기초 웹

22 Winter CNU 기초 스터디

21 남정연
21 박준서

<table> tag

<table></table>

표를 나타내기 위한 태그

<table> tag

1. **Layout tables reduce accessibility for visually impaired users**: Screenreaders, used by blind people, interpret the tags that exist in an HTML page and read out the contents to the user. Because tables are not the right tool for layout, and the markup is more complex than with CSS layout techniques, the screenreaders' output will be confusing to their users.
2. **Tables produce tag soup**: As mentioned above, table layouts generally involve more complex markup structures than proper layout techniques. This can result in the code being harder to write, maintain, and debug.
3. **Tables are not automatically responsive**: When you use proper layout containers (such as `<header>`, `<section>`, `<article>`, or `<div>`), their width defaults to 100% of their parent element. Tables on the other hand are sized according to their content by default, so extra measures are needed to get table layout styling to effectively work across a variety of devices.

<table> tag

```
<table>
  <tr>
    <td> </td>
    <td>Knocky</td>
    <td>Flor</td>
    <td>Ella</td>
    <td>Juan</td>
  </tr>
  <tr>
    <td>Breed</td>
    <td>Jack Russell</td>
    <td>Poodle</td>
    <td>Streetdog</td>
    <td>Cocker Spaniel</td>
  </tr>
  <tr>
    <td>Age</td>
    <td>16</td>
    <td>9</td>
    <td>10</td>
    <td>5</td>
  </tr>
  <tr>
    <td>Owner</td>
```

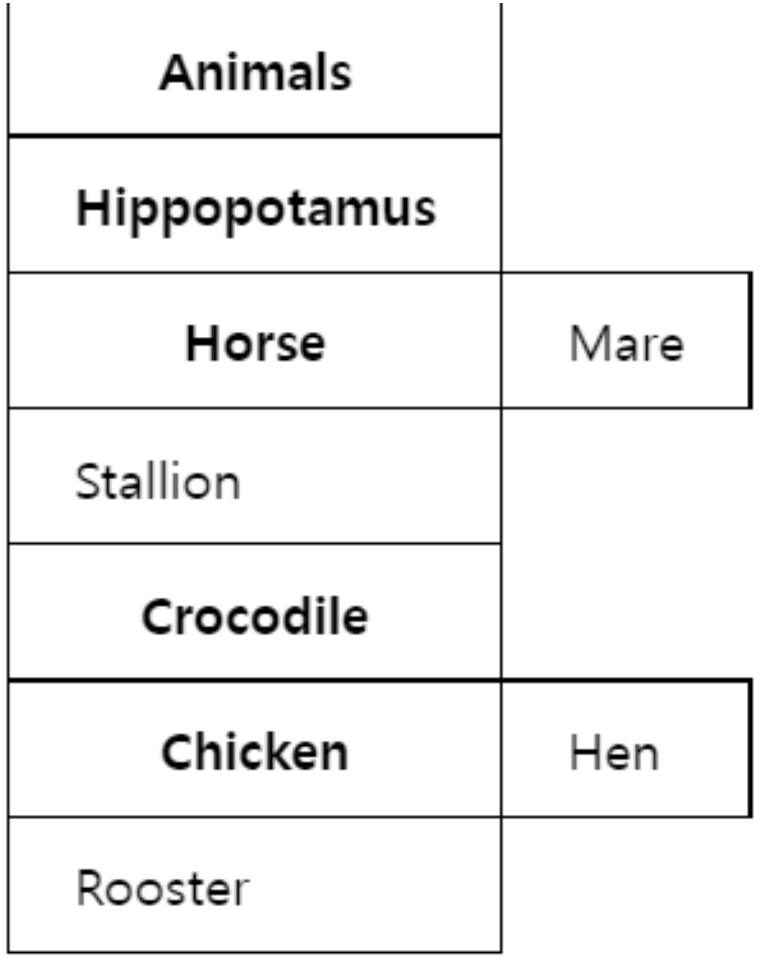
	Knocky	Flor	Ella	Juan
Breed	Jack Russell	Poodle	Streetdog	Cocker Spaniel
Age	16	9	10	5
Owner	Mother-in-law	Me	Me	Sister-in-law
Eating Habits	Eats everyone's leftovers	Nibbles at food	Hearty eater	Will eat till he explodes

Spans

Animals	
Hippopotamus	
Horse	Mare
	Stallion
Crocodile	
Chicken	Hen
	Rooster

Spans

```
<table>
  <tr>
    <th>Animals</th>
  </tr>
  <tr>
    <th>Hippopotamus</th>
  </tr>
  <tr>
    <th>Horse</th>
    <td>Mare</td>
  </tr>
  <tr>
    <td>Stallion</td>
  </tr>
  <tr>
    <th>Crocodile</th>
  </tr>
  <tr>
    <th>Chicken</th>
    <td>Hen</td>
  </tr>
  <tr>
    <td>Rooster</td>
  </tr>
</table>
```



Spans

```
<table>
  <tr>
    <th colspan="2">Animals</th>
  </tr>
  <tr>
    <th colspan="2">Hippopotamus</th>
  </tr>
  <tr>
    <th rowspan="2">Horse</th>
    <td>Mare</td>
  </tr>
  <tr>
    <td>Stallion</td>
  </tr>
  <tr>
    <th colspan="2">Crocodile</th>
  </tr>
  <tr>
    <th rowspan="2">Chicken</th>
    <td>Hen</td>
  </tr>
  <tr>
    <td>Rooster</td>
  </tr>
</table>
```

<table>
<tr><th colspan="2">Animals</th></tr>
<tr><th colspan="2">Hippopotamus</th></tr>
<tr><th rowspan="2">Horse</th><td>Mare</td></tr>
<tr><td>Stallion</td></tr>
<tr><th colspan="2">Crocodile</th></tr>
<tr><th rowspan="2">Chicken</th><td>Hen</td></tr>
<tr><td>Rooster</td></tr>
</table>

Spans

```
<table>
  <tr>
    <th colspan="2">Animals</th>
  </tr>
  <tr>
    <th colspan="2">Hippopotamus</th>
  </tr>
  <tr>
    <th rowspan="2">Horse</th>
    <td>Mare</td>
  </tr>
  <tr>
    <td>Stallion</td>
  </tr>
  <tr>
    <th colspan="2">Crocodile</th>
  </tr>
  <tr>
    <th rowspan="2">Chicken</th>
    <td>Hen</td>
  </tr>
  <tr>
    <td>Rooster</td>
  </tr>
</table>
```

<table>
<tr><th colspan="2">Animals</th></tr>
<tr><th colspan="2">Hippopotamus</th></tr>
<tr><th rowspan="2">Horse</th><td>Mare</td></tr>
<tr><td>Stallion</td></tr>
<tr><th colspan="2">Crocodile</th></tr>
<tr><th rowspan="2">Chicken</th><td>Hen</td></tr>
<tr><td>Rooster</td></tr>
</table>

<colgroup>

```
<table>
  <colgroup>
    <col>
    <col style="background-color: yellow">
  </colgroup>
  <tr>
    <th>Data 1</th>
    <th>Data 2</th>
  </tr>
  <tr>
    <td>Calcutta</td>
    <td>Orange</td>
  </tr>
  <tr>
    <td>Robots</td>
    <td>Jazz</td>
  </tr>
</table>
```

<colgroup>

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
1st period	English			German	Dutch		
2nd period	English	English		German	Dutch		
3rd period		German		German	Dutch		
4th period		English		English	Dutch		

<colgroup>

```
<colgroup>
  <col span="2">
  <col style="background-color:#97DB9A;">
  <col style="width:42px;">
  <col style="background-color:#97DB9A;">
  <col style="background-color:#DCC48E; border:4px solid #C1437A;">
  <col span="2" style="width:42px;">
</colgroup>
```

<colgroup>

```
<colgroup>
  <col style="background-color: yellow" span="2">
</colgroup>
```

Just like `colspan` and `rowspan`, `span` takes a unitless number value that specifies the number of columns you want the styling to apply to.

<caption>

```
<table>
  <caption>Dinosaurs in the Jurassic period</caption>

  ...
</table>
```

<caption>

Florence's weekly lesson timetable

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
1st period	English			German	Dutch		
2nd period	English	English		German	Dutch		
3rd period		German		German	Dutch		
4th period		English		English	Dutch		

<caption>

Florence's weekly lesson timetable

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
1st period	English			German	Dutch		
2nd period	English	English		German	Dutch		
3rd period		German		German	Dutch		
4th period		English		English	Dutch		

<t~>

- The `<thead>` element must wrap the part of the table that is the header — this is usually the first row containing the column headings, but this is not necessarily always the case. If you are using `<col>` / `<colgroup>` element, the table header should come just below those.
- The `<tfoot>` element needs to wrap the part of the table that is the footer — this might be a final row with items in the previous rows summed, for example. You can include the table footer right at the bottom of the table as you'd expect, or just below the table header (the browser will still render it at the bottom of the table).
- The `<tbody>` element needs to wrap the other parts of the table content that aren't in the table header or footer. It will appear below the table header or sometimes footer, depending on how you decided to structure it.

⟨t~⟩

```
<thead>
  <tr>
    <th>Purchase</th>
    <th>Location</th>
    <th>Date</th>
    <th>Evaluation</th>
    <th>Cost (€)</th>
  </tr>
</thead>
<tfoot>
  <tr>
    <td colspan="4">SUM</td>
    <td>118</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <td>Haircut</td>
    <td>Hairdresser</td>
    <td>12/09</td>
    <td>Great idea</td>
    <td>30</td>
  </tr>
```

How I chose to spend my money

Purchase	Location	Date	Evaluation	Cost (€)
Haircut	*Hairdresser*	*12/09*	*Great idea*	*30*
Lasagna	*Restaurant*	*12/09*	*Regrets*	*18*
Shoes	*Shoeshop*	*13/09*	*Big regrets*	*65*
Toothpaste	*Supermarket*	*13/09*	*Good*	*5*
SUM				**118**

Nesting Tables

```
<table id="table1">
  <tr>
    <th>title1</th>
    <th>title2</th>
    <th>title3</th>
  </tr>
  <tr>
    <td id="nested">
      <table id="table2">
        <tr>
          <td>cell1</td>
          <td>cell2</td>
          <td>cell3</td>
        </tr>
      </table>
    </td>
    <td>cell2</td>
    <td>cell3</td>
  </tr>
  <tr>
    <td>cell4</td>
    <td>cell5</td>
    <td>cell6</td>
  </tr>
</table>
```

<table>
<tr><th colspan="2">title1</th><th>title2</th><th>title3</th></tr>
<tr><td colspan="2"><table><tr><td>cell1</td><td>cell2</td><td>cell3</td></tr></table></td><td>cell2</td><td>cell3</td></tr>
<tr><td colspan="2">cell4</td><td>cell5</td><td>cell6</td></tr>
</table>

Accessibility

Tables for visually impaired users

Let's recap briefly on how we use data tables. A table can be a handy tool, for giving us quick access to data and allowing us to look up different values. For example, it takes only a short glance at the table below to find out how many rings were sold in Gent during August 2016. To understand its information we make visual associations between the data in this table and its column and/or row headers.

Items Sold August 2016

		Clothes			Accessories	
		Trousers	Skirts	Dresses	Bracelets	Rings
Belgium	Antwerp	56	22	43	72	23
	Gent	46	18	50	61	15
	Brussels	51	27	38	69	28
The Netherlands	Amsterdam	89	34	69	85	38
	Utrecht	80	12	43	36	19

Scope attribute

```
<thead>
  <tr>
    <th scope="col">Purchase</th>
    <th scope="col">Location</th>
    <th scope="col">Date</th>
    <th scope="col">Evaluation</th>
    <th scope="col">Cost (€)</th>
  </tr>
</thead>
```

```
<tr>
  <th scope="row">Haircut</th>
  <td>Hairdresser</td>
  <td>12/09</td>
  <td>Great idea</td>
  <td>30</td>
</tr>
```

Id and headers

The id and headers attributes

An alternative to using the `scope` attribute is to use `id` and `headers` attributes to create associations between headers and cells. The way they are used is as follows:

1. You add a unique `id` to each `<th>` element.
2. You add a `headers` attribute to each `<td>` element. Each `headers` attribute has to contain a list of the `id`s of all the `<th>` elements that act as a header for that cell, separated by spaces.

This gives your HTML table an explicit definition of the position of each cell in the table, defined by the header(s) for each column and row it is part of, kind of like a spreadsheet. For it to work well, the table really needs both column and row headers.

Returning to our spending costs example, the previous two snippets could be rewritten like this:

Id and headers

```
<thead>
  <tr>
    <th id="purchase">Purchase</th>
    <th id="location">Location</th>
    <th id="date">Date</th>
    <th id="evaluation">Evaluation</th>
    <th id="cost">Cost (€)</th>
  </tr>
</thead>
<tbody>
<tr>
  <th id="haircut">Haircut</th>
  <td headers="location haircut">Hairdresser</td>
  <td headers="date haircut">12/09</td>
  <td headers="evaluation haircut">Great idea</td>
  <td headers="cost haircut">30</td>
</tr>

  ...

</tbody>
```