Events Round 2

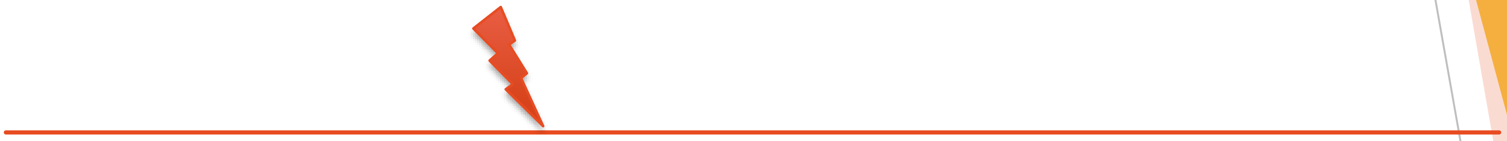
Timers and the Event Object

Review: Events

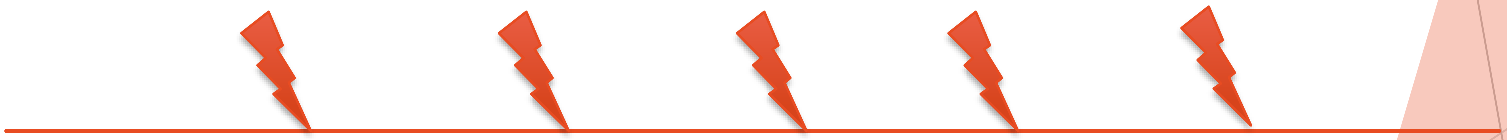
- Reminder:
  - Events are any thing that happens on the webpage, that our program can respond too.
  - We are familiar with the Key and Click Event, but there are others.
  - Event handlers is the code that is run as a result of the event.
  - In our examples so far, we had multiple buttons on the page.
  - Each button can have a different event handler associated.
  - We’ve kept things relatively simple so far, and we shouldn’t feel compelled to make things more complicated until we have to.
  - We will show you how to make things way more complicated now.

Timers

- One time timers
  - Generates an event after a fixed delay



- Periodic timers
  - Generate an event periodically, after a fixed delay
  - Does not stop generating events till you stop it.



- Other stuff can happen in between timer events

Periodic Timers

- We use the command setInterval() to create periodic timers

```
var timerId = setInterval(eventHandler,
interval);
```

Argument	Explanation
eventHandler	A function that will run every time the timer event occurs
interval	The time interval between each timer event, the period of the event. 1/frequency
timerID	The ID to identify the timer event to javaScript. Needed to stop the timer

One Time Event

- we can use the setTimeout to create one time events

```
var timerID = setTimeout(eventHandler, delay);
```

Argument	Explanation
eventHandler	A function that will run after the specified delay
delay	The amount of delay before the eventHandler will run.
timerID	The ID to identify the timer event to javaScript. Needed to stop the timer

Example - Demo

- The following example will create a timer that counts down from 30 to 0 and displays the count down on the HTML page

```
var time = 30;
var timer = setInterval(function(){
myTimer() }, 1000);
function myTimer(){
   time = (time - 1);
   timeDisplay.innerHTML = time;
   if(time === 0)
      clearInterval(timer);
}
```

Review: The Event Object:

- The Event object
  - The event object is how the web browser tells the handler information about the event
  - Different types of events produce event objects that contain different methods and properties

Review: Adding Handlers

- Including it as a tag property.
  - `<input type="button" onclick="func()" >`
  - `<button onclick=“func()”>`
  - Which responds to the button being clicked

- In order to access the event object using this syntax, we just add the event object as a parameter
  - `<input type="button" onclick="func(event)">`
- Then change the way we write the function

```
function func(event){
   // code goes here
}
```

Review: Handlers in JavaScript

- We don't need to change the way we add a Event Handler in JavaScript, because this way doesn’t accept parameters

```
var button = document.getElementById("button4");
button.onclick = myHandler;

function myHandler(){
   alert(event.which);
}
```

- myHandler will use the global event object

Mouse Event objects

- Mouse Event objects contains information about mouse events.
- Mouse Events:
  - onclick: When the user clicks and element
  - onmoousemove: occurs when mouse moves while over an element
  - onmouseenter: When the mouse moves over and element
  - onmouseover: when the pointer moves onto an element or one of it’s children.
  - Onmouseout: When the pointer moves out of an element or one of it’s children.

Mouse Event Object

- You can get the location of the mouse cursor, when a mouse event occurs

```
function mouseEventHandler(event){
   console.log(event.clientX);
   console.log(event.clientY);
}
```

Properties	
event.clientX event.clientY	horizontal/vertical coordinate (within current browser window) of the mouse pointer during mouse event
event.offsetX event.offsetY	retrieves the coordinate of the mouse pointer relative to the top-left corner of the parent element of the element that fires the event

Exercise

- Write the JavaScript to change the background color of the body of the page every ten seconds
- You should change the color to be one of three **random** colors (red, blue, green)

Solution

```
var myVar = setInterval(function(){setColor()}, 10000);
function setColor(){
    var body = document.body;
    var random = Math.random();

    if(random < 0.3)
        body.style.backgroundColor = "red";
    else if(random < 0.6)
        body.style.backgroundColor = "blue";
    else body.style.backgroundColor = "green";
}
function stopColor(){
    clearInterval(myVar);
}
```

Demos!