

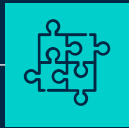
HTML

Tables/Forms/Validations

Unit 1: The Web and HTML

1

Lesson Objectives



01

HLT's and
Employability



02

HTML
Tables



03

HTML Forms

1:1 Progress Review

2

Tables

3



There are many websites on the Internet that display information (bookings, classes, prices etc.)

This data is naturally tabular in nature; a table is often the best way of presenting the data.



4



Creating Tables

Before displaying any information, we must create a table that will hold the data using the `<table>` element.

In HTML, all components must be created.

We need to add rows using the table row element: `<tr>`

```
<table>
</table>
```

```
<table>
  <tr>
  </tr>
  <tr>
  </tr>
</table>
```

5



Creating Tables

Each cell element inside table must also be defined. We can add data using the element: table data.

To add titles to rows and columns, you use the element: table heading `<th>`.

```
<table>
  <tr>
    <td>13</td>
    <td>37</td>
  </tr>
</table>
```

6



Creating Tables

What happened in the code on the right?

First, a new row was added. The blank heading creates the extra table cell necessary to align the table headings correctly over the data they correspond to.

In the second row, one table heading was added as a row title: Name.

```
<tr>
  <th></th>
  <th scope="col">Dolphins</th>
  <th scope="col">Time</th>
  <th scope="col">Capacity</th>
  <th scope="col">Session Number</th>
</tr>
<tr>
  <th scope="row">Name</th>
  <td>Milo</td>
  <td>10:30am</td>
  <td>58 people</td>
  <td>42</td>
</tr>
```

7



Creating Tables

The scope attribute:

row - this value makes it clear that the heading is for a row.

col - this value makes it clear that the heading is for a column.

HTML code for tables may look a little strange at first, but analysing it piece by piece helps make the code more understandable.

```
<tr>
  <th></th>
  <th scope="col">Dolphins</th>
  <th scope="col">Time</th>
  <th scope="col">Capacity</th>
  <th scope="col">Session Number</th>
</tr>
<tr>
  <th scope="row">Name</th>
  <td>Milo</td>
  <td>10:30am</td>
  <td>58 people</td>
  <td>42</td>
</tr>
```

8



Creating Tables

We use CSS to add style to HTML documents, because it helps us to separate the structure of a page from how it looks. (more on this next week!)

```
table {
  height: 40%;
  left: 10%;
  margin: 20px auto;
  border: 1px solid black;
  overflow-y: scroll;
  position: static;
  width: 60%;
}

thead th {
  background-color: #88CCF1;
  color: #FFF;
  font-family: 'Lato', sans-serif;
  font-size: 16px;
  font-weight: 100;
  letter-spacing: 2px;
  text-transform: uppercase;
}
```

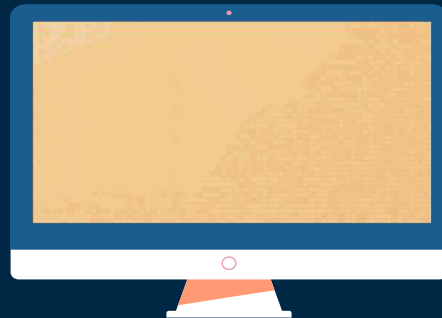
9



Your turn!

Add the following data to your table:

	Dolphins	Time	Capacity	Overall Session Count
Session 1	Milo	10:30am	58 people	42
Session 2	Dolly	11:15am	22 people	12
Session 3	Noah	14:30pm	57 people	182



10



Creating Tables

What if the table contains data that spans multiple columns?
For example, a personal calendar could have events that span across multiple hours, or even multiple days.
Data can span columns using the colspan attribute. The attribute accepts an integer (greater than or equal to 1) to denote the number of columns it spans across.

```
<tr>
  <th scope="row">Session 2</th>
  <td>Dolly</td>
  <td colspan="2">11:15am</td>
  <td>22 people</td>
  <td>12</td>
</tr>
```


11



Forms

12



Forms are a part of everyday life. When we use a physical form in real life, we write down information and give it to someone to process. Think of the times you've had to fill out information for various applications like a job, or a bank account, or dropped off a completed suggestion card — each instance is a form!
























13



We can think of the internet as a network of computers which send and receive information. Computers need an *HTTP request* to know how to communicate. The HTTP request instructs the receiving computer how to handle the incoming information.

14



```
<form action="/example.html" method="POST">
</form>
```

In the above example, we've created the skeleton for a `<form>` that will send information to **example.html** as a POST request:

- The action attribute determines where the information is sent.
- The method attribute is assigned a HTTP verb that is included in the HTTP request.



15



Creating Forms

If we want to create an input field in our `<form>`, we'll need the help of the `<input>` element.

The `<input>` element has a type attribute which determines how it renders on the web page and what kind of data it can accept.

The first value for the type attribute we're going to explore is "text". When we create an `<input>` element with `type="text"`, it renders a text field that users can type into.

```
<p>Get your favourite sweets here!</p>
<form action="/example.html" method="POST">
  <input type="text" name="first-text-field">
</form>
```

16



Creating Forms

After users type into the `<input>` element, the value of the `value` attribute becomes what is typed into the text field. The value of the `value` attribute is paired with the value of the `name` attribute and sent as text when the form is submitted.

This is a sweet shop

Get your favourite sweets here!

This is a sweet shop.

Get your favourite sweets here!

17



Creating Forms

For a user to properly identify an `<input>` we use the appropriately named `<label>` element.

The `<label>` element has an opening and closing tag and displays text that is written between the opening and closing tags. To associate a `<label>` and an `<input>`, the `<input>` needs an `id` attribute. We then assign the `for` attribute of the `<label>` element with the value of the `id` attribute of `<input>`, like so:

Login to place your order!

Please enter your username here:

18



Password Input

An `<input type="password">` element will replace input text with another character like an asterisk (*) or a dot (•). The code below provides an example of how to create a password field:

Login to place your order!

Please enter your username here:

Password:

```
<label for="user-password">Password:</label>
<input type="password" id="user-password" name="user-password">
</form>
```

19



Number Input

By setting `type="number"` for an `<input>` we can restrict what users type into the input field to just numbers (and a few special characters like -, +, and .). We can also provide a `step` attribute which creates arrows inside the input field to increase or decrease by the value of the `step` attribute.

Please enter your name here:

Sweets name:

How many packets would you like?

```
<label for="amount">How many packets would you like?</label>
<input id="amount" type="number" step="1" name="amount">
```

20