



PEC 3

Lenguajes y estándares web

Juan Salinas Gómez

Pregunta 1. Sobre la utilización de tablas, responde a las siguientes cuestiones:

1. ¿Para qué sirven los elementos `thead`, `tbody`, y `tfoot` en una tabla? Indica para cada uno si es obligatorio, si puede haber más de uno y en qué orden debe aparecer en la tabla.

El elemento `thead` contiene a un bloque de filas (`tr`) que representa a la sección de encabezado de una tabla (`table`), donde las celdas de encabezado (`th`) son encontradas habitualmente. Este elemento puede ser útil para especificar qué filas proveen información de encabezado para el resto de la tabla.

solo puede estar presente una vez en la tabla y debe ubicarse justo antes de su cuerpo.

La etiqueta de inicio es obligatoria. La etiqueta final se puede omitir si el elemento `<thead>` es seguido inmediatamente por un `<tbody>` o `<tfoot>`.

El elemento `tbody` contiene a un bloque de filas (`tr`) que representa a la sección del cuerpo de una tabla (`table`). Este elemento puede ser útil para especificar qué filas proveen información de contenido en la tabla, en contraste con aquellas que son parte del encabezado (`header`) o el pie (`footer`).

Pueden declararse múltiples cuerpos en una sola tabla, como una forma de separar temáticamente las filas de contenido.

El elemento `tbody` debe ser declarado luego de cualquier otro elemento de la tabla, con la excepción de `tfoot`.

El elemento `tfoot` contiene a un bloque de filas (`tr`) que representa a la sección de pie de una tabla (`table`). Este elemento puede ser útil para insertar celdas que provean, entre otros, resúmenes o valores promedio para la información de cada columna.

El elemento `tfoot` solo puede estar presente una vez en la tabla y debe ubicarse al final de esta, justo después de todos sus cuerpos.

A menos que estemos codificando una tabla realmente compleja con muchas columnas y filas de datos, si los utilizamos corremos el riesgo de excedernos.

2. Un estilo muy popular en la presentación de las tablas es el de las "líneas de cebra". Muestra mediante un ejemplo cómo se utiliza. E investiga en Internet si CSS3 aporta nuevas posibilidades para conseguir este estilo.

The screenshot shows a code editor with two panels: HTML and CSS. The HTML panel contains the following code:

```

1 <table>
2 <caption>
3   Prueba líneas de cebra
4 </caption>
5 <tr>
6 <th>Encabezado 1</th>
7 <th>Encabezado 2</th>
8 <th>Encabezado 3</th>
9 </tr>
10 <tr>
11 <td class="impar">contenido 1.1</td>
12 <td class="impar">contenido 1.2</td>
13 <td class="impar">contenido 1.3</td>
14 </tr>
15 <tr>
16 <td>contenido 2.1</td>
17 <td>contenido 2.2</td>
18 <td>contenido 2.3</td>
19 </tr>
20 <tr>
21 <td class="impar">contenido 3.1</td>
22 <td class="impar">contenido 3.2</td>
23 <td class="impar">contenido 3.3</td>
24 </tr>
25 <tr>
26 <td>contenido 4.1</td>
27 <td>contenido 4.2</td>
28 <td>contenido 4.3</td>
29 </tr>
30

```

The CSS panel contains the following code:

```

1 table {
2   border: #000 solid 1px;
3   border-collapse: collapse;
4 }
5 th {
6   border: #000 solid 1px;
7   padding: 0.5em;
8   background: #0066B3;
9 }
10 td {
11   border: #000 solid 1px;
12   padding: 0.5em;
13 }
14 .impar {
15   background: #BFE4FF;
16 }

```

Prueba líneas de cebra		
Encabezado 1	Encabezado 2	Encabezado 3
contenido 1.1	contenido 1.2	contenido 1.3
contenido 2.1	contenido 2.2	contenido 2.3
contenido 3.1	contenido 3.2	contenido 3.3
contenido 4.1	contenido 4.2	contenido 4.3

En CSS3 tenemos el selector `nth-child`: coincide con uno o más elementos en función de su posición entre un grupo de hermanos.

`nth-child(n)` permite seleccionar determinado elemento pero bajo la condición que sea hijo enésimo (n) de su padre.

La fórmula para usar este selector es `:nth-child(a+b)`, donde **n** es un contador que inicia en cero y avanza incrementalmente hasta que se le agoten los elementos hijos. La variable **b** representa el punto de partida del contador, es decir, la posición del primer elemento desde el cual se inicia el conteo. La variable **a** indica la posición de los elementos que deben afectarse luego de **b**. Ejemplo:

`:nth-child(4n)` → Selecciona cada cuarto elemento entre cualquier grupo de hermanos.

`:nth-child(2n+1)` → Selecciona las filas impares de una tabla.

`:nth-child(2n)` → Selecciona las filas pares de una tabla.

Pregunta 2. Sobre la utilización de formularios:

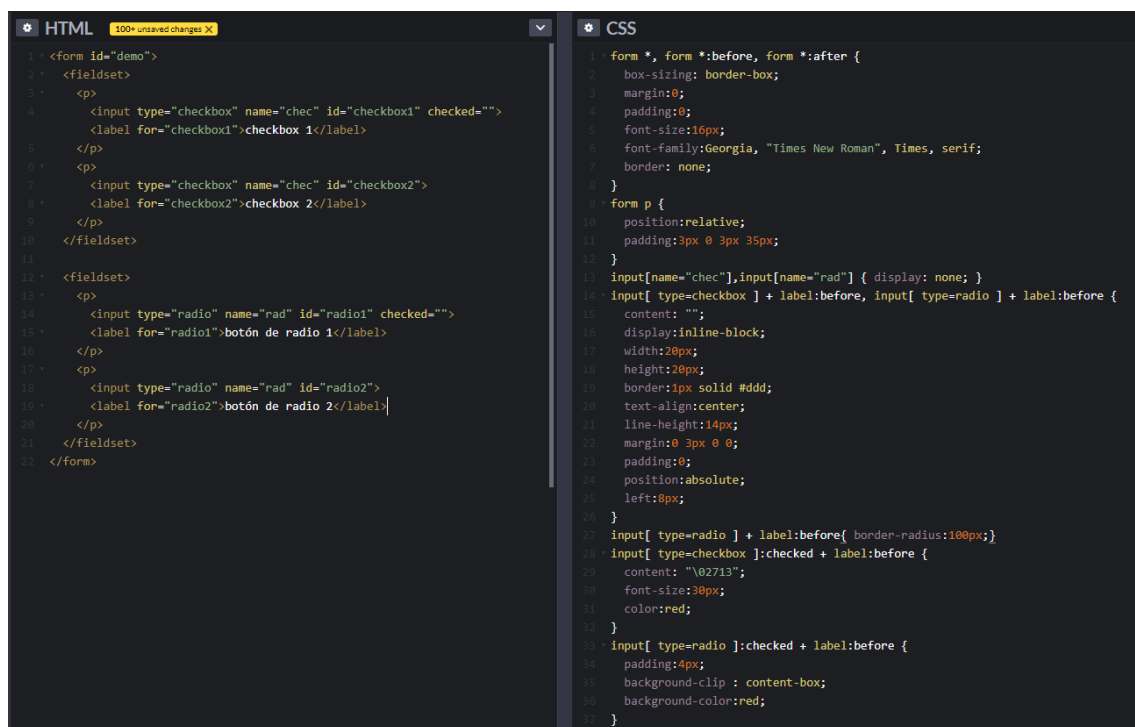
1. ¿Cómo se relaciona un control de un formulario (por ejemplo, un cuadro de texto, un botón de radio...) y su etiqueta? ¿Qué utilidad tiene hacerlo?

Para los usuarios lectores de pantalla cuando navegan a través de un formulario, a menudo no hay indicación de qué información se relaciona con qué elemento del formulario. Este problema se puede resolver mediante la asociación de las etiquetas de un formulario a elementos de la página. La etiqueta debe casi siempre figurar al lado del elemento del formulario. Cuando un lector de pantalla accede a un elemento del formulario que tiene un código <label> asociado a ella, se leerá el texto en el código <label> e indicará el tipo de elemento de formulario que es (por ejemplo, "Nombre. Cuadro de texto" o "Edad. Casilla de verificación "). Las etiquetas son necesarias para todos los elementos de formulario, a excepción de los botones, donde el lector de pantalla lee el texto que se encuentra en el botón.

2. Dos elementos a los que tradicionalmente se ha prestado poca atención a la hora de dar formato a un formulario son `checkbox` y `radio`. Pon un ejemplo de un formato algo más trabajado para estos elementos, e indica las propiedades CSS utilizadas (puedes investigar las posibilidades que ofrece CSS3 al respecto).

Buscando información por internet he encontrado el pseudoelemento `:before` y `:checked` para poder dar formato a los `checkbox` y `radio`.

En muchos casos de los que he encontrado añadían un elemento `span` al `label` para poder incluir una imagen, pero me ha gustado más este método en el que directamente modifica los elementos.

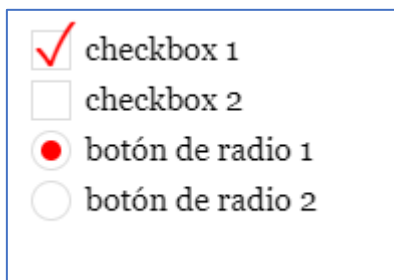


```

HTML
1 <form id="demo">
2 <fieldset>
3 <p>
4 <input type="checkbox" name="chec" id="checkbox1" checked="">
5 <label for="checkbox1">checkbox 1</label>
6 </p>
7 <p>
8 <input type="checkbox" name="chec" id="checkbox2">
9 <label for="checkbox2">checkbox 2</label>
10 </p>
11 </fieldset>
12 <fieldset>
13 <p>
14 <input type="radio" name="rad" id="radio1" checked="">
15 <label for="radio1">botón de radio 1</label>
16 </p>
17 <p>
18 <input type="radio" name="rad" id="radio2">
19 <label for="radio2">botón de radio 2</label>
20 </p>
21 </fieldset>
22 </form>

CSS
1 form *, form *:before, form *:after {
2 box-sizing: border-box;
3 margin:0;
4 padding:0;
5 font-size:16px;
6 font-family:Georgia, "Times New Roman", Times, serif;
7 border: none;
8 }
9 form p {
10 position:relative;
11 padding:3px 0 3px 35px;
12 }
13 input[name="chec"],input[name="rad"] { display: none; }
14 input[ type=checkbox ] + label:before, input[ type=radio ] + label:before {
15 content: "";
16 display:inline-block;
17 width:20px;
18 height:20px;
19 border:1px solid #ddd;
20 text-align:center;
21 line-height:14px;
22 margin:0 3px 0 0;
23 padding:0;
24 position:absolute;
25 left:8px;
26 }
27 input[ type=radio ] + label:before{ border-radius:100px;}
28 input[ type=checkbox ]:checked + label:before {
29 content: "\02713";
30 font-size:30px;
31 color:red;
32 }
33 input[ type=radio ]:checked + label:before {
34 padding:4px;
35 background-clip : content-box;
36 background-color:red;
37 }
  
```

Y el resultado es este:



checkbox 1

checkbox 2

botón de radio 1

botón de radio 2

Pregunta 3. Sobre Layout, responde a la siguiente cuestión:

1. Investiga en Internet sobre **flexbox** (modelo de caja flexible) y explica al menos 3 diferencias respecto a posicionar los elementos con **float** (modelo bloque). Escribe un ejemplo de código de cómo crearías, **utilizando flexbox**, un menú de navegación con las siguientes características:

- Tres elementos ("Inicio", "Conócenos", "Contacto")
- Fondo azul claro
- Altura de 100 píxeles
- Anchura de 100% con un máximo de 1024px
- Que los elementos se distribuyan de forma uniforme abarcando toda la anchura del menú
- Que los elementos estén alineados verticalmente en el centro

Algunas diferencias entre float y flexbox pueden ser:

- 1- Permite igualar las alturas de distintas columnas sin necesidad de usar falsas columnas.
- 2- Puede ajustar automáticamente la anchura de los elementos para colocarlos en una fila.
- 3- Viene bien para reubicar elementos en distintos tamaños de pantalla, ya que permite desplazar y reordenar elementos automáticamente si no caben en el ancho asignado.

Y el menú de navegación quedaría de la siguiente forma:

HTML

```

1 <div>
2 <ul>
3 <li>Inicio</li>
4 <li>Conócenos</li>
5 <li>Contacto</li>
6 </ul>
7 </div>

```

CSS 100+ unsaved changes

```

1 div {
2   background: #33cccc;
3 }
4
5 ul {
6   margin: auto;
7   padding: 0 10px;
8   width: 100%;
9   max-width: 1024px;
10  height: 100px;
11  list-style: none;
12  display: flex;
13  flex-direction: row;
14  border: 1px solid;
15  justify-content: space-between;
16  align-items: center;
17 }
18

```



Explicación de las entidades HTML y CSS utilizadas

Deberás añadir en el documento de texto donde habrás respondido las preguntas de la primera parte una explicación de las entidades HTML y CSS utilizadas:

1. Sobre las entidades HTML: debe explicarse el uso de etiquetas elegidas, excepto aquellas que especifican la estructura de la página (`doctype`, `html`, `head` y `body`)

`header` → Para definir la cabecera de la página.

`nav` → Utilizado para agrupar el listado que marca el menú principal de navegación.

`a` → Para los enlaces. Va con el atributo `href` para indicar el sitio con el que enlaza, el atributo `title` para describir lo que se va a encontrar en la página.

`img` → En el primer caso que se usa dentro del menú de navegación es para incorporar el logo en ese menú, además va entre las etiquetas `<a>` para que funcione como un enlace a la página de inicio. En el resto para insertar una imagen. Se le han añadido los atributos de `alt` para mostrar el texto alternativo a la imagen, el `title` para que al pasar el ratón por encima se muestre una pequeña descripción y `width` y `height` para las medidas de la imagen.

`ul`, `li` → Usado para definir el menú principal, en el primer caso de la página, y para listas no ordenadas en el resto.

`main` → Utilizado para englobar el contenido más importante o principal de la página.

`figure`, `figcaption` → Para insertar una imagen y un pie de foto que, en mi opinión, no son imprescindibles para que la página quede clara.

`article` → Lo he utilizado para englobar todo el cuerpo del texto de la página.

`h1` → Para los títulos de primer nivel.

`p` → Los he usado para distinguir los párrafos. Además, a varios les he añadido el atributo `id` para darles estilos distintos del resto.

`h2` → Para los títulos de segundo nivel.

`strong` → Para remarcar el texto en negrita.

`footer` → Para indicar el pie de página.

`span` → Lo he usado con el atributo `class` para darle estilos.

`cite` → Para marcar una referencia a la fuente/autor del logo indicado en los créditos del pie de página.

`table` → Elemento de Tabla HTML representa datos en dos o más dimensiones.

caption → Usado para darle un título descriptivo a las tablas.

thead → Utilizado para agrupar las filas que presentan información de encabezado.

tbody → Usado para representar un bloque de filas que consiste en un cuerpo de datos.

tfoot → Usado para contener a las filas que representan un pie o resumen.

tr → Define una fila de celdas en una tabla.

th → Define una celda como encabezado de un grupo de celdas. En algunos casos se le han añadido atributos como rowspan (para combinar verticalmente -filas- varias celdas adyacentes en una tabla), colspan (para combinar horizontalmente -columnas- varias celdas adyacentes en una tabla) y scope (para indicar explícitamente a qué celdas afecta una celda de encabezado.)

td → Para representar a una celda de datos de una tabla. En algún caso se le ha añadido el atributo class para aplicar estilos y colspan para que la celda ocupara más espacio que una celda.

form → El elemento form inserta un componente diseñado para contener controles con los que el usuario puede interactuar para enviar información de regreso al servidor, es decir, un formulario.

fieldset → Lo he usado para agrupar los diferentes campos que se solicitan para rellenar en grupos.

legend → Utilizado para crear un título para las agrupaciones hechas con fieldset.

label → Representa una etiqueta que puede ser asociada a un control de formulario, y que se supone provee una descripción corta para éste. Se pueden asociar de dos formas la etiqueta con el control, insertando dentro de la etiqueta el texto y el control o coincidiendo el valor del atributo id en el control con el del atributo for en la etiqueta. En mi caso he usado la segunda opción.

input → Se usa para crear controles interactivos para formularios basados en la web, que reciban datos del usuario.

select → Representa un control que permite al usuario elegir de entre un número de opciones. Lo he usado para crear los campos del día y la hora de entrega.

optgroup → Para agrupar en bloques diferentes las distintas opciones que se muestran en el elemento select, como si fuera un grupo de opciones.

option → Para representar un ítem u opción dentro del elemento select.

2. Sobre el CSS: debe explicarse todo el CSS utilizado, incluyendo cuando se han utilizado y por qué y para qué se aplican los atributos definidos.

```
@import url('https://fonts.googleapis.com/css?family=Roboto');
```

Este import se usa para que se vea la fuente de tipo Roboto (importar la fuente).

```
body {
  background: #6dffa3;
  font-family: 'Roboto', sans-serif;
  font-size: 16px;
  margin: 0;
  padding: 0;
}
```

Para el body he usado background para darle el color de fondo a la página. El margin y el padding los he puesto a 0 para que se aplique el color a toda la página y la ocupe todo el body. He aplicado el font-size para que este tamaño se use en toda la página y el font-family para usar la letra sans-serif (o sin rabito).

```
header {
  background-color: #fff;
  border-bottom: 5px solid #006800;
  height: 72px;
}
```

El background-color para darle un color a la cabecera donde está el menú de navegación, con un borde inferior (border-bottom) y una altura (height) para insertar la imagen y el menú.

```
nav {
  max-width: 940px;
  margin: auto;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

Con max-width se le ha limitado el espacio de ancho tal como como se pedía en la práctica, el margin en auto es para que se centre en la pantalla y he usado varias propiedades de flex-box: el valor flex en la propiedad display es para crear el contenedor flex, la propiedad justify-content con el valor space-between para que, en el espacio que tienen, los ítems se alineen y haya un espacio equitativo entre cada ítem. align-items: center es para alinear los ítems en el centro de la altura del contenedor. Esto se ha aplicado al elemento nav.

```
nav li {
  list-style: none;
  display: inline;
  font-size: 1.2em;
  font-weight: bold;
}
```

Para los descendientes de nav que sean elementos de una lista (li) se les ha quitado el pico, display: inline para poner los elementos de una lista en horizontal, el font-size para aumentar un poco el tamaño de la fuente y el font-weight para poner la fuente en negrita.

```
ul a {
  text-decoration: none;
  color: #006c00;
  padding: 15px;
}
```

Descendientes de una lista no ordenada que sean enlaces, se les quita el subrayado que muestran los enlaces con la propiedad `text-decoration: none`, se les da el color deseado (color) y se les aumenta el padding.

```
#actual > a, nav #actual > a:hover {
  background-color: #fff;
  color: #00da00;
  border-bottom: 5px solid #00da00;
  padding-bottom: 22px;
}
```

Se aplican los mismos estilos a los enlaces que sean hijos del id actual y a los enlaces, cuando el ratón está por encima, que son hijos del id actual y que este es descendiente del elemento `nav`. Esto lo hice para que, cuando está seleccionado un elemento del menú, este elemento no cambie el estilo cuando el ratón pase por encima como hacen los otros.

```
nav li > a:hover {
  background-color: #006800;
  color: #fff;
  padding: 25px 15px;
}
```

Se aplica estilo a los enlaces que son hijos de un elemento lista y éste a su vez el descendiente del elemento `nav` cuando el ratón pasa por encima. Se le aplica color de fondo con `background-color`, color a la fuente con `color` y padding para que el color de fondo cubra una zona más amplia al pasar el ratón.

```
main {
  max-width: 940px;
  margin: 50px auto;
  height: 690px;
}
```

Se le aplica al elemento `main` una anchura máxima (`max-width`), un margen para separarlo del menú de navegación y que quede centrado en la página (`margin`) y una altura para que el pie quede al final de la página (`height`).

```
footer {
  background-color: #006800;
  color: #fff;
  height: 70px;
  clear: left;
  /* position: absolute;
  bottom: 0em;
  right: 0em;
  left: 0em;*/
  display: flex;
  flex-direction: row;
```

```
justify-content: space-between;
flex-wrap: nowrap;
}
```

Al pie de página (elemento footer) también le he aplicado propiedades para hacerlo un contenedor flex (display: flex). Con flex-direction: row se despliegan los ítems a lo largo de la fila, justify-content con el valor space-between para que, en el espacio que tienen, los ítems se alineen y haya un espacio equitativo entre cada ítem. Con flex-wrap: nowrap se evita que los ítems ocupen otra línea si no caben en una. El color de fondo con background-color, el color de la fuente con color, le he fijado la altura con la propiedad height, con clear: left se está limpiando la propiedad float: left que había aplicado en la página de inicio y así evitar que el pie se viera afectado por ella.

Había usado la propiedad position: absolute con los valores bottom, right y left a 0 para que el pie quedara al final de la página y ocupara todo el ancho. Es una propiedad que no me convenció mucho cuando la utilicé porque al hacer la página más pequeña el pie se subía hacia arriba y tapaba una parte de la página que se debería ver, pero estuve probando varias cosas y es lo único que me funcionó para que el pie quedara al final de la página. Finalmente la comenté para que no se aplicara este estilo y le definí una altura fija al cuerpo de la página.

```
footer > p {
  margin: auto;
}
```

margin: auto para que el texto quede centrado. Se le aplica al párrafo que es hijo del footer.

```
.autor {
  font-weight: bold;
  font-style: normal;
  margin-right: 50px;
}
```

Se le dan estilos a la clase autor, negrita (font-weight), se le quita la cursiva porque la clase es un atributo de la etiqueta span a la cual le aplico cursiva al texto en otras partes, y se añade una separación con el texto de la derecha con el margin-right.

```
footer a {
  color: #fff;
  padding-left: 0;
}
```

Al enlace descendiente del footer se le aplica un color diferente al que se genera con la etiqueta a de los enlaces y se le reduce el padding de la izquierda para que quede más pegado al texto que tiene a la izquierda.

```
/*INICIO*/
figure {
  float: left;
  max-width: 30%;
  margin: 0 15px 5px -10px;
}
```

Al elemento figure se le aplica la propiedad float: left para que se coloque en la parte izquierda de la pantalla (tal como se pedía en la práctica). Como al elemento que viene a continuación (que es article) también se le aplica la propiedad float, se le pone un ancho máximo a la

imagen para que quepan los dos elementos en el ancho que hay disponible (max-width: 30% - la imagen ocupará como máximo el 30% del ancho que hay disponible). Con margin se coloca la imagen más pegada al borde del ancho permitido.

```
h1, h2 {
  color: #006800;
}
```

Le aplico un color a los encabezamientos de primer y segundo nivel.

```
h1 {
  font-size: 2em;
}
```

Le aumento el tamaño a la fuente en los encabezamientos del primer nivel.

```
h2 {
  font-size: 1.2em;
  font-weight: lighter;
}
```

Le aumento el tamaño a la fuente y hago que sea más “fina” en los encabezamientos del segundo nivel.

```
cite {
  font-style: normal;
  font-weight: bold;
}
```

A los elementos que se encuentran dentro de esta etiqueta le elimino la cursiva que aplica cite con font-style y le aplico negrita con font-weight.

```
article {
  float: left;
  max-width: 62%;
  margin-left: 65px;
  margin-top: -30px;
}
```

Como he comentado antes, a article le he dado la propiedad float: left para que se coloque a la izquierda en la pantalla, como ya hay otro elemento con esa misma etiqueta anteriormente, article se coloca al lado de ese elemento. Con max-width, como ya he dicho, le limito el ancho para que quepan los dos elementos en la pantalla sin que uno pase debajo del otro. Y los margin son para colocar correctamente el elemento.

```
span {
  font-style: italic;
}
```

Se le aplica cursiva al elemento span.

```
/* PRODUCTOS */
table {
  border: 1px solid #000;
  border-collapse: collapse;
  width: 100%;
  text-align: center;
```

```
}
```

Al elemento table se le aplica un borde con la propiedad border (grueso, tipo y color de línea), con border-collapse: collapse se solapan los bordes, width: 100% para que la tabla ocupe todo el ancho disponible y con text-align: center para centrar el texto.

```
th, td {
  border: 1px solid #000;
  padding: 5px 0;
}
```

Se le aplican a los elementos th (celda de encabezado) y td (celda de la tabla) los valores al borde con la propiedad border y padding arriba y abajo.

```
#tit_productos {
  text-align: center;
}
```

Se centra el texto del elemento con este id.

```
caption {
  margin-top: 30px;
  background-color: #006800;
  color: #fff;
  font-weight: bold;
  font-size: 1.1em;
  padding: 15px 0;
}
```

Se le aplica al caption (título descriptivo de la tabla) un margen arriba para separarlo de la parte del texto superior (margin), el color de fondo con background-color, el color a la fuente con color, se aplica negrita (font-weight) y se aumenta un poco el tamaño a la fuente (font-size) y se le da un padding arriba y abajo para espaciarlo de los bordes.

```
th {
  background-color: #00d94a;
}
```

Se le aplica el color de fondo a las celdas de encabezado.

```
tr > td {
  background-color: #00fa57;
  font-weight: bold;
  border-color: #cdcdcd;
}
```

A los elementos td que sean hijos directos de la etiqueta tr se les aplica el color de fondo (background-color), se pone el texto en negrita con font-weight y se les pone el color especificado a los bordes con border-color.

```
td + td {
  background-color: #fff;
  border-color: #cdcdcd;
  font-weight: normal;
}
```

Se aplica el color de fondo especificado, el color del borde y se quita la negrita (background-color, border-color y font-weight respectivamente) a todos los elementos td que sean hermanos adyacentes de elementos td.

```
td:last-child {
    border-right: 0px;
}
```

Se le aplica al último hijo que tenga la etiqueta td que no tenga el borde derecho.

```
tfoot td {
    background-color: #00fa57;
    font-style: italic;
    font-size: 0.8em;
    border: 1px solid #000;
    font-weight: normal;
    padding: 20px 0;
}
```

A los elementos td que sean descendientes de tfoot que tengan el color de fondo especificado (background-color), que la fuente sea en cursiva y algo más pequeña que el resto y sin negrita (font-style, font-size y font-weight en este orden), el color, grosor y tipo de línea del borde con border y un padding para que quede centrado.

```
.fin {
    border-bottom: 0px;
}
```

A los elementos con esta class se les quita el borde inferior.

```
/* COMPRA */
fieldset {
    border: none;
}
```

Le he quitado el borde a los elementos fieldset para que no se vean los elementos dentro de cada fieldset como si fuera una caja.

```
.datos {
    float: left;
    margin: 20px 20px;
}
```

Se le aplica la propiedad float: left al fieldset con este atributo class para que se coloque a la izquierda de la pantalla. Con margin lo coloco más o menos en el lugar adecuado.

```
.direccion {
    float: left;
    margin: 20px 20px;
}
```

Lo mismo que para la class anterior, Se colocará a la derecha de .datos.

```
.datosPago {
    clear: left;
    margin: auto 100px 20px 20px;
}
```

Se le aplica el `clear: left` para borrar el float aplicado al elemento anterior y se coloque debajo de los otros elementos. Los margin son para dejarlo centrado, y al derecho le he aplicado un margen superior para que el borde que le he aplicado al elemento legend queda alineado con el del elemento superior.

```
#formulario ul {
  padding-left: 20px;
}
```

Desplazo un poco a la derecha el formulario completo.

```
#formulario li {
  list-style-type: none;
  padding-bottom: 5px;
}
```

Le quito los picos a la lista y separo un poco los elementos entre sí.

```
.datos ul li label {
  width: 150px;
  float: left;
}
```

Con esto lo que consigo es que los textos que aparecen al lado de las casillas de texto queden alineados a la izquierda y con el mismo ancho para que también los textbox queden alineados.

```
.direccion ul li label {
  width: 120px;
  float: left;
}
```

Lo mismo que con la class anterior. Se aplica a los descendientes label de li que son descendientes de ul y que estos a su vez son descendientes de la class especificada.

```
input[type="text"], input[type="email"], input[type="tel"] {
  width: 200px;
  padding: 2px;
  border-radius: 0.2em;
}
```

Con estos estilos, le aplico a los input que contienen los atributos especificados entre corchetes [] (en este caso son todo textbox), un ancho mayor que el que tienen con width y aumento su tamaño interior con el padding, con border-radius les redondeo un poco las esquinas a las cajas.

```
input[type="date"] {
  width: 200px;
  border-radius: 0.2em;
  padding-right: 2px;
}
```

Igual que los elementos anteriores, pero aquí solo le he aplicado padding al lado derecho.

```
legend {
  font-weight: bold;
  color: #006c00;
  border-bottom: 3px solid #00da00;
}
```

```
width: 100%;
padding-bottom: 5px;
}
```

Aquí le aplico a los títulos (legend) de cada “caja” (fieldset) la negrita, un color de fuente, un borde inferior simulando un subrayado, una anchura para que ocupe el total del espacio disponible y un padding para separar ese “subrayado” del texto. Las propiedades son font-weight, color, border-bottom, width y padding-bottom respectivamente.

```
#dia {
  float: left;
  margin: 0 20px;
}
```

El elemento con este id se coloca a la izquierda (float: left) y se le da un margen en los laterales con margin.

```
#hora {
  float: left;
  margin: 0 20px;
}
```

Hace lo mismo que el elemento anterior.

```
#dia select {
  margin: 0 10px;
  padding: 3px;
  border-radius: 0.2em;
}
```

A los elementos select (caja desplegable) que son descendientes del id, se les da unos márgenes laterales para desplazarlos (margin), se les aplica un padding para ampliar el tamaño interior de la caja y se redondean los bodes de la caja.

```
#hora select {
  margin: 0 10px;
  padding: 3px;
  border-radius: 0.2em;
}
```

Lo mismo que el elemento anterior.

```
.metodo legend, .fecha legend {
  border: none;
  font-style: italic;
}
```

Se le quita el borde y se pone el texto en cursiva a los elementos legend descendientes de cada una de las clases indicadas.

```
.metodo, .fecha {
  margin-top: 16px;
  padding-left: 20px;
}
```

A estas clases se las aumenta el espacio para que queden más separadas.


```
input[type="radio"] {
  margin-left: 20px;
}
```

A los selectores input que tengan este atributo type se les aumenta el margen izquierdo para que se separen algo más los radiobutton entre ellos.

```
.metodo label {
  margin-right: 5px;
}
```

Se aumenta el margen derecho de los elementos label que son descendientes de esta clase.

```
.metodo label > img {
  margin: 0 0 -3px 4px;
}
```

A las imágenes que sean hijos de label y que este elemento sea hijo de esta clase, se le aplican estos márgenes. Lo he hecho para alinear la imagen con el texto.

```
.boton {
  margin: 20px 10px;
  padding: 10px 15px;
  font-size: 14px;
  border-radius: 5px;
  color: #006800;
  border: 2px solid #006800;
  font-weight: bold;
  font-style: italic;
  box-shadow: rgba(0,0,0,0.7) 0px 5px 15px; /* offset-x | offset-
y | blur-radius | color */
}
```

A las clases botón le aplico unos márgenes para que se separen un poco, un padding para aumentar el tamaño de los botones, indico el tamaño y aplico negrita y cursiva a la fuente (con font-size, font-weight y font-style), les redondeo las esquinas con border-radius, les doy un color a la fuente con color, marco el tamaño-tipo-color de línea con border y le aplico una sombra exterior con box-shadow (los valores de esta propiedad, según el orden en el que aparecen son: color – aquí se le da también la transparencia con el último valor -, sombra en el eje x – en este caso 0 -, sombra en el eje y, y el radio del borde para que no sea una sombra completamente recta.

```
.boton:hover {
  color: #fff;
  background-color: #008900;
  box-shadow: inset rgba(0,0,0,0.7) 0px 5px 15px; /* offset-x | offset-
y | blur-radius | color */
}
```

Se le aplica al botón cuando el ratón pasa por encima y se cambia el color de la fuente y del fondo, y se genera una sombra en el interior del botón (valor inset, el resto es igual que el anterior).

Aplicando las sombras a los botones lo que pretendía era simular que el botón sobresale de la pantalla y al pasar el ratón por encima como si se clicara.

```
#check {  
  border-top: 1px #006800 dotted;  
  padding-top: 15px;  
  margin: 0 100px 0 30px;  
}
```

Con esto simplemente pretendía marcar una separación entre el check que aparece en la página y los botones del resto del formulario.

* Al final del fichero de CSS hay una parte de código comentada porque le apliqué también unos estilos al checkbox para que la casilla fuera algo más grande y la verificación al marcarlo se hiciera en color verde, pero como era un campo requerido y lo había modificado, al darle al botón de Enviar si no había marcado el check no pasaba nada y no se indicaba que era necesario marcar ese check. Probé a que siempre estuviera la casilla en rojo, pero no me gustó el resultado, así que decidí dejar esta parte tal cual para que apareciera el error si se intentaba enviar el formulario sin marcar la casilla.

El código está ahí por si lo quieren probar, imagino que el problema que tengo es porque hay que aplicarle algo de javascript para que funcione todo correcto.