

Is this a  
Nut?



It looks like a **Nut**, and cracks like a **Nut**, so sure, but if you really want to know, ask C++.



No, but it shares some of the same **traits**. Don't worry, I'll protect you from using it improperly.



Not technically, but it implements the **Crackable** interface so you can treat it like one.



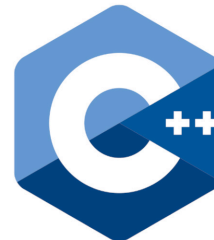
Doesn't matter. You can multiply it with **string** for all I care. It'll probably work.



I don't **classify** things. *You're* supposed to know what it is.



What Java said.



No, but we can work around that limitation if you're willing to risk the stability of the universe.

[https://www.reddit.com/r/ProgrammerHumor/  
comments/10uhk2o/programming\\_legumes\\_v20/](https://www.reddit.com/r/ProgrammerHumor/comments/10uhk2o/programming_legumes_v20/)

*JS doesn't tell you how to live your life. It just does what you told it to do to the best of its ability to make sense of your monkey code.*

—TurboGranny

*When you ask for heinous #@\*! other languages would squeal and cry and complain. But JS is like "LET'S #@\*!ING GOOOOO"*

—r00x

CS 252:

*Advanced Programming Language Principles*



# Event-based Programming

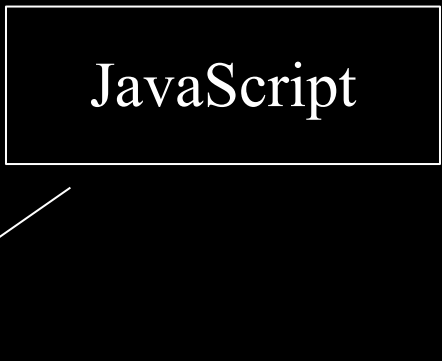
Prof. Tom Austin

San José State University

# Review JS Intro Lab

# Inline JavaScript

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



JavaScript

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"); }  
);
```

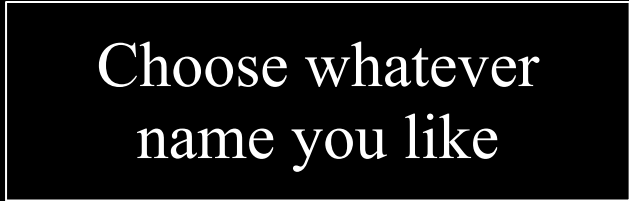
```
function sayGroovy() {  
    alert('Groovy');  
    btn.removeEventListener('click',  
                                sayGroovy);  
}  
btn.addEventListener('click',  
                    sayGroovy);  
btn.addEventListener('click',  
    function() { alert("Bogus"); }  
);
```

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

An event runs to completion before the next event begins.

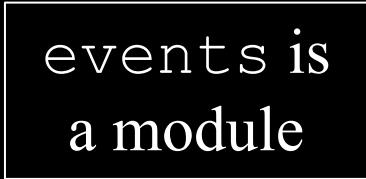
# Importing events in Node.js

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