

CS 4241 WebWare Events



Web 2.0 Land

Now you can retrieve
data without reload:
with *AJAX*!

Web 2.0 Land

Now you can retrieve
data without reload:
with AJAX!

```
// "callback"
function reqListener () {
    getMyList()
}
// "request"
var oReq = new XMLHttpRequest();
oReq.addEventListener("load", reqListener);
oReq.open("POST", "/putNewClass", true);
oReq.send('newclass='+div.value)
```

```
8
7 <body>
6   <h1> Courses I Teach </h1>
5   <div id="classes"> </div>
4   <form action="putNewClass" method="post" onsubmit="return false;">
3     <!-- TODO -->
2     <label for="newclass"> My New Class Is: </label>
1     <input name="newclass" type="text" id="theNewClassTextBox"/>
0     <input type="submit" onclick="populate();" />
1   </form>
2 </body>
```

Web 2.0 Land

Now you can retrieve data without reload:
with AJAX!

```
// "callback"
function reqListener () {
    getMyList()
}
// "request"
var oReq = new XMLHttpRequest();
oReq.addEventListener("load", reqListener);
oReq.open("POST", "/putNewClass", true);
oReq.send('newclass='+div.value)
```

```
8
7 <body>
6   <h1> Courses I Teach </h1>
5   <div id="classes"> </div>
4   <form action="putNewClass" method="post" onsubmit="return false;">
3     <!-- TODO -->
2     <label for="newclass"> My New Class Is: </label>
1     <input name="newclass" type="text" id="theNewClassTextBox"/>
0     <input type="submit" onclick="populate();" />
1   </form>
2 </body>
```

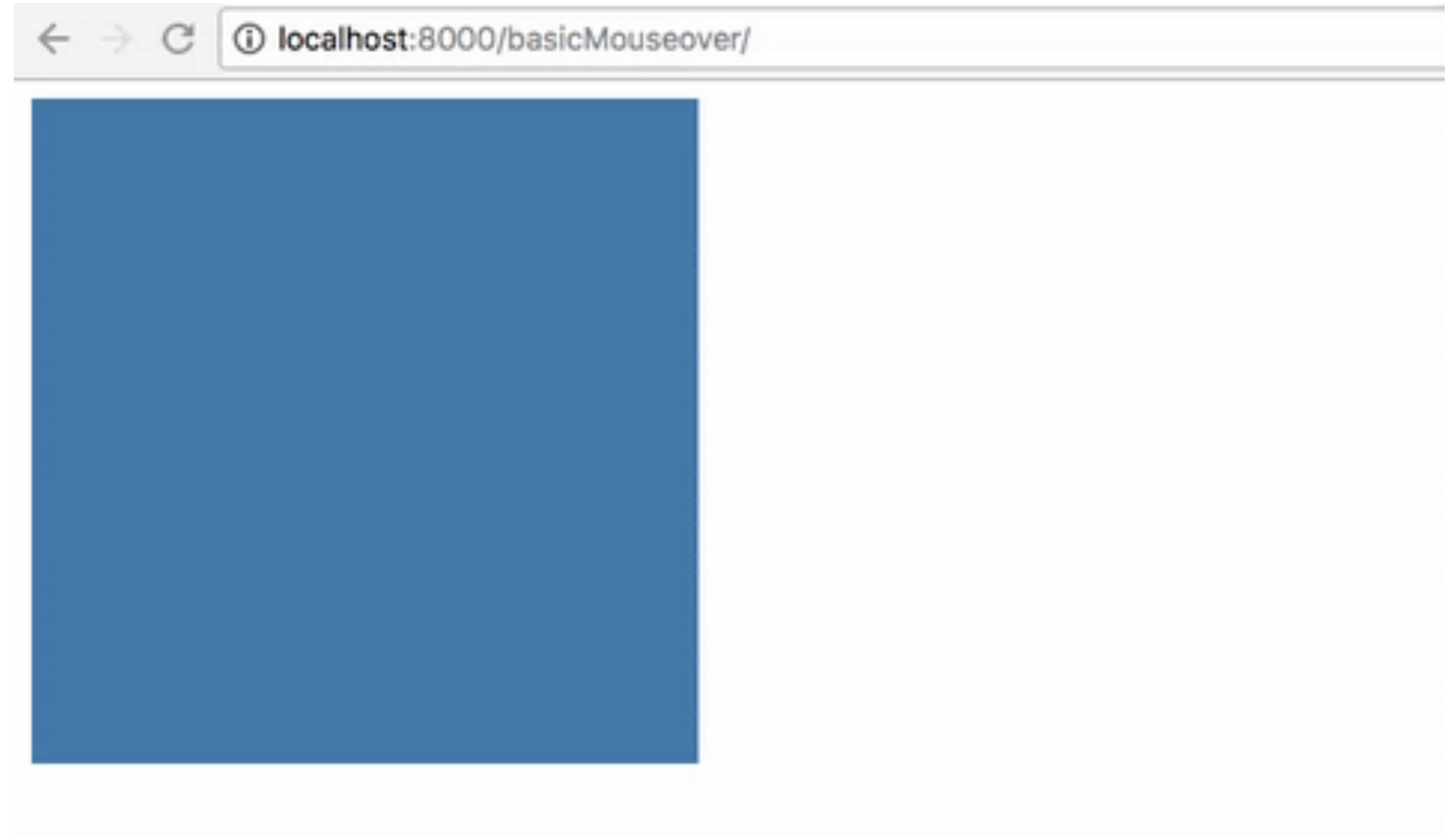
How about
responding to
user input without
forms?

Events

Many triggers:

User interaction:
mouse, keyboard, touch, etc.

Operation completion: animation
end; download complete, etc.

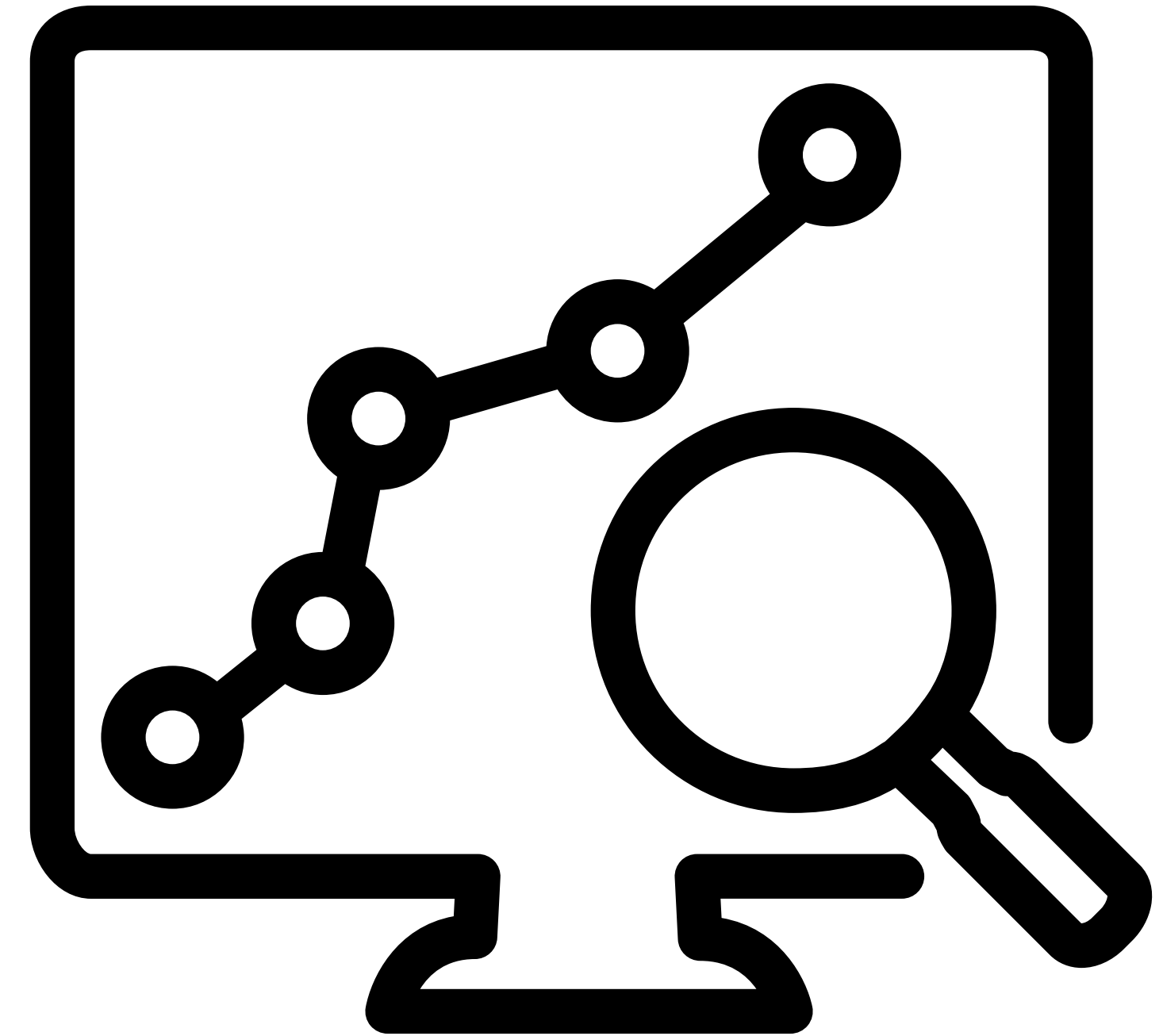


```
0 <div style="width:300px;height:300px;background-color:steelblue"
  onmouseover="console.log(this.style.backgroundColor='gray')">
1 </div>
```

Events Types/Triggers

SO MANY

Read MDN, experiment, get creative.



Database events
Drag & Drop Events
CSS events
Notification events
Media Events

This is your design space of interaction in web!!!

Using Events: addEventListener

Learn. This. Function.

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```


Using Events: addEventListener

Learn. This. Function.

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```

```
<div id='tweet'></div>
```

```
...
```

```
var tweet =  
document.getElementById('tweet')  
tweet.addEventListener(  
  'mouseover',  
  function() { console.log('hi'); },  
)
```


Using Events: addEventListener

Learn. This. Function.

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```

```
<div id='tweet'></div>
```

```
...  
var tweet =  
document.getElementById('tweet')  
tweet.addEventListener(  
  'mouseover',  
  function() { console.log('hi'); },  
)
```

mousing over on div with id “tweet” logs
“hi”

Demotime: basic events

Using Events: addEventListener

Learn. This. Function.

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```

```
<div id='tweet'></div>
```

```
...  
var tweet =  
document.getElementById('tweet')  
tweet.addEventListener(  
  'mouseover',  
  function() { console.log('hi'); },  
)
```

clicking on div with id “tweet” logs “hi”

how would you apply to a CSS class
instead of id?

Inside your Callback

```
<div id='tweet'></div>
```

...

```
var tweet =  
document.getElementById( 'tweet' )  
tweet.addEventListener(  
  'mouseover',  
  function(evt) { console.log(evt); },  
)
```

yields ->



Inside your Callback

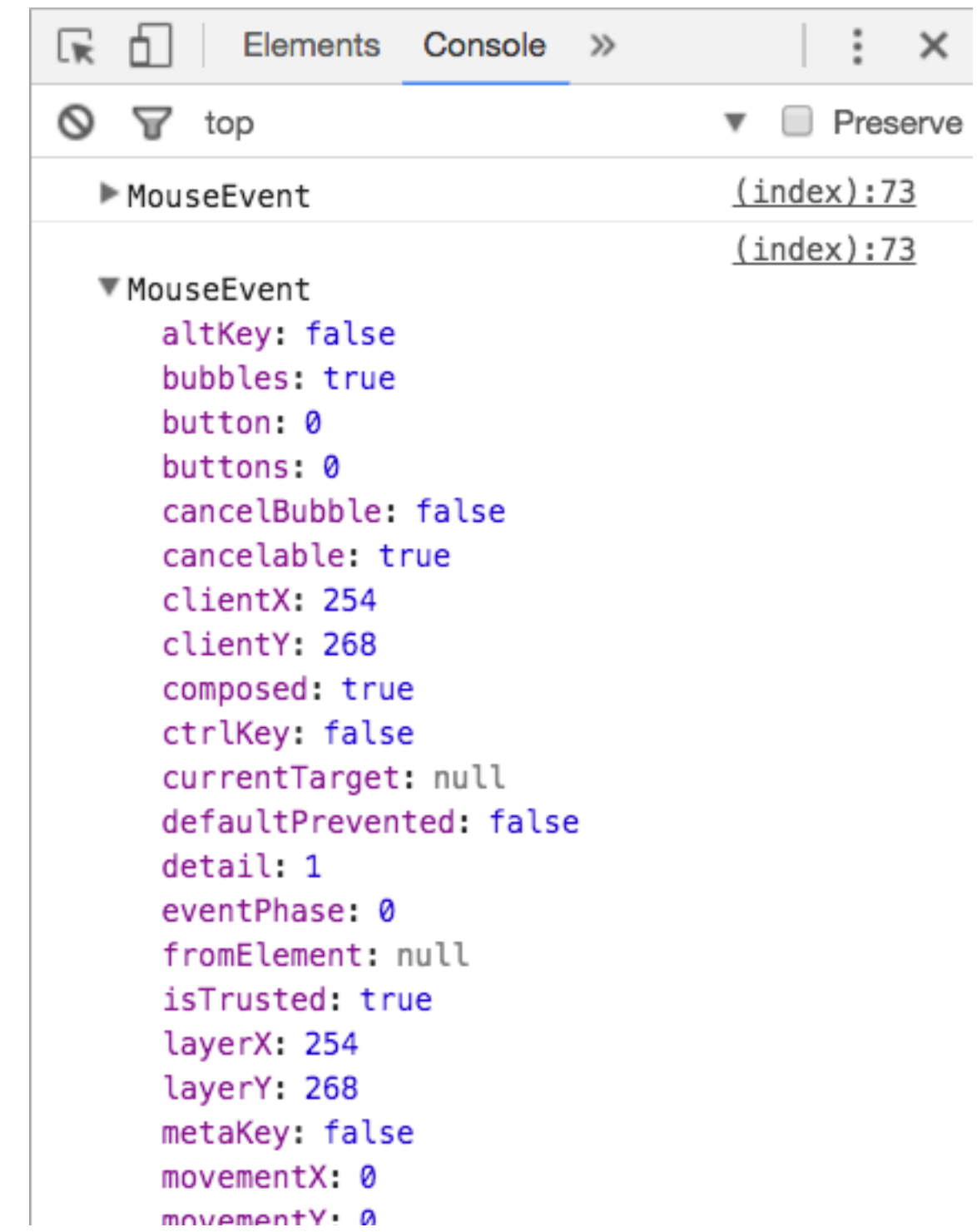
```
<div id='tweet'></div>
```

...

```
var tweet =  
document.getElementById( 'tweet' )  
tweet.addEventListener(  
  'mouseover',  
  function(evt) { console.log(evt); },  
)
```

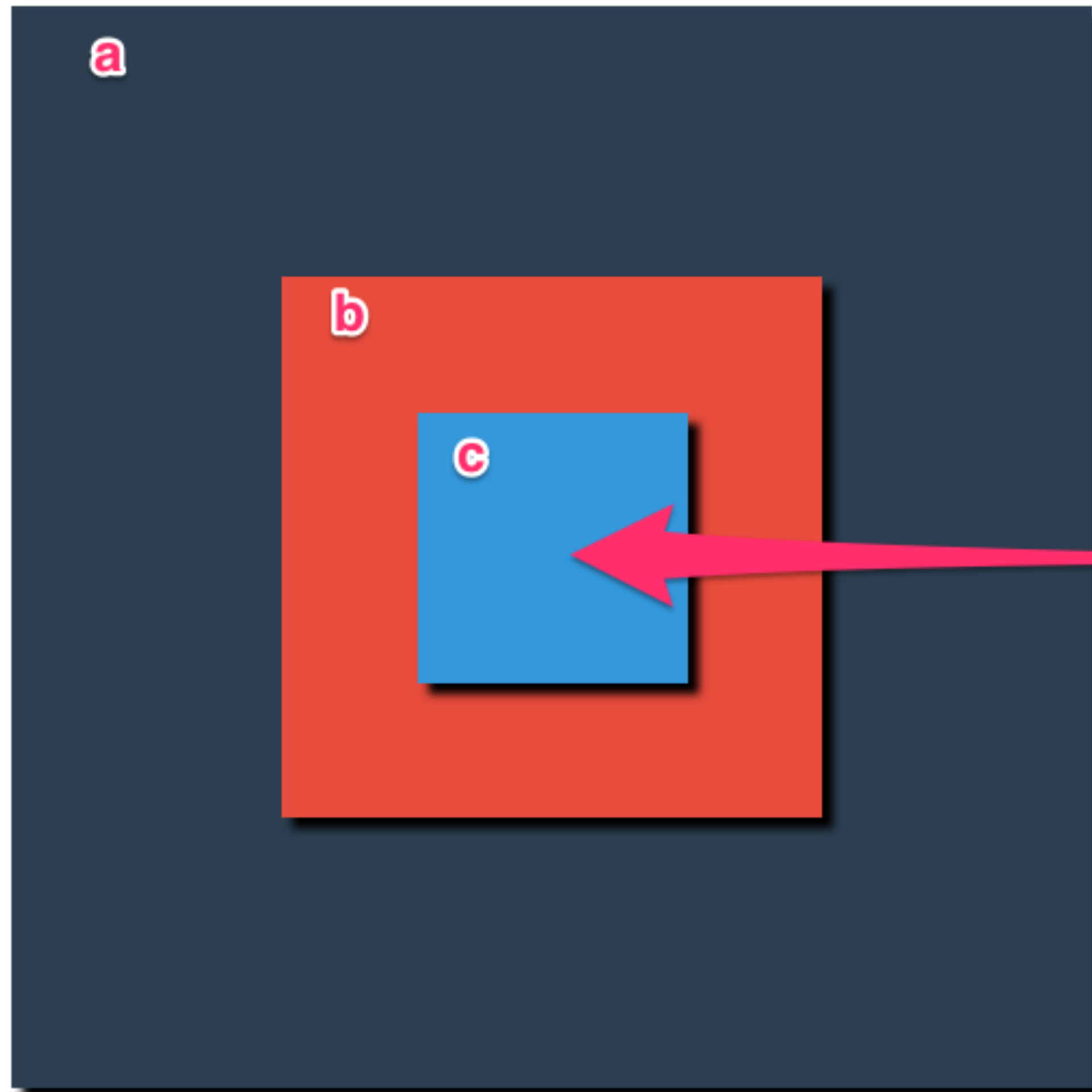
yields ->

```
▼ MouseEvent  
  altKey: false  
  bubbles: true  
  button: 0  
  buttons: 0  
  cancelBubble: false  
  cancelable: true  
  clientX: 254  
  clientY: 268
```



(index):73

Event Propagation



Key concept:
bubble vs. capture

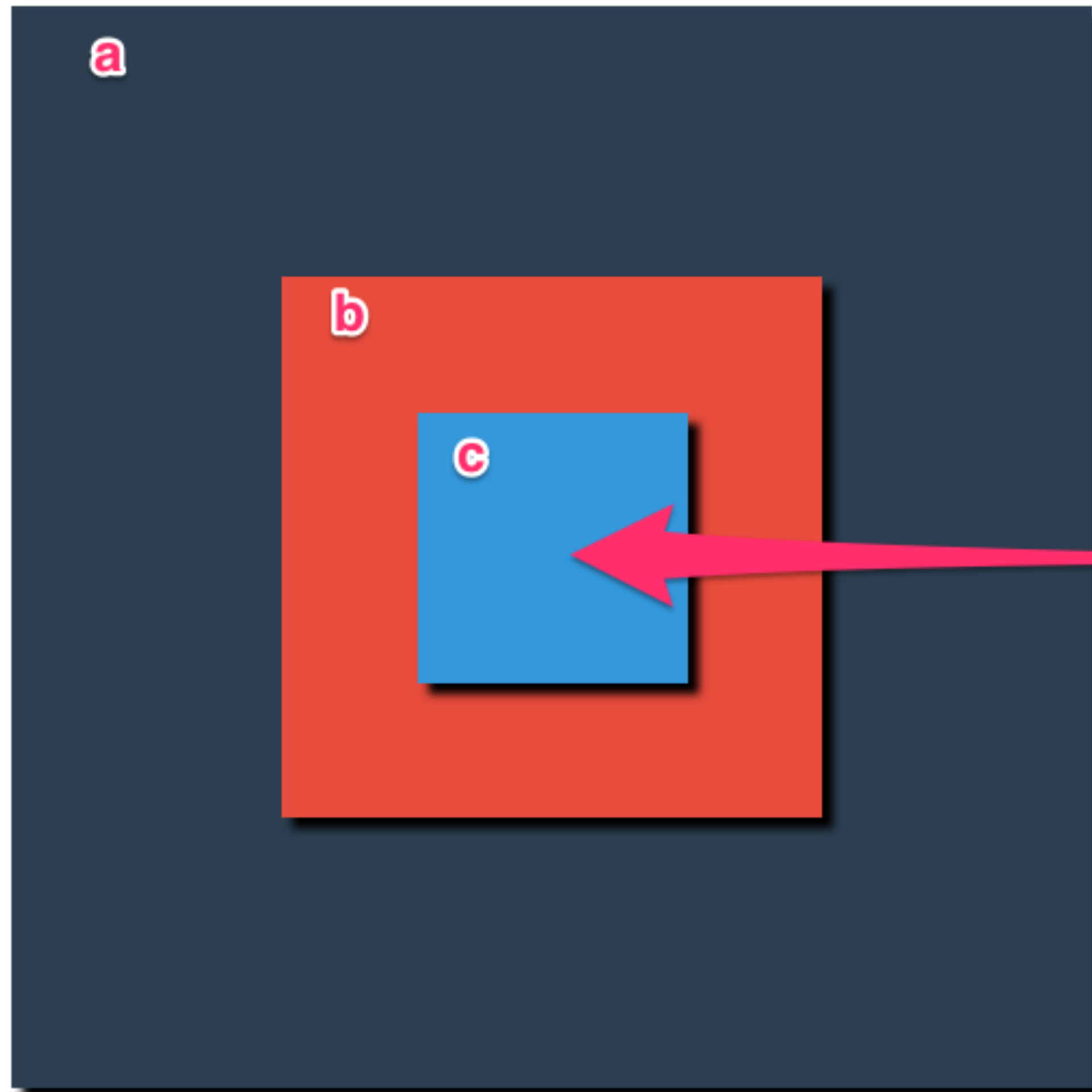
If I click here; who
responds first?

<- Think: bubble up;
capture down

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```


Demotime: event propagation

Event Propagation



What if I want to stop after c?

If I click here; who responds first?

<- Think: bubble up; capture down

```
<element>.addEventListener(  
  type, // event type e.g. click  
  callback, // your handler function  
  useCapture // bubble vs. capture (link)  
)
```


Bonus: Custom Events

USE ALL THE EVENTS

Goal:

- (1) Use MDN
- (2) Find cool events
- (3) Make amazing demos
- (4) Keep it neat:

One event / demo per function/div; Explain event on page; team names

Upload working code as copy/pastable text in Slack thread

Database events

Drag & Drop Events

CSS events

Notification events

Media Events

