

Lenguajes y estándares Web

PEC 3

CARLOS DÍAZ PÉREZ

NOTA: Los códigos presentes en el siguiente documento se pueden consultar directamente en los html adjuntos con nombres `primeraparte.html` y `pregunta32.html`

PRIMERA PARTE

Pregunta 1.

Sobre la utilización de tablas, responde a las siguientes cuestiones:

¿Para qué se utiliza el atributo scope? Pon un ejemplo de su uso.

En HTML5, el atributo scope se utiliza en los elementos th para indicar si la celda “header” seleccionada se trata de columna, fila, grupo de columnas o de filas.

Si estamos en HTML 4.01 se puede usar también en td.

```
<th scope="col">Mes</th>
```

¿De qué formas se puede eliminar el espacio que aparece entre las celdas de una tabla al aplicarles un borde?

Mediante CSS se puede utilizar la propiedad border-collapse: collapse, que elimina directamente el espacio entre bordes.

Pon un ejemplo que lo ilustre.

Todos los ejemplos los he representado en el html “primeraparte.html” dentro de la carpeta de la PEC.

Pregunta 2.

Sobre la utilización de formularios, responde a las siguientes cuestiones:

¿Qué atributos podemos utilizar para el elemento form? Describe brevemente para qué sirve cada uno de ellos, y pon un ejemplo de su uso.

Además de los atributos globales genéricos de HTML, la etiqueta form acepta también los siguientes atributos:

Atributo	Valor	Descripción
Accept	File_type	No valido en HTML5 Especifica una lista separada por comas de los archivos que acepta.
Accept-charset	character_set	Especifica el character encoding que se utilizará al enviar el formulario

Action	URL	Especifica donde hay que enviar la información cuando se envíe el formulario
Autocomplete Nuevo en HTML5	On / off	Especifica si el formulario debe tener autocompletar o no
Enctype	application/x-www-form-urlencoded multipart/form-data text/plain	Especifica como se deberá codificar la información del formulario cuando se envíe al servidor
Method	Get/post	Especifica el método http que se deberá usar al enviar datos de formulario
Name	Text	Especifica un nombre para el formulario
Novalidate Nuevo en HTML5	Novalidate	Especifica que el formulario no debe ser validado cuando se envía
Target	_blank _self _parent _top	Especifica donde se debe mostrar la respuesta que se recibe tras enviar el formulario

¿Para qué sirve el uso de una cuadrícula (grid) en el diseño de una página web? Pon un ejemplo de un formulario en el que se emplee una cuadrícula para distribuir sus controles.

La cuadrícula la usaremos para realizar un diseño ordenado y coherente, dispuestos a intervalos regulares y ordenados.

Un grid efectivo:

1. Hace que los márgenes de columna se sitúen a intervalos consistentes
2. Las secciones secuenciales de un mismo documento compartirán sus márgenes superiores con los elementos de columnas adyacentes
3. Las ilustraciones son enmascaradas o recortadas hasta respetar las anchuras de las columnas como los intervalos de la cuadrícula atómica
4. Incluso en los casos de tratarse de una sola columna con elementos float, estos cumplirán los estándares de tamaño y composición.

Comparativa uso de grid vs sin grid

WITH BASELINE ALIGNMENT

Lorem ipsum dolor sit amet,	Pellentesque finibus magna
consectetur adipiscing elit.	quis arcu malesuada, et
Sed hendrerit erat enim, non	varius elit ullamcorper.
pulvinar purus aliquam	Vivamus venenatis dolor dui,
tempus. Etiam et odio sit	ac vulputate ex consectetur
amet nisi interdum volutpat	eu. In hac habitasse platea
non in felis. Donec vehicula.	dictumst. Nulla facilisi.



WITHOUT BASELINE ALIGNMENT

Lorem ipsum dolor sit amet,	Pellentesque finibus magna
consectetur adipiscing elit.	quis arcu malesuada, et
Sed hendrerit erat enim, non	varius elit ullamcorper.
pulvinar purus aliquam	Vivamus venenatis dolor dui,
tempus. Etiam et odio sit	ac vulputate ex consectetur
amet nisi interdum volutpat	eu. In hac habitasse platea
non in felis. Donec vehicula.	dictumst. Nulla facilisi.



Imágenes extraídas de <https://designschool.canva.com/blog/grid-design/>

Pregunta 3.

Sobre elementos flotantes, responde a las siguientes cuestiones:

- 1. ¿Cómo podemos hacer que el tamaño de un elemento que contiene a varios elementos flotantes se amplíe para contener a estos (de manera que los hijos flotantes no "se salgan" de su elemento padre/contenedor)? Pon un ejemplo de cada uno de los métodos descritos, si hay más de uno.*

Dispondríamos de dos maneras de hacerlo. Por un lado, hacer que el **elemento padre** de aquellos elementos flotantes **también lo sea**, de esta manera conseguiremos que uno contenga a otro.

Si por alguna razón no podemos hacer que el elemento padre sea flotante, podemos utilizar la propiedad **overflow**, que si le asignamos el valor hidden sin una altura determinada ajustará su tamaño al de sus hijos.

- 2. Escribe el código HTML y el código CSS necesarios para conseguir tener en una página un menú de navegación horizontal de cuatro ítems en la parte superior de*

la página, dos columnas de contenido a continuación (debajo del menú) y un pie de página que se sitúe por debajo de los elementos anteriores.

HTML

```
<header>
  <h1>PÁGINA DE EJEMPLO PEC3</h1>
  <nav>
    <ul>
      <li><a href="" title="">UNO</a></li>
      <li><a href="" title="">DOS</a></li>
      <li><a href="" title="">TRES</a></li>
      <li><a href="" title="">CUATRO</a></li>
    </ul>
  </nav>
</header>
<article class="columnas">
  <div class="col col2_de_4">
    <h2>Seccion1</h2>
    <p>Seccion 1 con sus elementos dentro</p>
    <p></p>
  </div>
  <div class="col col2_de_4">
    <h2>Seccion2</h2>
    <p>Seccion 2 con sus elementos dentro</p>
  </div>
</article>
<footer>
  <h4>FIN DE LA PÁGINA DE EJEMPLO PEC 3</h4>
</footer>
```

CSS

```
body {
  background: #248ACE;
  color: #FFDB75;
  max-width: 60%;
  margin: auto;
}

footer, header {
  padding-top: 2em;
  background: #333399;
  text-align: center;
}

h4 {
  padding: 2em;
}

footer {
  clear: both;
}
```

```
}  
  
.columnas {  
  clear:both;  
  padding:0;  
  margin: auto;  
}  
  
.col {  
  display:block;  
  float:left;  
  margin:1% 0 1% 1.6%;  
}  
  
.col2_de_4 {  
  width:47%;  
}
```

SEGUNDA PARTE

HTML

Index.html

Header	Etiqueta con la que definimos el bloque de “encabezamiento”, con la imagen y la lista de navegación.
Nav	Etiqueta semántica con la que englobamos la lista / menú de la página
Ul	Listas de menú
Div	En este caso la he utilizado para englobar las secciones y poder darle estilos con CSS
Section	Utilizada para separar las tres columnas
H2	Etiquetas necesarias para cada sección en la que indico el título.
P	Etiqueta de párrafo
Footer	Etiqueta con la que defino la firma / pié de la pagina
span	Etiqueta correspondiente al elemento de línea con el que doy estilo al nombre de la plataforma en la firma

Libros.html

Table	Elemento tabla
Theader	Elemento donde definimos el header de la tabla
Tr	Table row, para definir contenido de una línea
Th	Definición te elemento “header” para tabla, que en combinación con “scope” podemos definir como “header columna” o “header tabla”
Tbody	Definición del contenido de la tabla
Td	Cada una de las celdas de la tabla. Usando “rowspan” podemos definir el ancho de cada una de las celdas en la tabla. Usando “colspan” podemos hacer lo mismo pero en lugar de tomar

Formulario.html

Form	Elemento formulario
Fieldset	Grupo de elementos dentro de un formulario con una serie de propiedades conjuntas
Legend	Titulo del fieldset
Label	Titulo del elemento input/select
Input	Elemento de entrada de información, dependiendo del atributo “type” puede ser “text” o “number” (hablo de este caso)
Select	Elemento que, en combinación con option, se utiliza para elaborar un menú desplegable
Atributo Required	Atributo que añadido al elemento INPUT del FORM indica qué campos son obligatorios rellenar en el formulario
Atributo Value	Atributo usado para darle un valor único a determinado elemento select del formulario

CSS

Tan solo comento aquellos elementos que considero nuevos y de interés

```
footer {  
  clear:both;  
  background: #130068;  
  color: #FFFFFF;  
  text-align: center;  
  padding: 2em 0 1em 0;  
}
```

En este caso uso “clear:both” para que el footer aparezca a continuación de aquellos elementos flotantes, ya sean a derecha o izquierda

```
.seccion {  
  float:left;  
  width:33%;  
}
```

Secciones centrales de INDEX.HTML, cada una al 33% de anchura del contenedor en el que se encuentran (div con clase “cuerpo-seccion)

```
.cuerpo-seccion {  
  display: flex;  
  padding:2em 0;  
}
```

En lugar de Display inline o display box, utilizamos flex en este caso para que las alturas se ajusten.

```
.sc-izq {  
  border-right: #1C1D6D 2px  
  solid;  
  text-align: justify;  
}
```

Clases que definen los estilos para cada una de las columnas.

```
.sc-ctr {  
  text-align: center;  
}
```

```
.sc-der {  
  border-left:#1C1D6D 2px  
  solid;  
  text-align: right;  
}
```



```
.col1d4 {  
  width:25%;  
  float:left;  
}
```

Desarrollo de un grid de diferentes dimensiones y tamaños que nos permitan colocar los elementos del formulario de una manera mas sencilla y alineados entre sí.

```
.col1d3 {  
  width:33%;  
  float:left;  
}
```

Cabe destacar la inclusión del elemento `display:flex` que nos iguala los tamaños de todos los elementos por cada fila.

```
.col2d4 {  
  width:50%;  
  float:left;  
}
```

```
.col3d4 {  
  width:75%;  
  float:left;  
}
```

```
.col1d5 {  
  width:20%;  
  float:left;  
}
```

```
.col1d20 {  
  width:5%;  
  float:left;  
}
```

```
.row {  
  display:flex;;  
  clear:both;  
  margin:auto;  
  padding:0;  
}
```