor o empiece por valor seguido de un guión.
- [atributo ^= “valor”] -> atributo que empiece por valor.
- [atributo $= “valor”] -> atributo que termine por valor.
- [atributo *= “valor”] -> atributo que contenga valor.

2.11. Opacidad y transparencia. Color.

Ver más información: https://www.w3schools.com/css/css_image_transparency.asp

- Propiedad opacity con un valor entre 0.0 – 1.0 (donde 0.0 es completamente transparente y 1.0 es completamente opaco).
- Propiedad opacity con efecto, asociada al selector :hover.
- Propiedad de transparencia, no heredable, con colores RGBA (alpha channel) con un valor entre 0.0 – 1.0 (donde 0.0 es completamente transparente y 1.0 es completamente opaco).

  rgba (255, 255, 255, 1.0) blanco opaco

  rgba (0, 0, 0, 0.0) negro transparente

Colores

Ver más información: https://encycolorpedia.es/

2.12. HTML5. Migración.

Ver más información:

https://www.w3schools.com/html/html5_intro.asp

https://www.w3schools.com/html/html5_new_elements.asp

https://www.w3schools.com/html/html5_semantic_elements.asp

https://www.w3schools.com/html/html5_migration.asp

https://www.w3schools.com/html/html5_syntax.asp

- ¿Qué hay de nuevo?
- Nuevos elementos.
- Nuevos API's.
- Elementos eliminados.
- Migración.

Typical HTML4	Typical HTML5
	<header>
	<nav>
	<section>
	<article>
	<footer>

2.13. CSS avanzados.

Ver ejemplos: https://www.w3schools.com/css/

2.13.1. Navigation bar (barra de navegación).

Ver ejemplos:

https://www.w3schools.com/css/css_navbar.asp

https://www.w3schools.com/css/css_dropdowns.asp

Lista de links: <ul>

- Vertical: (display: block).

  Link activo (actual).

  Centrar links y añadir bordes.

  Altura completa, fijar barra de navegación.

- Horizontal: (display: float / inline).

Link activo (actual).

Alinear a la derecha los links.

Bordes divisores.

Fijar la barra de navegación.

Barra de navegación pegajosa (sticky).

* Navegación superior adaptable.
* Navegación lateral adaptable.
* Navegación desplegable.
    - Desplegable básico.
    - Menú desplegable.
    - Desplegable de imagen.
    - Barra de navegación desplegable

2.13.2. Flexbox

Ver ejemplos: https://www.w3schools.com/css/css3_flexbox.asp

Sin flotación ni posicionamiento. Pasos para lograrlo:

1. Definir un contenedor flexible:

    display: flex

2. Propiedad de apilado de elementos flexibles:

    flex-direction: [column] [column-reverse] [row] [row-reverse]

3. Propiedad de salto de línea flexible:

    flex-wrap: [wrap] [nowrap] [wrap-reverse]

4. Propiedades apilado + salto, juntas (en ese orden):

    flex-flow: row wrap (por ejemplo)

5. Propiedad de justificación del contenido para alinear elementos Flex:

    justify-content: [flex-start] [flex-end] [center] [space-around] [space-between]

6. Propiedad de alineación de elementos flexibles verticalmente:

    align-items: [flex-start] [flex-end] [center] [stretch] [baseline]

7. Propiedad de alineación de las líneas flexibles:

    align-content: [stretch] [center] [space-around] [space-between] [flex-start] [flex-end]

2.13.2.1. ¿Cómo lograr el centrado perfecto?

justify-content: center;

align-items: center;

2.13.2.2. Hijos de elementos flexibles

Los hijos directos de contenedores flexibles se convierten en elementos flexibles.

Propiedades de los elementos Flex:

- Orden de los elementos flexibles (permite cambiar el orden):

  order: [0, 1, 2, 3...] (número)

- Crecimiento relativo respecto al resto de elementos flexibles:

  flex-grow: [0, 1, 2, 3...] (número)

- Encogimiento relativo que puede soportar respecto al resto de los elementos:

  flex-shrink: [0, 1, 2, 3...] (número)

- Longitud inicial de un elemento flexible:

  flex-basis: [auto, number] (longitud unidad o %, px, em…)

- Propiedad crecimiento + encogimiento + longitud inicial, juntas (en ese orden):

  flex: [flex-grow flex-shrink flex-basis]

  [auto -> 1 1 auto]

  [initial -> 0 1 auto]

  [none -> 0 0 auto]

- Propiedad de alineamiento de un elemento seleccionado dentro del contenedor Flex:

  align-self: [auto] [stretch] [center] [flex-start] [flex-end] [baseline]

  Sobreescribe propiedad align-items de contenedor.

2.13.3. Layout, display inline vs display inline block vs display block.

display: inline Elemento en línea; no permite width, height.	*display: inline-block* Permite width, height del elemento; respeta top y bottom; respeta margin y padding.
display: block Elemento en bloque; con salto de línea; ocupa todo el ancho.	*display: inline-block* Elemento en bloque que ocupa todo el ancho pero sin salto de línea.

2.13.4. Grid layout.

Ver ejemplos: https://www.w3schools.com/css/css_grid.asp

Retícula o cuadrícula:

display: grid / display: inline-grid

Conceptos:

Rows | Columns | Gaps

Propiedades de filas, columnas y espacio:

- Propiedad de espacio entre columnas:

  grid-column-gap

- Propiedad de espacio entre filas:

  grid-row-gap

- Propiedad de atajo para los dos espacios, que se puede unificar en un solo valor:

  grid-gap

Grid lines:

Row lines | Column lines

Propiedades para definir los elementos respecto a las líneas:

- Propiedad grid-column-start.
- Propiedad grid-column-end.
- Propiedad grid-row-start.
- Propiedad grid- row-end.

Propiedades para inicializar el layout:

- Propiedad de número de columnas del layout y la anchura de cada una de ellas:

  grid-template-columns: [auto / number]

- Propiedad de la altura de cada fila:

  grid-template-rows: [number]

Propiedades para alinear:

- Propiedad para alinear toda la cuadrícula dentro del contenedor:

  justify-content: [space-evenly] espacio igual

  [space-around] [space-between] [start] [end] [center]

- Propiedad para alinear verticalmente toda la cuadrícula dentro del contenedor:

  align-content: [space- evenly] [space-around] [space-between] [start] [end] [center]

2.13.4.1. Hijos de elementos grid.

Los hijos de un contenedor grid, son elementos grid. Por defecto un contenedor grid tiene un elemento por cada columna en cada fila, pero se puede hacer que se amplíen para ocupar múltiples columnas y/o filas.

- Propiedad para definir en qué columna se coloca un elemento. Se define dónde comienza y dónde acaba (/). Atajo de grid-column-start y grid-column-end (en ese orden):

  grid-column: [line numbers] [span number]

  e.g.: grid-column: 1 / 5

  e.g.: grid-column: 1 / span 3

- Propiedad para definir en qué fila se coloca un elemento. Se define dónde comienza y dónde acaba (/). Atajo de grid-row-start y grid-row-end (en ese orden):

  grid-row: [row-line numbers] [span number]

  e.g.: grid-row: 1 / 4

  e.g.: grid-row: 1 / span 3

- Propiedad atajo de grid-row-start, grid-column-start, grid-row-end, grid-column-end (en ese orden):

  grid-area: [row line numbers | column line numbers] [span number]

Con la propiedad grid-area se pueden nombrar elementos que pueden ser referenciados con la propiedad grid-template-areas del contenedor.

grid-area: nombre;

grid-template-areas: 'nombre nombre nombre';

(sería un elemento llamado *nombre* que ocuparía 1 fila y 3 columnas)

Cada fila se define por apostrofes ' '. Las columnas en cada fila definida por los apostrofes se separan por un espacio, un elemento sin nombre se representa con un punto.

Orden de los elementos

El diseño de cuadrícula permite posicionar los elementos donde se quieran. El primer elemento del código HTML, no tiene porqué aparecer el primero en la cuadrícula. Además, se pueden reordenar según el tamaño de la pantalla usando Media Queries.

2.14. Validación y verificación de CSS.

Validador HTML: https://validator.w3.org/

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