

# HTML



# CSS



## HTML & CSS: FUNDAMENTALS OF DEVELOPMENT

Instructor: Beck Johnson  
Week 1



**TABLES**

# WHY TABLES?

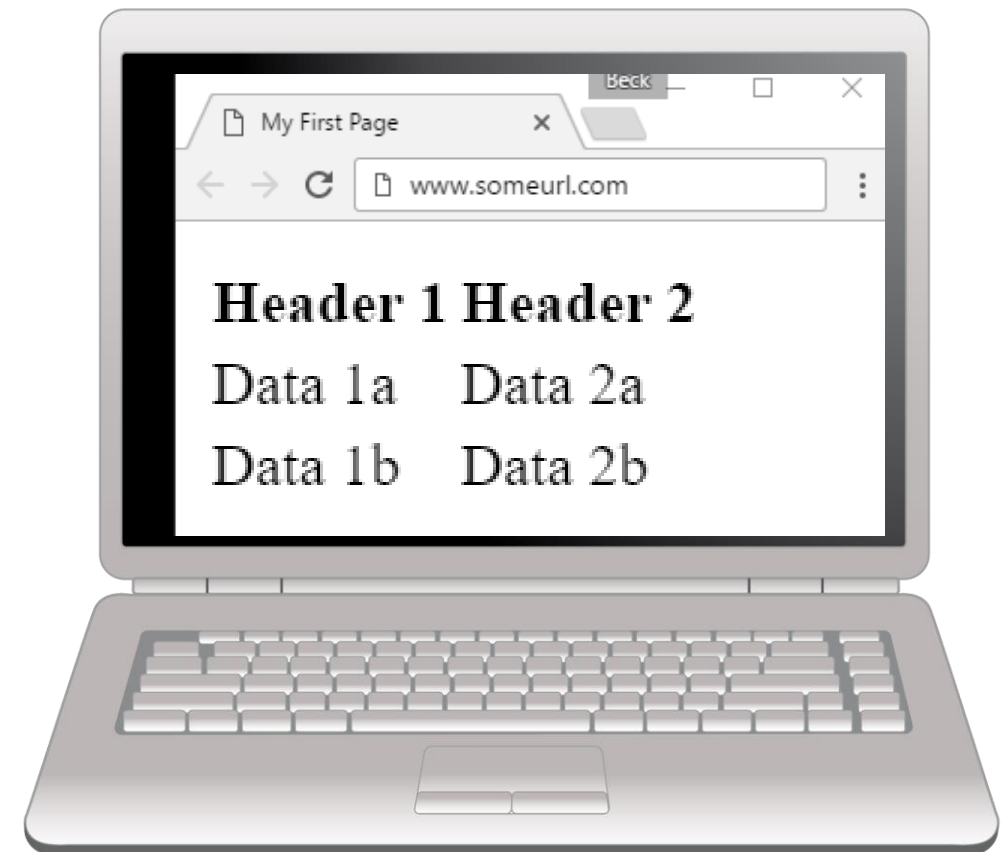
Use tables to present data in a tabular format.

- Listings of people, addresses, etc
- Financial data
- Sports stats
- Product features

Month	Sales
January	\$ 50,000.00
February	\$ 10,000.00
March	\$ 85,000.00
April	\$ 56,000.00
May	\$ 98,000.00

# BASIC TABLE

```
<table>
  <thead>
    <tr>
      <th>Header 1</th>
      <th>Header 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Data 1a</td>
      <td>Data 2a</td>
    </tr>
    <tr>
      <td>Data 1b</td>
      <td>Data 2b</td>
    </tr>
  </tbody>
</table>
```



# TABLE ELEMENTS

`<table>` wraps the whole table

`<thead>` wraps the table header

`<tbody>` wraps the main data

`<tr>` wraps a **t**able **r**ow

`<th>` wraps a **t**able **h**header cell

`<td>` wraps a regular **t**able **d**ata cell

# TABLE ELEMENTS

`<table>` starts a table

- This doesn't generate anything visible in the browser if it's empty – it's just a container

Only `<thead>`, `<tbody>`, or `<tr>` should go directly inside a table

# TABLE ELEMENTS

`<tr></tr>` creates a row in a table

- Every row is a new line
- Header rows and body rows both use `<tr>`
- Make sure the same number of **cells** appear in every row (or things will look broken)

# TABLE ELEMENTS

`<th></th>` creates a **heading** table cell.

- Put inside a row (`<tr></tr>`) inside a `<thead>` element
- Displays as bold by default
- Screen readers will read the heading of a column when reading the corresponding item in a row



# TABLE ELEMENTS

`<td></td>` creates a table cell.

- Put inside a row (`<tr></tr>`) inside the `<tbody>` element
- Table cells can hold any kind of content (images, links, lists, etc)
- Table cells always expand to fit their content, both horizontally and vertically
- Browsers will choose a table cell width that best fits the content (but you can change this in CSS)



**PRACTICE TIME!**

# MAKE A TABLE

Create a new page and add a table with at least four rows and two columns.

What happens if you put different number of cells in one row?



# FORMS

# FORMS

**Forms** are used to collect information from a user.

- Like `<table>`, the `<form>` element doesn't visually appear on the page – it's just a wrapper for form content
- The `<form>` element has two attributes:
  - **action** – what page to load when the form is submitted
  - **method** – what type of HTTP method the form is using (POST or GET)

# FORMS

## Customer Support

<input type="text"/>	<input type="text"/>
<div></div>	
<input type="text"/>	<input type="text"/>

*Fields marked with \* are required*

Submit >

# FORMS

```
<form method="GET" > Customer Support
```

\*First Name

<input type="text" name="fName" />

\*Last Name

<input type="text" name="lName" />

\*How may we help you?

<textarea></textarea>

\*Email

<input type="email" name="email" />

Phone

<input type="tel" name="phone" />

Fields marked with \* are required

Submit >

```
<button type="submit">Submit</button>
```

```
</form>
```

# INPUT

The `input` element allows users to enter data in various ways.

- Looks different depending on its `type`
- Displays inline by default

`<input type="text" />`

`<input type="radio" />`

☐

`<input type="checkbox" />`

☐



# INPUT TYPES

Some **input** types change the keyboard that displays when you fill out that part of the form

This only works for some browsers and some devices

- Email
- Telephone
- Number
- Date
- Color
- Range

date

Markup

```
<input type="date">
```

color

Markup

```
<input type="color" value="#c04019">
```

range

Markup

```
<input type="range" min="1" max="10">
```

# EMAIL INPUT

Some **input** types look like a normal textbox, but change what keyboard displays when you fill out that part of the form

```
<input type="email"  
      name="emailAddress" />
```



# PHONE INPUT

```
<input type="tel" name="phone" />
```



# INPUT NAMES

The `input` element has a **required** attribute called `name`

- Identifies what data is entered for that field
- Not visible to the user in the form itself, but *may* appear in the URL after the form is submitted
- You pick what it's called – so be descriptive!

```
<input type="text" name="firstName" />
```

# INPUT NAMES

When the form is submitted, the browser sends the **name** of the input is associated with the text that the user enters in the textbox when the form is submitted

FIRST NAME

```
<input type="text" name="firstName" />
```

```
1 | GET /?firstName=Beck HTTP/1.1
```

# INPUT

A **placeholder** attribute in a text input creates a “watermarked” label that is inside the box.

- Often used to show formatting suggestions
- By default, the browser displays as a light grey
- But, like anything else, it can be styled with CSS using the **:placeholder** pseudo-class

```
<input type="text" name="phoneNumber"  
      placeholder="(999) 999-9999" />
```



(999) 999-9999

# TEXTAREA

A **textarea** element creates a multi-line text box

TELL US HOW YOU FEEL



```
<textarea rows="3" cols="20"></textarea>
```

- The size of a **textarea** can be specified by **rows** and **cols**, or (even better) set the size using CSS

# LABELS

A label describes what should go inside the input on a form.

```
<label for="address">Address:</label>  
<input type="text" name="address" />
```

You tell the label what it's **for** by giving it the **name** of the input field it's related to.

- By default a **label** appears on the same line as an **input**

Address:



# DROPDOWNS

A `select` element creates a dropdown box.

A rectangular dropdown menu with a thin border. Inside, the text 'Cromwell' is displayed in a standard sans-serif font. To the right of the text is a small, dark downward-pointing triangle (caret) indicating that the menu can be expanded.

```
<select name="cities">
  <option value="sydney">Sydney</option>
  <option value="melbourne">Melbourne</option>
  <option value="cromwell" selected="selected">Cromwell</option>
  <option value="queenstown">Queenstown</option>
</select>
```

# DROPDOWNS

```
<select name="cities">
  <option value="sydney">Sydney</option>
  <option value="melbourne">Melbourne</option>
  <option value="cromwell" selected="selected">Cromwell</option>
  <option value="queenstown">Queenstown</option>
</select>
```

Each **option** in a dropdown is a child of a **select** element.

- To choose an option when the page loads, apply the attribute **selected="selected"**

# BUTTONS

Finally, a button is how the form is submitted

```
<button type="submit">Click me!</button>
```

A rectangular button with a light gray background and a thin black border. The text "Click me!" is centered on the button in a black, sans-serif font.

- When a button has `type="submit"`, hitting enter will automatically submit the form
- When a button has `type="reset"`, the form is cleared when you click it



**PRACTICE TIME!**

# MAKE A FORM

Create a new page that contains a form.

- Include at least 2 different types of **input**, at least one **label**, and a submit **button**
- See a demo of how this might look:  
<https://kweeket.github.io/dev-101/demos/forms.html>
- What happens when you click the submit button?