Apuntes examen teórico 2ª evaluación Lenguaje de Marcas

*Estilos css*

Definición de estilos:

● En línea: Dentro de la etiqueta HTML

● Incrustados: Dentro del documento HTML

● Enlazados/Importados: En un fichero independiente

Utilizar siempre tamaños de letra relativos, como em. Indicar los colores con números en lugar de con letra. Hacer desarrollo responsive.

Diferencia entre clase e id:

* Clase: la misma se puede repetir en varios elementos. Si utilizas su selector (.miclase) se incluirán todos los elementos en cuyo atributo class aparezca el nombre de la clase, sin importar si el nombre de la clase tiene otras cosas delante o detrás (class= “texto” y class= “texto-azul”, valen las dos).
* Id: identifica a un solo elemento, no se puede repetir. Si utilizas su selector (#id), se incluirán todos los elementos con atributo id igual a ese nombre, no se vale si tienen cosas delante o detrás, a diferencia de la clase.

Puedes concatenar selectores tipo: h1#menu.miclase.

Y puedes seleccionar varios selectores separándolos por comas: .miclase, .miotraclase

RECUERDA: los estilos definidos para IDs tienen prioridad sobre los definidos sobre una clase.

*Selectores de contexto*

Son selectores en función de la posición entre ellos y de cómo se escriban. Es decir:

* Sel1 sel2 … selN: se lee de forma inversa. Seleccionamos el selN que está dentro del sel2 que a su vez se incluye en el sel1.
* Sel1+sel2: selecciona el elemento sel2 definido inmediatamente a continuación del elemento sel1.
* Sel1>sel2: selecciona los elementos sel2 contenidos dentro de un elemento sel1, que se llamaría elemento padre.
* Sel1~sel2: selecciona los elementos de tipo sel2 que están definidos a raíz de la primera definición de sel1. CONFUSO: ver ejemplo 4.

*Pseudo-selectores*

Hacen referencia a estados del selector asociado, principalmente enlaces:

● :link

● :visited

● :hover

Hacen referencia a estados del selector asociado:

● :focus: este selector define una regla para el elemento HTML que “tiene el foco”, normalmente un elemento de formulario.

● :active: elecciona el elemento activado por el usuario, mediante ratón normalmente. Es un efecto “instantáneo”, ocurre en una mínima fracción de tiempo, haciéndolo prácticamente imperceptible

● :target: Selecciona el elemento “target” de un enlace interno, tipo ancla.

Hacen referencia a una parte del selector asociado:

● ::first-line

● ::first-letter

● ::before

● ::after

Hacen referencia a la posición del elemento:

● :first-child: Selecciona el elemento de ese tipo que sea primer elemento hijo de su contenedor.

● :last-child: Selecciona el elemento de ese tipo que sea el último elemento hijo de su contenedor.

● :only-child: Selecciona el elemento de ese tipo que sea el único elemento hijo de su contenedor.

● :nth-child(N): Selecciona el elemento de ese tipo que sea el hijo N de su contenedor.

Hacen referencia a la posición del elemento:

● :nth-of-type(N)

● :nth-last-of-type(N)

● :first-of-type

● :last-of-type

Permiten seleccionar un elemento según el valor de sus atributos:

● Elemento[atributo =”valor”]: Permite definir reglas para aquellos elementos cuyo atributo sea igual a “valor”.

● Elemento[atributo ^ =”valor”]: Permite definir reglas para aquellos elementos cuyo atributo comience por la cadena “valor”.

● Elemento[atributo$=”valor”]: Permite definir reglas para aquellos elementos cuyo atributo termine por la cadena “valor”.

● Elemento[atributo\*=”valor”]: Permite definir reglas para aquellos elementos cuyo atributo contenga la cadena “valor”.

*Atributos de texto*

Font-family:

Se puede usar cualquier tipo de fuentes disponible en la carpeta web del servidor, o desde alguna librería de fuentes online.

● **@font-face:** es una regla css que permite descargar una fuente en particular de su servidor parautilizar en una web si el usuario no la tiene instalada.

● **@import url():** Se puede usar para importar un archivo CSS de otra web, pero también se usa para importar fuentes de librerías públicas online (por ejemplo, Google Fonts).

Font-size:

* em: es una medida relativa al tamaño de la letra del elemento padre.
* rem: es una medida relativa al tamaño de la letra del Body.

Font-weight (grosor de la fuente):

● normal: Valor por defecto.

● bold: permite definir una fuente en negrita simple.

● bolder: Define una fuente negrita de mayor grosor.

● lighter: permite definir una fuente de poco peso.

● 100-900: siendo 400 el peso normal, podemos utilizar 100 para fuentes con poco peso y 900 para fuentes muy negritas. El rango va de 100 en 100.

Font-style:

○ **normal**: es el estilo por defecto. El navegador muestra el texto en su estilo plano.

○ **italic**: es el estilo conocido como cursiva .

○ **oblique**: es prácticamente idéntico a italic. Este realmente usa el estilo normal y lo inclina, mientras que *italic* usa la versión itálica de la propia fuente.

Font-variant: pone versalitas.

Color: color de la fuente.

Text-align: alinea el texto (centrar, justificado, etc.)

Text-decoration (pa ponerle raya: subrayado, tachado, etc.)

Text-indent (sangría, con px o porcentaje)

Text-shadow (sombreado)

Text-transform (jugar con las mayúsculas, none, capitalize, uppercase, lowercase)

Line-height (interlineado)

Letter-spacing

Word-spacing

List-style-type (para cambiar el aspecto de los puntitos de la lista)

Type (para las listas ordenadas, cambia el tipo de numeración, en plan números romanos, números normales, etc.)

ETC.

Margin y padding la misma mierda son.

*Modelo de cajas*

Content, padding, border y margin.

Display inline: las cajas en línea toman el espacio según su contenido: van todas una detrás de otra, en línea. No admiten ni width ni height.

Display block: las cajas en bloque ocupan todo el espacio a lo ancho del contenedor. Es decir, un bloque por espacio horizontal. Su altura viene determinada por su contenido.

Display inline-block: una combinación de ambas. Los elementos admiten dimensiones, pero se colocarán uno al lado del otro.

Display none: oculta un elemento y la página se comporta como si no existiera. Pero si en lugar de eso usamos visibility: hidden, se hará invisible el elemento, aunque seguirá ocupando su espacio.

Con el DOM (Document Object Model), hacemos una representación del html en versión árbol genealógico, aunque de manera descendente. Cada rama representa un elemento del HTML.

*Diseño responsive*

Nos referimos a el uso de reglas CSS que adaptan el contenido de un documento HTML al tamaño de la pantalla desde el que se está visualizando.

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

**width=device-width:** establece el ancho de la página a usar, en este caso, el ancho de pantalla del dispositivo.

**initial-scale=1.0** establece el nivel de zoom inicial cuando la página se carga por primera vez en el navegador del dispositivo.

Media queries:

@media **screen** and (min-width: npx) and (max-width: npx){}

● **all** : indica que desea apuntar a todos los dispositivos, sin excepciones.

● **print** : se dirige a dispositivos que envían resultados a una pantalla de impresión como la ventana "Vista previa de impresión" en un navegador web.

● **screen** : esto es lo contrario de print , y se dirige a todos los dispositivos que no pertenecen a la categoría de print.

@media screen **and** (min-width: npx) and (max-width: npx){}

● **not**: si precede al media type, lo excluye de la selección.

● **and**: se usa para enlazar varias condiciones del selector de medios.

● only: se utilizaba tiempo atrás para navegadores que no se manejaban muy bien con los selectores @media.

Break points:

* 600px para móviles
* 768px para tablets
* 1200px para escritorios

Variables:

:root {

--color\_texto: xxxxxx;

--fuente\_texto: xxxxxx;

--color\_titulo: xxxxxx;

--fuente\_titulo: xxxxxxx;

}

Esta declaración de variables se ha de poner al inicio del css. Para utilizar la variable correspondiente, se hace así:

font-family: var(--fuente\_texto);

color: var(--color\_texto);

Y arreando.