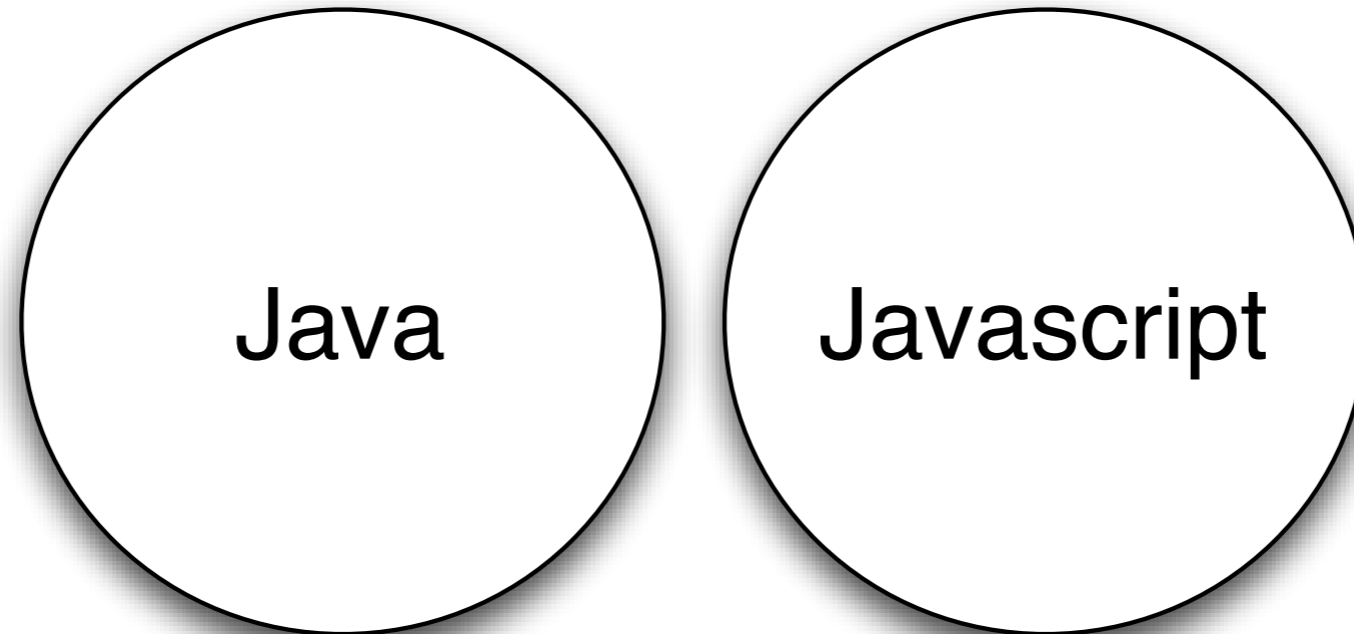


# User Interaction: Introduction to Javascript

Assoc. Professor Donald J. Patterson  
INF 133 Fall 2014





- Java and Javascript are not the same
  - They unfortunately share the same name
  - They are different in what they are designed to do
  - They are different in how they came to be
- Java is more powerful and complex

## How does a web page work?

- <http://www.ics.uci.edu/~djp3>
- Domain Names
- IP Addresses (IPv4 vs IPv6)
  - DHCP vs static IPs
- Routing a request
- a “web server”
- What is a web page?
- What does a browser do?
- HTML and CSS



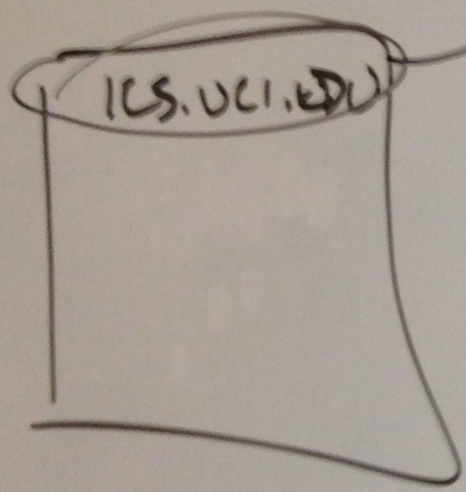
### Web Fundamentals

Create your own website and learn the building blocks of web development with HTML and CSS.

Explore >

<http://www.codecademy.com/learn>

128.195.183



BROWSER

O.S.

ISP

GOOGLE DNS

GLOBAL AUTHORITY

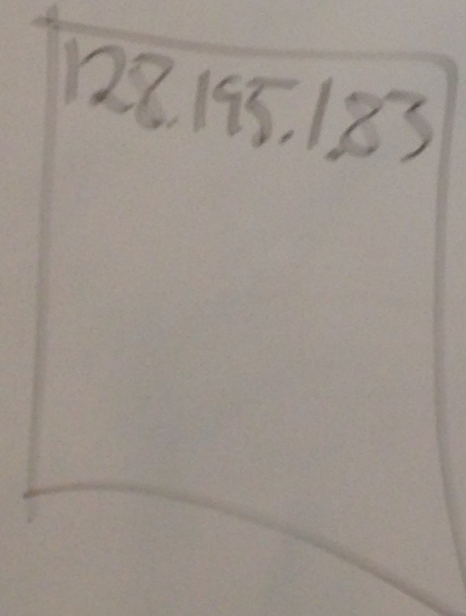
DIG + TRACE ADDRESS

GLOBAL

EDU

UCI

→ IPV4

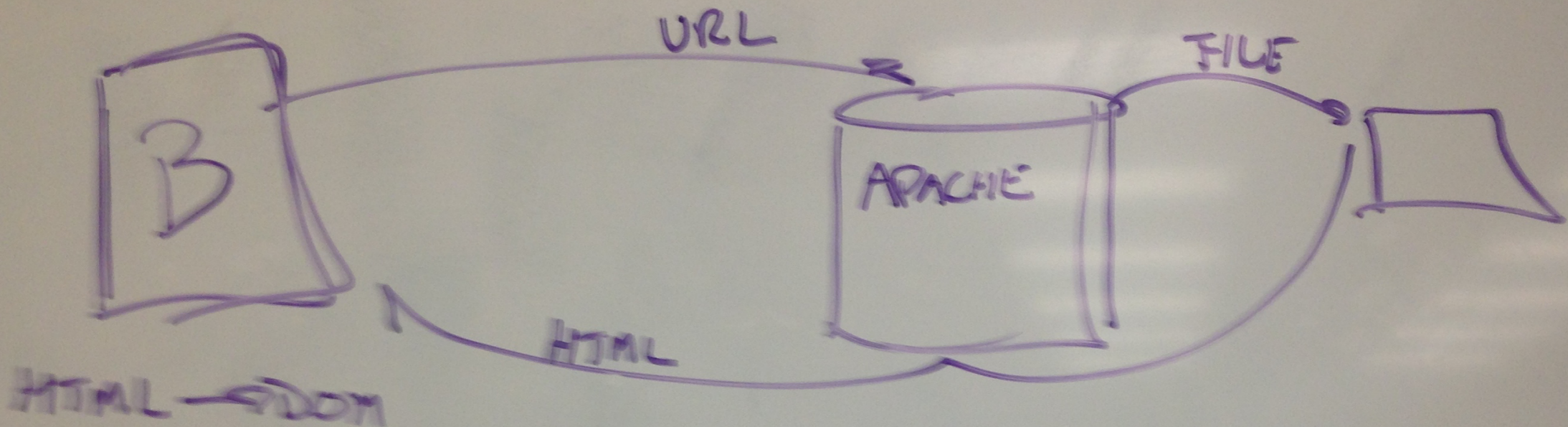


128.195.183

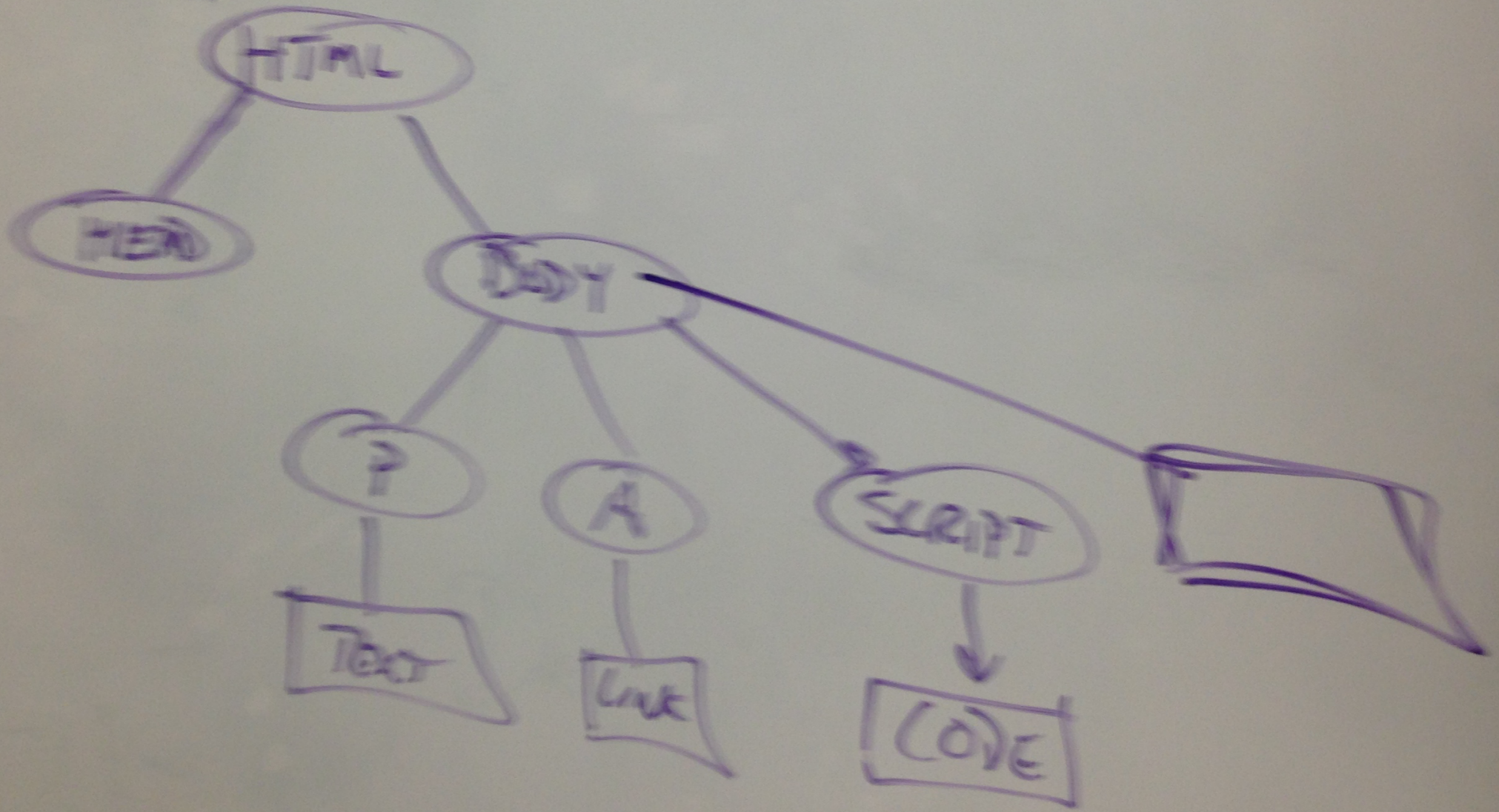
HTML  
DATA (XML)  
STRUCTURE

CSS  
PRESENTATION

JAVASCRIPT  
INTERACTIVITY



HTML → DOM



- JavaScript gives web designers a programming tool
  - Simple syntax
  - Supports putting "snippets" of code into web pages
- JavaScript can put dynamic text into an HTML page
  - A JavaScript statement like this:
    - `document.write("<h1>" + name + "</h1>")`
  - can write the content of a variable into a web page
- JavaScript can react to events
  - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element

Here is the most confusing thing



An HTML web page is the data that javascript works with

but

An HTML web page is also where the javascript is stored

and

the web browser is constantly redrawing the HTML (DOM)  
as the Javascript changes it.

- Javascript can read and write HTML elements
  - A Javascript can read and change the content of an HTML element
- JavaScript can be used to validate data
  - A JavaScript can be used to validate form data before it is submitted to a server.

- The data structure that is rendered by the browser is called
  - The **DOM**
    - Document Object Model
    - “is a platform- and language-neutral interface that allows 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.”

# Javascript

- get a Hello World running
  - This eliminates errors related to your hosting provider and web set up
  - It ensures that you have resources necessary on a web host

```
<html>  
  <head>  
    <title>Step 1</title>  
  </head>  
  <body>  
    Hello World  
  </body>  
</html>
```



- Let's put some Javascript in it
  - `<script>` tag
  - encloses javascript code that is run by the browser

```
<html>
  <head>
    <title>Step 2</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write("This is my first JavaScript!");
    </script>
  </body>
</html>
```

```
<html>
  <head>
    <title>Step 3</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write("<font color=\"red\">This is my second JavaScript!</font>");
    </script>
  </body>
</html>
```

- document.write()
  - when this command is run
  - the parameter is inserted into the html document just like it was written there without any javascript
- Compare “View Source” and “Inspect Element”
- Review Google Chrome tools

- JavaScripts in a page will be executed immediately while the page loads into the browser.
- This is not always what we want.
  - Sometimes we want to execute a script when a page loads, or at a later event, such as when a user clicks a button.
- Scripts to be executed when they are called, or when an event is triggered, are placed in functions.
- **TIP:** Put your functions in one place (e.g., the head section), this way they are all in one place, and they do not interfere with page content or confuse you.

# Javascript

```
<html>
  <head>
    <script type="text/javascript">
      function message()
      {
        alert("This alert box was called with the onload event");
      }
    </script>
  </head>

  <body onload="message()">
    Hello World!
  </body>
</html>
```

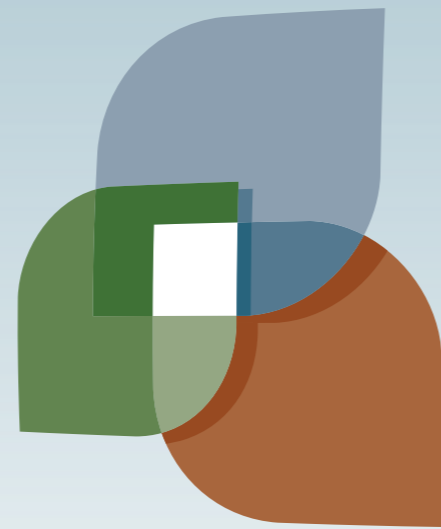
- Javascripts can be put in the head and the body
- Multiple javascripts are fine

- To reuse javascript in multiple different webpages
  - store it in another file
  - load it externally

```
function message()
{
    alert("This alert box was called from a remote library.");
}
```

```
<html>
  <head>
    <script type="text/javascript" src="Step5.js"></script>
  </head>

  <body onload="message()">
    Hello World!
  </body>
</html>
```



L U C I

