What is Interactivity Anyway?

Interactivity is:

A **user action** followed by a feedback, usually in a **change/reaction** in what you see

We will be showing you how to:

1. **change what you see** (manipulate elements)
2. **handle a user action** (event handling)

When does JavaScript run?

Browser loads HTML + CSS, then JS runs

Manipulating Elements

Interactivity Part 1

Selecting elements in JS

Many **ways of selecting elements**. The most basic way is to use:

```
const myTitle = document.selectElementById("#title");

myTitle.innerHTML = "A New Title";
```

document.querySelector()

HTML:

```html
<header>
 <a href="http://example.com" id="title" class="large">
Welcome to my world</a>
</header>
```

JS:

```javascript
const element = document.querySelector('header .large');
```

document.getElementById()

HTML:

```html
<a href="http://example.com" id="title" class="large">

    Welcome to my world

</a>
```

JS:

```js
const element = document.getElementById('title');
```

HTML contents

Get the text

```
const element = document.querySelector('#title')

element.innerHTML;
```

Change the text

```
element.innerHTML = "New title";
```

Styles

Changing styles [examples]

```
element.style.padding = "30px";

element.style.backgroundColor = "#222222";
```

Styles

Getting styles

```
//only if it's inline CSS (see inline css section)

element.style.backgroundColor;
```

Attributes

Getting attribute example

//gives you "http://example.com"

element.getAttribute('href');

Set attribute example

element.setAttribute('href', 'http://google.com');

Classes

Get classes [example](#)

```
//gives you "large"

element.getAttribute('class');
```

Classes

Add classes [examples](#)

```
element.classList.add('newClass');
```

Remove classes

```
element.classList.remove('newClass');
```

Children

Adding children [create element](link), [add child](link)

```
//create element

const newSpan = document.createElement('span');

//add it to element

element.appendChild(newSpan);
```
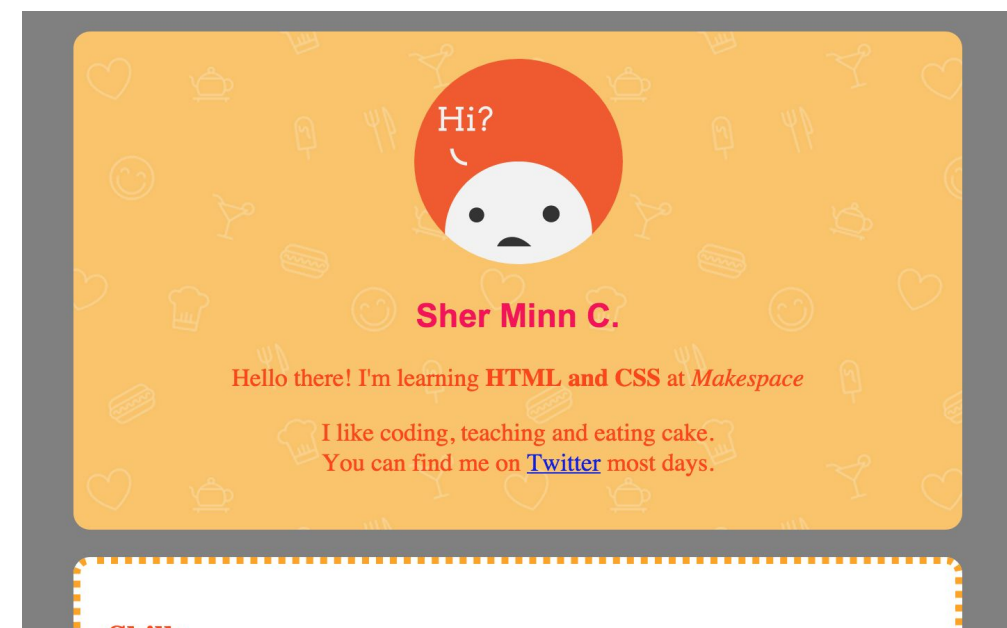
Bonus

Hack this site

Work in pairs for this exercise. Here's a nice site that looks familiar.

Using Javascript only, manipulate the elements to add your personal details. No editing the HTML or CSS.

Start: http://codepen.io/piratefsh/pen/qdmEPG

Handling User Events

Interactivity Part 2

Detecting when a user clicks on a button?

Users can cause events.

Events can be a `click`, `mouseover`, `keydown`, etc.

Events can also be things that are happening in a browser (e.g. `load` is an browser event that fires when the page is finished loading).

List of [important events](#).

Terminology

Event:

A thing that happens, either triggered by a user or browser.

Fire:

Term to say that an event happened.

e.g. a click event was fired

Terminology

Trigger:

Cause an event to be fired

e.g. clicking on the button causes the click event to be fired

Event Handling:

An event handler is a function that is used to do something when an event is fired.

e.g. I wrote an event handler to display a kitten when a button is clicked
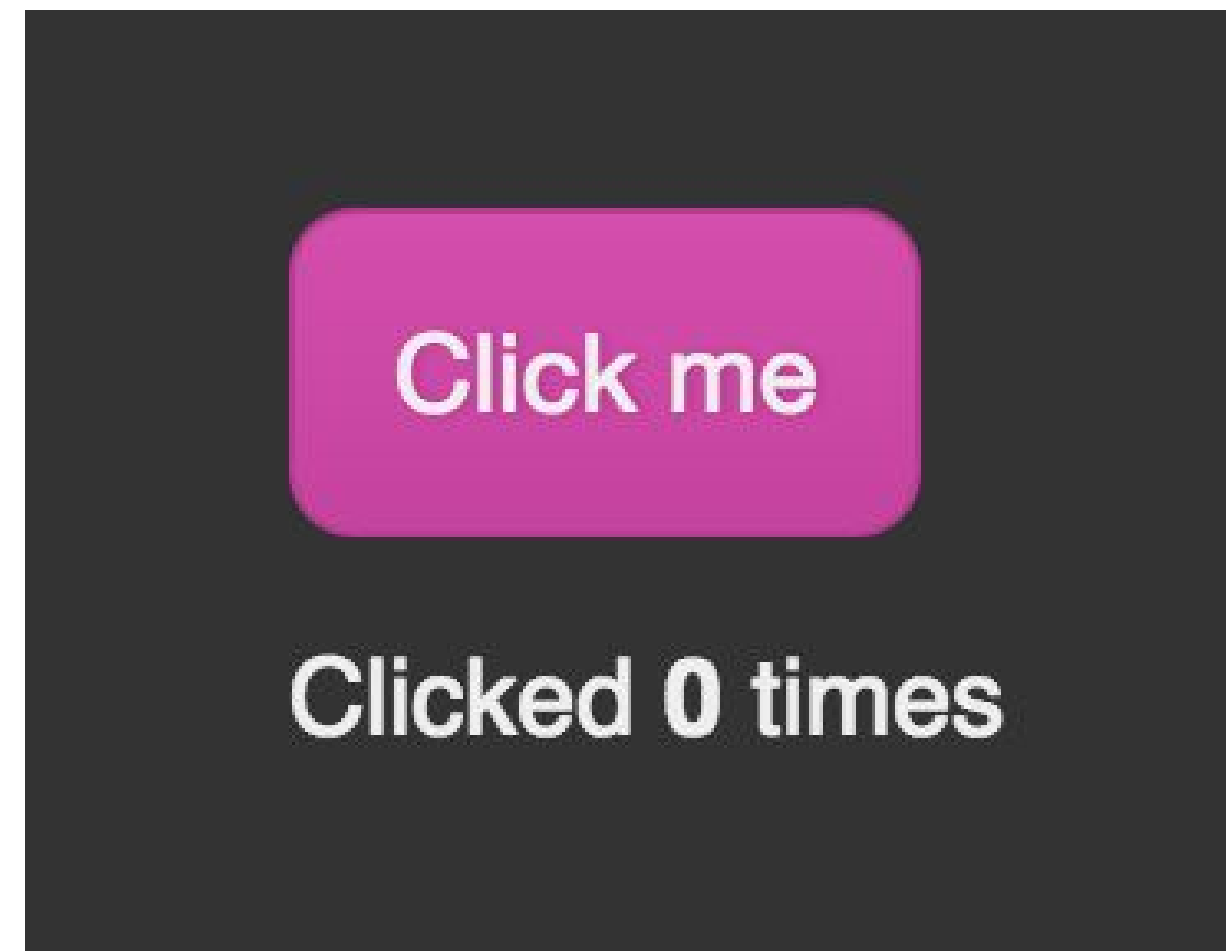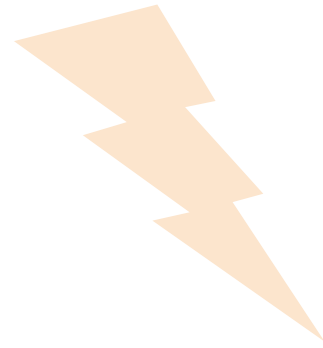
Anatomy of an Event Handler

on this event

do stuff in function

for this element

```
button.onclick = function (event){
    window.alert('I GOT CLICKED ON');
}
```

* Assume `const button = document.getElementById('my-button');` is set before this.

Example: Counting Clicks

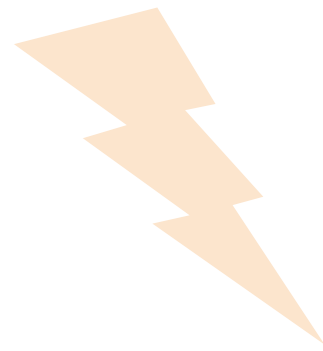## Display the number of clicks on the button.

Bonus: The Elusive Button

Make a button that just refuses to be clicked.

When a user hovers their pointer over it, it runs away!

After you hover over it 10 times, it will finally let you click it.

Hint: Use CSS position to move your button. Use Math.random() to randomize your button position.

Bonus: A To-do List

Create a to-do list. User will enter a to-do item in an input box. When they click on the 'Add' button, the item will be added to the list of existing items.

Bonus 1 : If the item is an empty string, don't add it

Bonus 2 : Clear the input box after item is added

Recap

We have covered

- **Manipulating HTML elements**
  - add, remove, update, finding, etc
- **Handling events**

Interactivity is about combining both! On event, change something!