

JavaScript Events

AND HOW TO MAKE YOUR WEB PAGES NOT SUCK

What the Heck are You Talking About?

- JavaScript Events Man!
 - What they are
 - How to use them
 - Common events
 - Examples! So many examples

What is a JavaScript Event Brosef?

- When you do something, something else happens!
 - With a little JavaScript thrown in.
- First we need to talk about HTML events and then JavaScript events

I Thought we were talking about JavaScript

- We are! But we gotta talk about HTML first.
- An HTML event is when something happens either by the user or the web page itself.
 - Examples:
 - onClick
 - onFocus
 - onChange
 - onMouseOver
 - onload
 - ^ that garbage

That's all well and good but what about JavaScript!?

- Dude, seriously I'm getting to it.
- HTML events are things that happen to HTML elements, JavaScript events are the *reactions* to those changes.
- Basically, HTML events are the triggers to run JavaScript code. That is a JavaScript Event
 - Example
 - `<onClick = "doFunction()"`
 - `OnClick` is the HTML Event and the `"doFunction"` is the reaction to the HTML event.

That's it?

- Yeah pretty much.

There's gotta be more...

- There are different ways to initiate a JavaScript Event.
 - Event Handler Properties
 - Inline Event Handlers
 - `addEventListener()` and `removeEventListener()`

Event Handler Properties? What is that

- Basically you put the event (OnClick, OnLoad) as part of your javaScript like so:

```
1  var btn = document.querySelector('button');  
2  
3  btn.onclick = function() {  
4      var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';  
5      document.body.style.backgroundColor = rndCol;  
6  }
```

- Here we can see the 'btn' variable which is assigned to the HTML document, Then the btn.onclick function.
- This is good practice so it keeps your HTML and JavaScript separate from each other.

What the What? Inline Event Handlers

- When you call a function in your HTML like so:

```
1 | <button onclick="bgChange()">Press me</button>
```

```
1 | function bgChange() {  
2 |     var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';  
3 |     document.body.style.backgroundColor = rndCol;  
4 | }
```

- This is considered bad practice as it mixes your HTML and JavaScript and can become unmanageable.
- Avoid this



AddEventListener() and RemoveEventListener(): Cool Kids on the Block

- This is the newest mechanism for event handling and is similar to event handler properties.

```
1  var btn = document.querySelector('button');
2
3  function bgChange() {
4      var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
5      document.body.style.backgroundColor = rndCol;
6  }
7
8  btn.addEventListener('click', bgChange);
```

- The benefit here is that the button variable is further removed from the function itself and can allow the bgChange function to be used by other events.

Cool! So when should I use these?

- As I mentioned, don't use inline event handlers because that leads to messy code and everyone will hate you.
- Other than that, Event Properties and `addEventListener()` are pretty much interchangeable.
- One benefit to `addEventListener()` is that you can have one HTML event kick off multiple JS Events:

```
1 | myElement.addEventListener('click', functionA);  
2 | myElement.addEventListener('click', functionB);
```

Thanks Zach! My Code is Going to be AWESOME!

- Yeah I know.
- If you wanna learn more about how to make your code awesome with events, check this out
 - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events