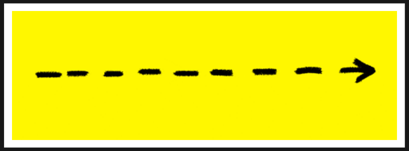
EVENT LISTENERS

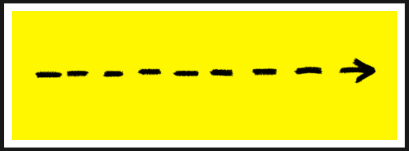
EVENT LISTENERS

(part of the DOM)

So far our interactions with the DOM have been one way

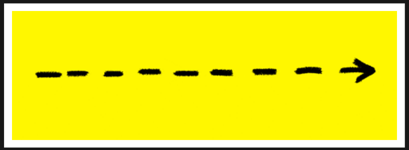


So far our interactions with the DOM have been one way



We've been telling the DOM what to do

So far our interactions with the DOM have been one way



We've been telling the DOM what to do

But until now we didn't know how to listen to the DOM

Introducing *Event Listeners!*

...or "how to listen to the browser"

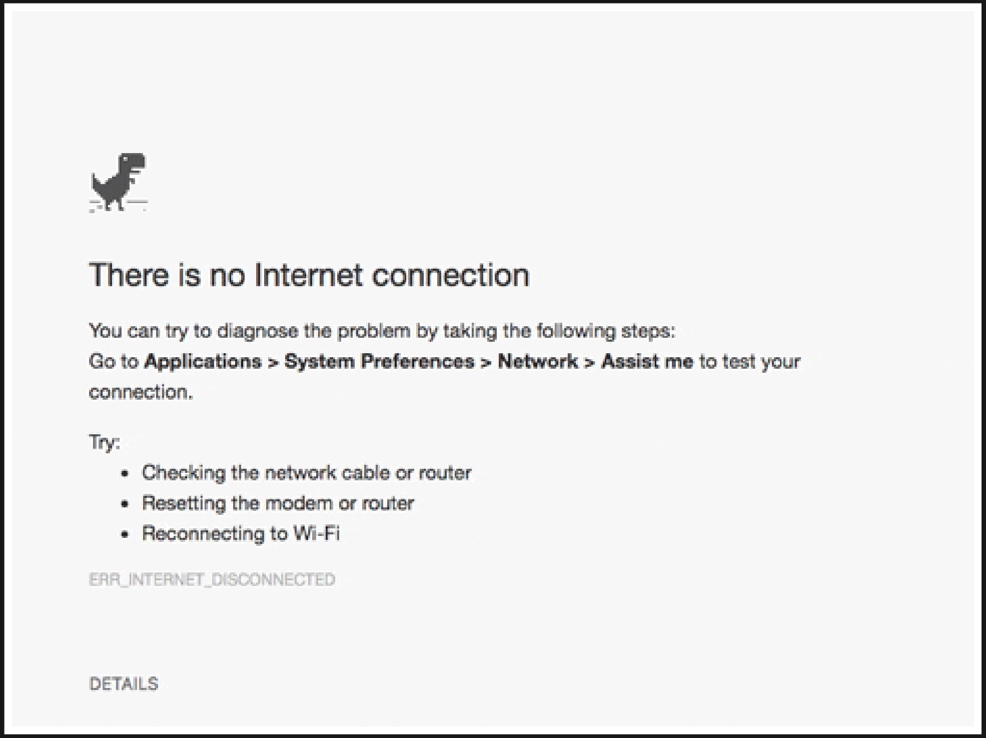
With event listeners, we can "listen" for certain actions (or events) in the browser

With event listeners, we can "listen" for certain actions (or events) in the browser

A few examples of where we would do this:

- Listen to the **mouse**
- Listen to the **keyboard**
- Listen for **media playback** (video or audio, for example is it being played or paused)
- Listen for **internet connectivity**

Have you all seen this before?



There is no Internet connection

You can try to diagnose the problem by taking the following steps:
Go to **Applications > System Preferences > Network > Assist me** to test your connection.

Try:

- Checking the network cable or router
- Resetting the modem or router
- Reconnecting to Wi-Fi

ERR_INTERNET_DISCONNECTED

An action could be a mouse click event

An action could be a mouse click event

Here is a short list of events we can listen to

An action could be a mouse click event

Here is a short list of events we can listen to

- **Mouse events**
- click
- dblclick
- mouseover
- contextmenu

An action could be a mouse click event

Here is a short list of events we can listen to

- **Mouse events**
- click
- dblclick
- mouseover
- contextmenu

- **Focus events**
- focus
- blur

An action could be a mouse click event

Here is a short list of events we can listen to

- **Mouse events**
- click
- dblclick
- mouseover
- contextmenu

- **Focus events**
- focus
- blur

- **Window events**
- resize

How do we add an event listener?

How do we add an event listener?

We use the `addEventListener` method. Example:

Element.addEventListener(*event type*, *callback*);

How do we add an event listener?

We use the `addEventListener` method. Example:

Element.addEventListener(*event type*, *callback*);

```
document.addEventListener('click', () => {
    // callback code
});
```

How do we add an event listener?

We use the `addEventListener` method. Example:

Element.addEventListener(*event type*, *callback*);

```
document.addEventListener('click', () => {
    // callback code
});
```

```
document.addEventListener('click', handleClick);
```

```javascript
document.addEventListener('click', () => {
    // callback code
});
```

What is happening here?

```javascript
document.addEventListener('click', () => {
    // callback code
});
```

What is happening here?

1. We attach a "listener" to the document object

```
document.addEventListener('click', () => {
    // callback code
});
```

What is happening here?

1. We attach a "listener" to the document object
2. We listen specifically to the mouse "click" event

```
document.addEventListener('click', () => {
    // callback code
});
```

What is happening here?

1. We attach a "listener" to the document object
2. We listen specifically to the mouse "click" event
3. We run the callback when the "click" event fires

When the event fires, it also returns to us an object, which we can use:

```
document.addEventListener('click',
(event) => { console.log(event); });
```

Events can have default behaviours, for example, if we "right click" on the page, we get a little context menu.

Events can have default behaviours, for example, if we "right click" on the page, we get a little context menu.

We can disable these "default behaviours" with the `preventDefault()` method

Events can have default behaviours, for example, if we "right click" on the page, we get a little context menu.

We can disable these "default behaviours" with the `preventDefault()` method

We run this method on the `event` object

Events can have default behaviours, for example, if we "right click" on the page, we get a little context menu.

We can disable these "default behaviours" with the `preventDefault()` method

We run this method on the `event` object

```
document.addEventListener('dblclick',
(event) => { event.preventDefault();
console.log(event); });
```

Adding event listeners is great, but if we add too many we soak up too much memory / resources from the browser. It's a good idea to remove unused listeners with `removeEventListener()`

Further reading:

MDN Event Reference

Custom events (also known as synthetic events)