EECS 368
Programming Language Paradigms

Dr. Andy Gill

Department of Electrical Engineering & Computer Science
University of Kansas

September 22, 2015

KU

JavaScript Types

- `string` — "Hello!"
- `number` — 64bit numbers
- `boolean` — true or false
- `object` — key-value pairs
- `function` — can be invoked
- `undefined`

KU

JavaScript Tricks

- It's all about objects
- methods are just functions attached to a specific object as a named field.
- A special variable `this` is assigned when you call a method.
- methods can:
  - change their object
  - change other objects
  - take objects (or numbers, booleans, strings or functions) as arguments.
  - return objects (or numbers, booleans, strings or functions).

KU

JavaScript's Purpose

JavaScipt exists to react to local changes inside a web browser

- Validate input before sending to a server
- GUI eye candy – shopping cart animation
- Presentation of information in sortable tables
- Basically low-latency tasks and offloading work from the web server

KU

JavaScript's purpose is to modify web pages

All changes are made throught the Document Object Model (DOM).

What is the Document Object Model?

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. [. . .]

(from `http://www.w3.org/DOM/#what`)

KU

DOM Concepts

- The idea is that the DOM is a tree of JavaScript objects.
- This tree can be read and modified.
- As you modify it, the changes are automatically reflected on the browser.

KU

Example HTML page

index.html

```
<html>
  <body>
    <h1>I am a title</h1>
    <ul id="mylist">
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
      <li>item 4</li>
    </ul>
  </body>
</html>
```

KU

Accessing the DOM

```
> document.children[0].children[1].children[1].children[2]
<li>item 3</li>
> var o = document.children[0].children[1].children[1].children[2]
undefined
> o
<li>item 3</li>
> o.innerText
"item 3"
> o.innerText = "Bla!"
"Bla"
```

KU

More accessing the DOM

```
> document.getElementsByTagName("li")
[<li>item 1</li>,<li>item 2</li>, <li>Bla!</li>, <li>item 4</li>]
> document.getElementById("mylist")
<ul id="mylist">...</ul>
> document.getElementById("mylist").children[2]
<li>Bla!</li>
```

All quite tedious!

JQuery

jQuery is the de facto way of accessing the DOM from JavaScript.

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

(from `https://jquery.com/`)

KU

Using JQuery

```
> $("#mylist")
[ <ul id="mylist">...</ul> ]
> $("li")
[<li>item 1</li>,<li>item 2</li>, <li>Bla!</li>, <li>item 4</li>]
> $("#mylist").append("<li>More</li>")
[...]
```

Much easier! Everything is done using `$(...)`.

JQuery can also effect CSS

```
> $("li").css("border","red 1px solid")
```

JQuery can also set up triggers

What does this do?

```
> $("li").on("click",function() {
        $(this).css("border","orange 1px solid");
  })
```

How does it work?