

CS 252:

Advanced Programming Language Principles



Event-based Programming

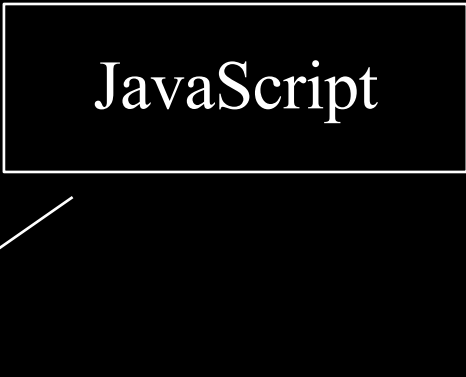
Prof. Tom Austin

San José State University

Review JS Intro Lab

Inline JavaScript

```
<html>  
  <input  
    type='button'  
    onclick='alert("Hello!");'  
    value='Say hi' />  
</html>
```



A diagram consisting of a white rectangular box with the text "JavaScript" inside. A thin white line extends from the bottom-left corner of the box, pointing towards the JavaScript code in the HTML snippet above.

Using in-line event handlers
is common, but considered
bad practice.

The "better" approach

```
<html>
  <input id='thebutton'
        type='button'
        value='Say hi' />
  <script type="text/javascript">
    var btn = document.
            getElementById('thebutton');
    btn.onclick = function() {
        alert('Groovy');
    };
  </script>
</html>
```

Note the
id attribute

Perhaps better still

```
<html>
  <input id='thebutton'
        type='button'
        value='Say hi' />
  <script type="text/javascript">
    var btn = document.
           getElementById('thebutton');
    function sayGroovy() {
      alert('Groovy');
    }
    btn.addEventListener('click', sayGroovy);
  </script>
</html>
```

```
function sayGroovy() {  
    alert('Groovy');  
}  
  
btn.addEventListener('click',  
                    sayGroovy);  
  
btn.addEventListener('click',  
    function() { alert("Bogus"); }  
);
```

```
function sayGroovy() {  
    alert('Groovy');  
    btn.removeEventListener('click',  
                                sayGroovy);  
}  
  
btn.addEventListener('click',  
                    sayGroovy);  
  
btn.addEventListener('click',  
    function() { alert("Bogus"); }  
);
```

Note that JavaScript (in a browser) is single threaded.

An event runs to completion before the next event begins.

Importing events in Node.js

Choose whatever
name you like

```
var EE =  
  require('events').EventEmitter;
```

events is
a module

```
var EE = require('events').EventEmitter;
var ee = new EE();

die = false;

ee.on('die', function() {
  console.log("I'm melting!!! Oh, what a world...");
  die = true;
});

setTimeout(function() {
  ee.emit('die');
}, 100);

while (!die) {}

console.log('done');
```

TCP Server example

```
var net = require('net');
var eol = require('os').EOL;

var srvr = net.createServer();

srvr.on('connection', function(client) {
  client.write('Hello there!' + eol);
  client.end();
});

srvr.listen(9000);
```

```
$ node tcpserver.js
```

```
$
```

```
$ nc 127.0.0.1 9000
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
$
```

```
Hello there!
```

```
Connection closed by foreign host.
```

```
$
```

Lab

- Lab: Write a chat server