

JavaScript is event driven. JS can't effect the files

Inside our browser events are constantly being triggered and ignored

↳ Every instance of the mouse doing anything, that is an event.

Common events → slide 2

onmouseover

onmousedown

We need to set up an event handler

↳ Two ways

→ Store event handler in an event property

→ Set up an event listener on the object that we want to have trigger

→

property events can only store one event

event listeners attach a brand new property to the object.

↳ Can have multiple on one object

↳ All event will be run

List of Window events slide 4? I think → Read the slide

onbeforeprint → runs after the document is printed

onerror → catch the error

onhashchange → changes to another part of url

on

all these events are just for the window

Form events

onblur → user on element loses focus (clicked somewhere else)
onchange → when we change something in the form but will only happen when focus is lost

oninput → similar to onblur but happens right away

onreset

onsearch →

onselect →

onsubmit →

if you type something into a field and then hit enter, it tries to submit the form

onsubmit often used to validate data

drag events

ondrag →

ondragend →

ondragenter →

ondragleave →

ondragover →

ondrop → dropped onto the drop target

Touch events

For touch screens.

Event properties

all objects inherit properties from the object prototype

↳ includes event properties

JS has an object object

all objects "inherit" from this object

don't technically inherit, it goes up the prototype chain
to try to find the property.

onclick →

onmouseover → there are both properties of the object

event properties for ALL event types.

↳ same as any other obj property. Can store anything
we want in the property.

↳ Typically will be a function

↳ can only store one thing so that cannot hold
multiple values (functions)

function init() {

window.onload = init; // function to be run when the event is
triggered.

name of the
object

Event
property
of the
object

Stores the content of init.

Event Listener

Something we can add to an object to listen for an event

- Unlike event properties, can have multiple events added of the same event type.

uses the `addEventListener`.

`Obj.addEventListener(event, function boolean)`

how to add an event listener

```
let panel = document.getElementById("panel");
```

```
panel.addEventListener("click", doSomething, false);
```

↑
not onclick

uses bubbling, we will
learn about it
later

true would mean
catching

Older Browser Compatibility

```
function addEvent(obj, event, fnName){  
    if (obj.addEventListener){ ← Checks that event listener works  
        obj.addEventListener( event, fnName, false);  
    }  
    else {  
        obj.attachEvent("on"+event, fnName);  
    }  
}
```

Turning events off

```
panel.onclick = null;
```

↳ turning off an event property

```
panel.removeEventListener("click", doSomething, false);
```

↳ removing an event listener.

older MS browsers use detachEvent.

Event Object

When event is triggered, the system creates an event obj
tells us

↳ which element triggered specifically
 " " told

↳ which the event was attached to

<Section>

```
<ul>
  <li> Coffee </li>      ← click on
  <li> Tea </li>
  <li>       </li>
</ul>
```

Coffee event triggered
event gets triggered cascading upwards
through the parent elements

event delegation

↳ can do something w/ the
elements within the section

Too much writing on The Event Object slide

Using the event object

```
function handler(e) {
```

```
    alert(e.target.id);           → finds id of element that triggered the event  
    (e.type),                  → what type of event  
    (e.clientX);               → report the horizontal x coord of the event when it was triggered.
```

For ie 9 or earlier.

Look at slick

```
function handler(e) {
```

```
    event = e || window.event;  
  
    target = evt.target || evt.srcElement;  
  
    alert(target.id);  
  
}
```

Event capturing phases

↳ Checks for events trigger in one of two different ways for any given event listener.

Moving down through the tree (capturing)

Moving up (bubbling)

Browser Phase Checking example

Look at diagram

Always starts at the top
Capture happens first

Event Delegation

Set an event on an element, but care about the children of that element

<See S>

<V1>

1. ← could add elements individually, but then the listeners would need to be added every time we add a new li.

<V2>

 → We can add an event to the section

<See S>

e. target would be the specific li that triggered the event.

currentTarget gives us the element that had the eventListener added to it.

Nybs code

for loop for setting mouse over & move out
unity list class

in the loop
e.target.style.borderColor =