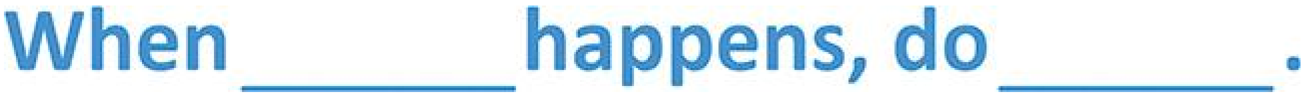
JavaScript

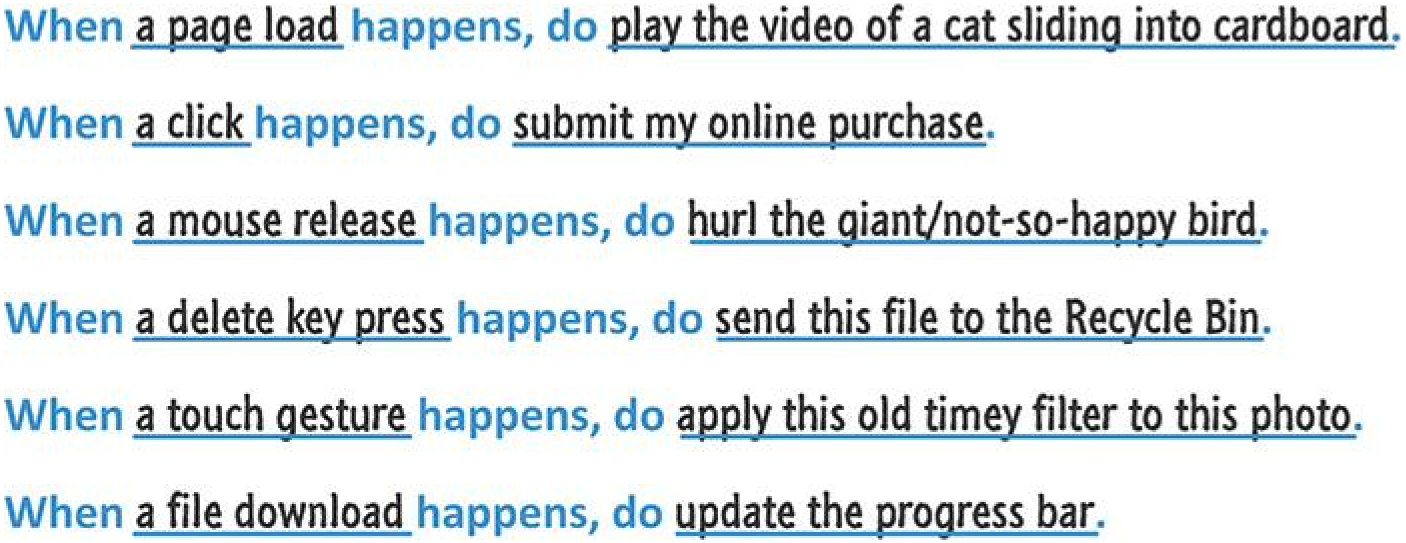
Responding to Events

How events work: listening and reacting



When ________ happens, do ________.

How events work: listening and reacting



When a page load happens, do play the video of a cat sliding into cardboard.

When a click happens, do submit my online purchase.

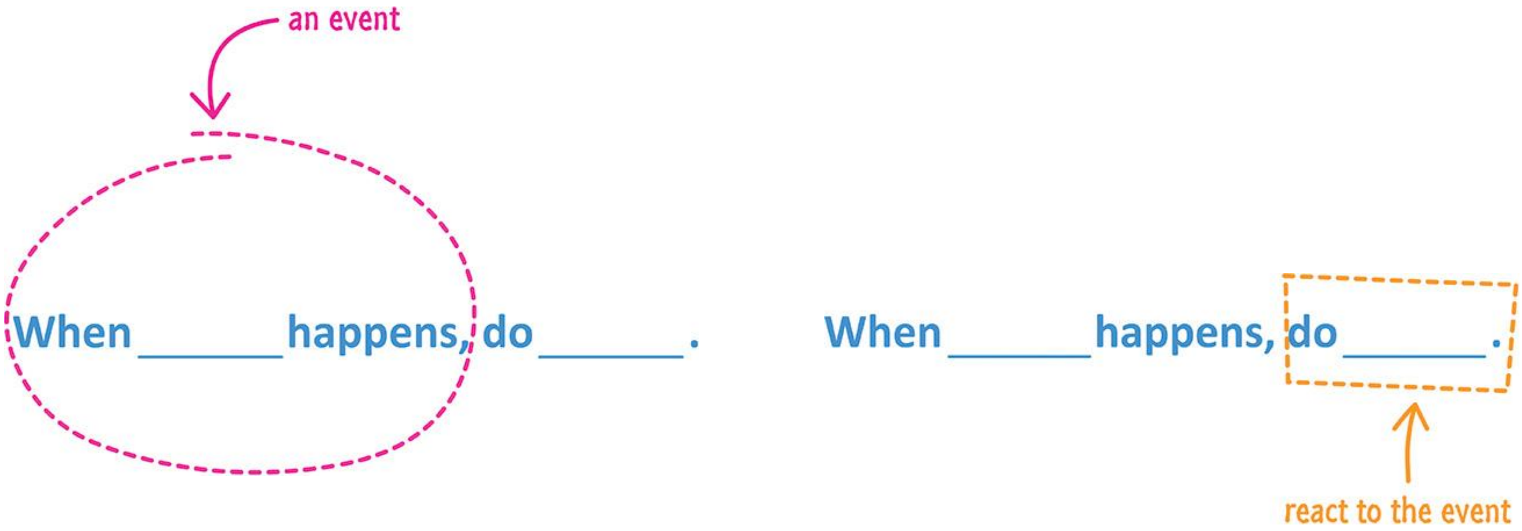
When a mouse release happens, do hurl the giant/not-so-happy bird.

When a delete key press happens, do send this file to the Recycle Bin.

When a touch gesture happens, do apply this old timey filter to this photo.

When a file download happens, do update the progress bar.

How events work: listening and reacting



an event

When ______ happens, do ______.

When ______ happens, do ______.

react to the event

Adding an event listener

```
source.addEventListener(eventName, eventHandler, useCapture);
```

Source: any element that fires events we want to listen to

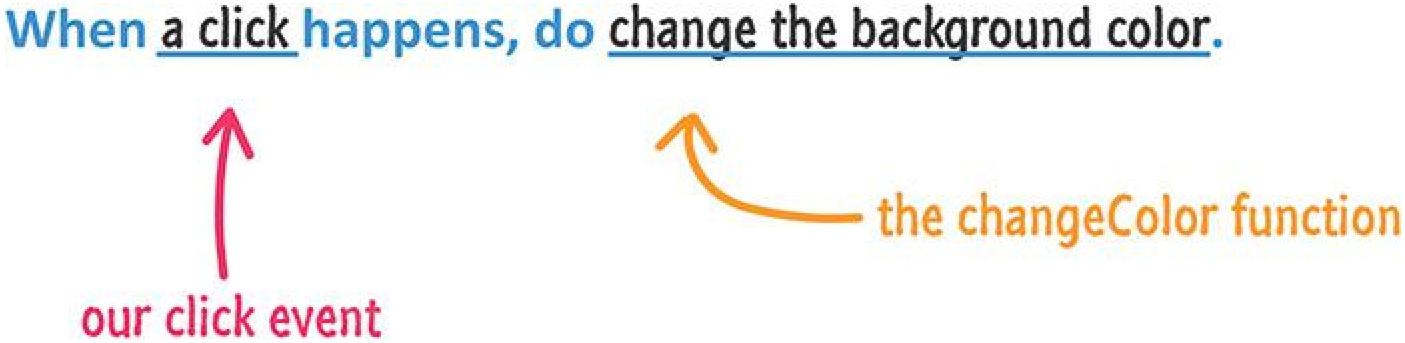
addEventListener: the method we call on the source

Event name: the name of the event we are listening for

Event handler: the name of the function that will get called when the event happens (known as the event handler).
Do NOT use parentheses (that would call the function immediately)

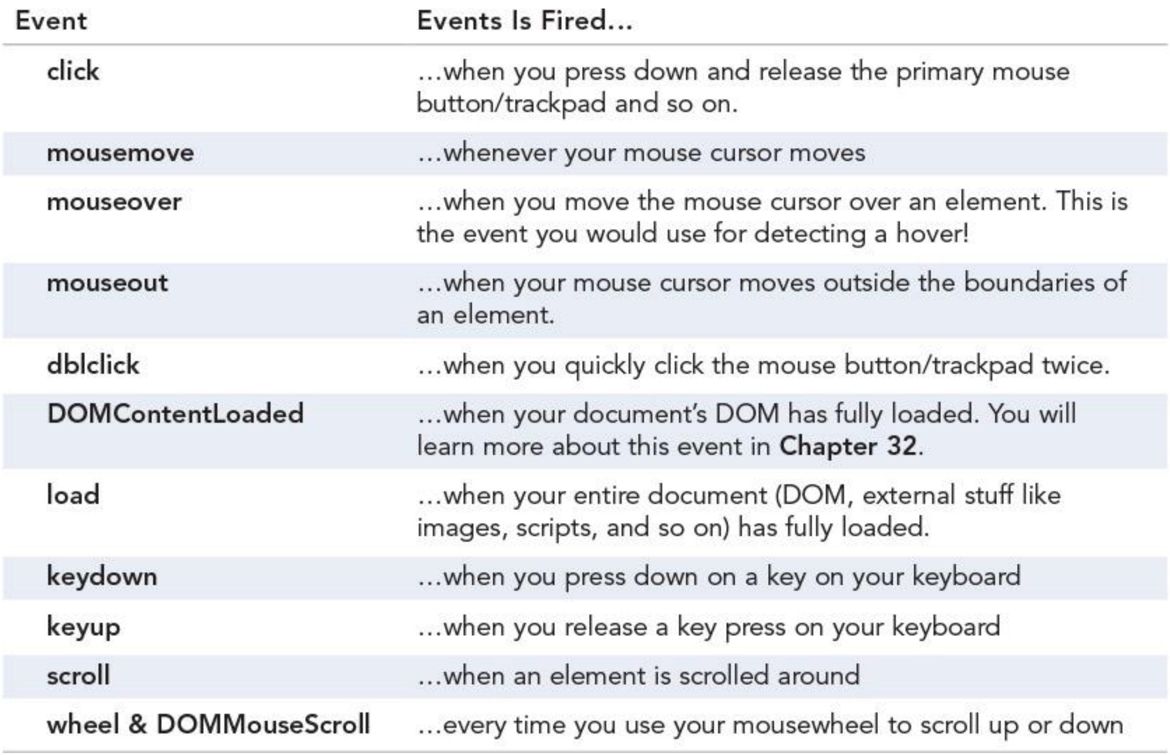
useCapture: just set this to false

How events work: listening and reacting



When a click happens, do change the background color.

Examples of events (there are many more)



Event	Events Is Fired...
click	...when you press down and release the primary mouse button/trackpad and so on.
mousemove	...whenever your mouse cursor moves
mouseover	...when you move the mouse cursor over an element. This is the event you would use for detecting a hover!
mouseout	...when your mouse cursor moves outside the boundaries of an element.
dblclick	...when you quickly click the mouse button/trackpad twice.
DOMContentLoaded	...when your document's DOM has fully loaded. You will learn more about this event in **Chapter 32**.
load	...when your entire document (DOM, external stuff like images, scripts, and so on) has fully loaded.
keydown	...when you press down on a key on your keyboard
keyup	...when you release a key press on your keyboard
scroll	...when an element is scrolled around
wheel & DOMMouseScroll	...every time you use your mousewheel to scroll up or down

Example

html file:

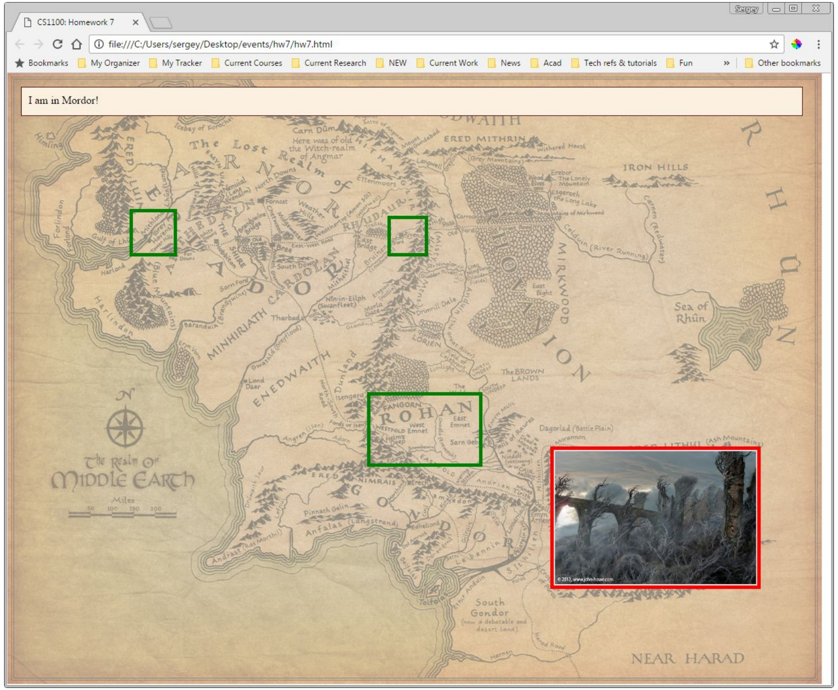
```
<div>click me!</div>
```

js file:

```
function handleAClick() {
    alert("I'm clicked!");
}
var myDiv = document.querySelector("div");
myDiv.addEventListener("click", handleAClick, false);
```

The Travelling Hobbit as Example

- Our task:
  - Display a message when "hobbit" is within a rectangle
  - Highlight the hobbit's location (optional)
- Our old approach:
  - tracking the mouse coordinates and check them after each mouse motion
- A better approach:
  - listen for mouseover events!
  - this works if we do not need to update and display mouse coordinates as the mouse moves


I am in Mordor!

The Travelling Hobbit: Step 1-a (Mordor)

```
function showMordor() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "I am in Mordor!";
}
function hideMordor() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "";
}

//find the #mordor element and add 2 event handlers
document.querySelector("#mordor").addEventListener("mouseover", showMordor, false);
document.querySelector("#mordor").addEventListener("mouseout", hideMordor, false);
```

The Travelling Hobbit: Step 1-b (add Rohan)

```
function showMordor() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "I am in Mordor!";
}
function hideMordor() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "";
}

//find the #mordor element and add 2 event handlers
document.querySelector("#mordor").addEventListener("mouseover", showMordor, false);
document.querySelector("#mordor").addEventListener("mouseout", hideMordor, false);

function showRohan() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "I am in Rohan!";
}
function hideRohan() {
        //find message box and display appropriate text
        document.querySelector("#message").innerText = "";
}

//find the #rohan element and add 2 event handlers
document.querySelector("#rohan").addEventListener("mouseover", showRohan, false);
document.querySelector("#rohan").addEventListener("mouseout", hideRohan, false);
```

The Travelling Hobbit: Step 2

We only need one "hide" function - for the message, regardless of location!

```
function showMordor() {
        document.querySelector("#message").innerText = "I am in Mordor!";
}
function showRohan() {
        document.querySelector("#message").innerText = "I am in Rohan!";
}
function hideMessage() {
        document.querySelector("#message").innerText = "";
}
```

Also, we can find the elements once, and then add any event handling to them:

```
var mordor = document.querySelector("#mordor"):
var rohan = document.querySelector("#rohan");

mordor.addEventListener("mouseover", showMordor, false);
mordor.addEventListener("mouseout", hideMessage, false);

rohan .addEventListener("mouseover", showRohan, false);
rohan .addEventListener("mouseout", hideMessage , false);
```

to be continued...