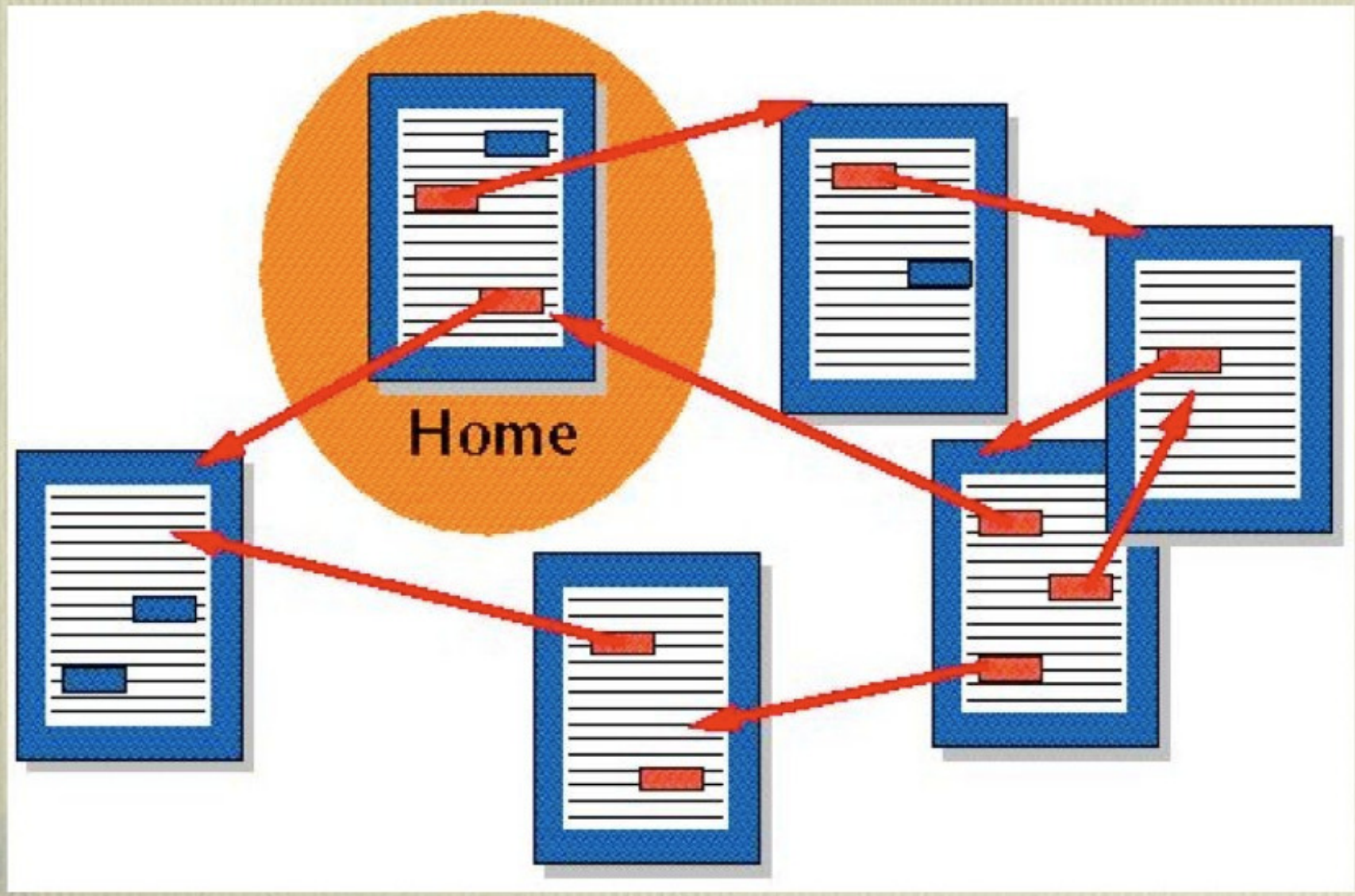
But First:
Review!

Hypertext



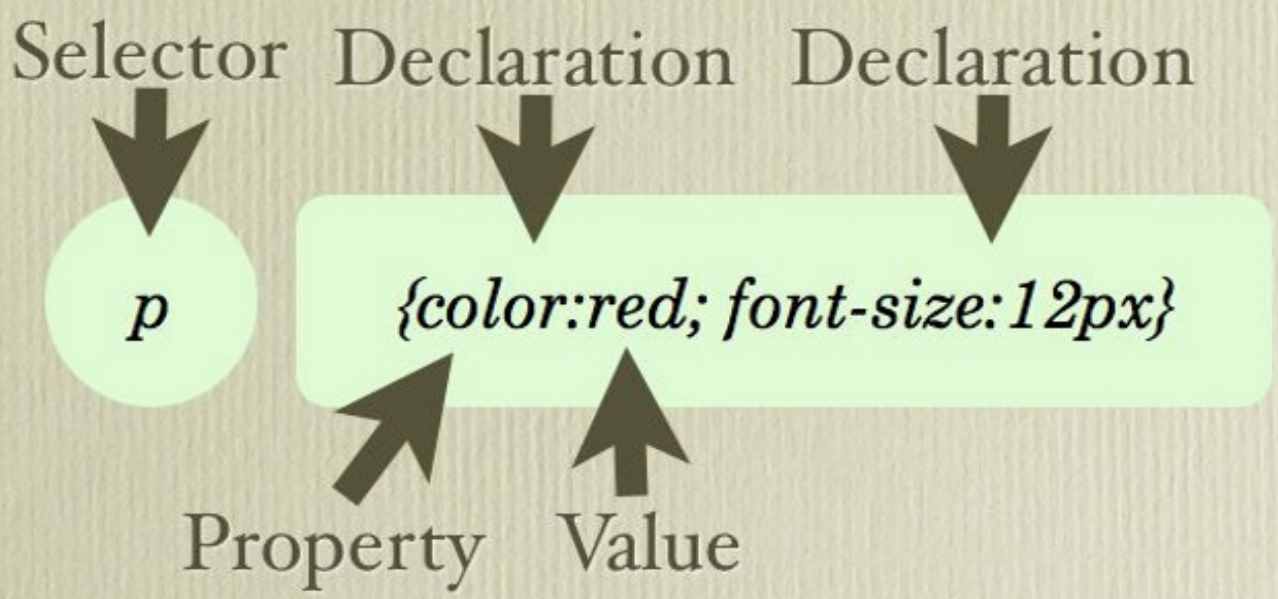
HTML Document Structure

```
<!DOCTYPE ...>
<html>
   <head>
      [HEAD ELEMENTS]
   </head>

   <body>
      [BODY ELEMENTS]
   </body>
</html>
```

CSS: Styles HTML

- Consists of Selectors and Declarations
- Declarations are Property:Value



JavaScript

- Used to Make Websites Dynamic
  - Reduced Latency
  - Richer Applications
  - Fewer Requests to the Server

JavaScript: Review

- Composed of Statements
  - End with ; or a newline
  - Executed In Order
  - Grouped into blocks
    { statement; statement;}
- Event Oriented
- Weakly Typed Variables
- Functional Scoping

JavaScript: Review

- variables: var x; var y = 10;
- Assignment Operators: = += -= *= /= %=
- Arithmetic Operators: + - * / % ++ --
- Logical Operators: && | | !
- Arrays: var x = new Array();
  var x= new Array("Nick", "Viola");
  var x = ["Nick", "Viola"];

JavaScript: Review

- Flow Control:
  - if (condition) { statements; }
  - if (condition) {} else {}
  - if (condition) {} else if {} else {}

JavaScript: Review

- while (condition) {}
- do {} while (condition)
- for (INIT; CONDITION; UPDATE) {}
- for (VAR in ARRAY) {}
- switch (VAR) {
  case VALUE:
  break;
  default:
  }

JavaScript: Review

- Functions give a name to a block
- Functions can have parameters
- Function can return a value
- function *functionName*(param1, param2,...)

```
{
    statements;
    return VALUE;
}
```

JavaScript: Error Handling

- try {
  // Code which may have problems
  } catch (err) {
  // Deal with the problem here
  }

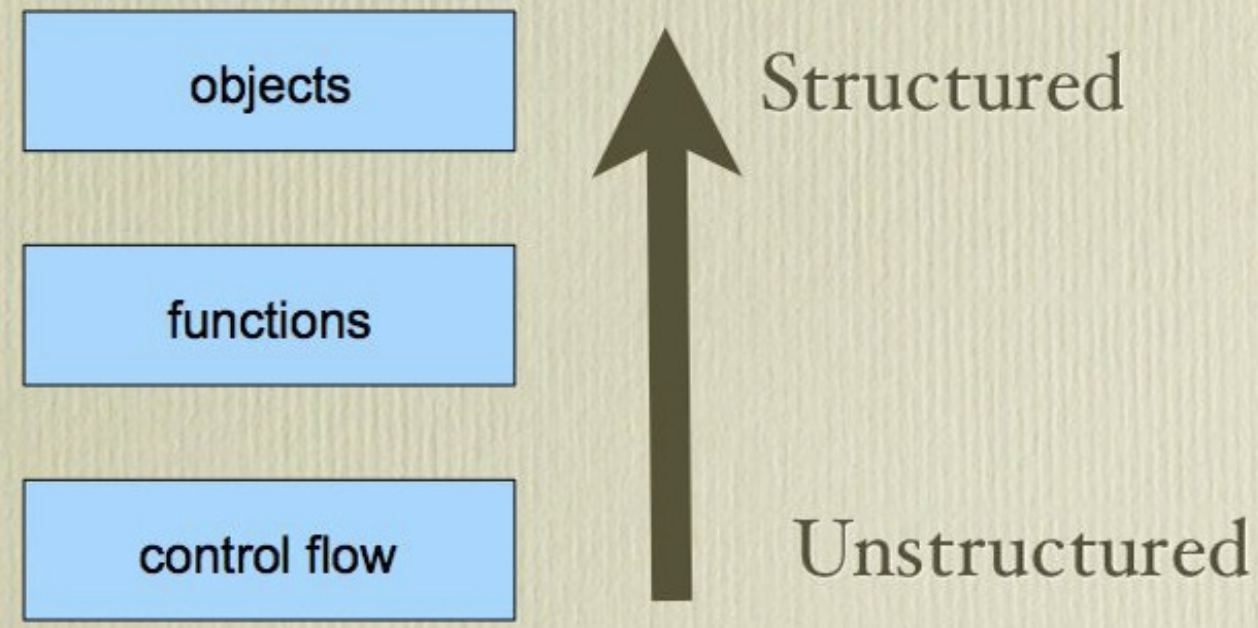
- throw err;
  allows you to jump to a catch

JavaScript

Object Oriented Programming

Object Oriented Programming

- Designed to make code more structured
- Designed to make code more modular



Intuition

- Think of Real World Objects

- Has Properties
  - Silver
- Can Do Things
  - Turn Left

JavaScript: Object Properties

- JavaScript is an Object Oriented Programming (OOP) Language
  - Objects are a “type” of value
  - Objects contain “sub-variables” called properties
    - var text = “Hello World!”
    - alert(“Hello World! length: “ + text.length);

JavaScript: Object Methods

- Objects also contain “functions” called methods
- Methods are functions which are performed on the associated object
  - var text = “Hello World!”;
    alert(text.toUpperCase());
- Called object is available within methods as the special variable ‘this’.

JavaScript: Constructing Objects

- function Person(first, last) {
  this.firstName = first;
  this.lastName = last;
  }

- var me = new Person("Nick", "Palmer");

- alert(me.firstName + " " + me.lastName);
  alert(me);

JavaScript: Objects

- All objects have their own state
- You can construct more than one of the same object
- function Person(first, last) {
  this.firstName = first;
  this.lastName = last;
  }
- var me = new Person("Nick", "Palmer");
  var girlfriend = new Person("Viola", "Caretti");

JavaScript: Object Methods

```
function Person(first, last) {
    this.firstName = first;
    this.lastName = last;
    this.toString = function() {
        return this.firstName + " "
                + this.lastName;
    }
}
```

- var me = new Person("Nick", "Palmer");
- alert(me.toString());
- alert(me);

Document Object Model

- global “window”
- window.status
- window.history
  - window.history.back();
- window.document (a.k.a document)
  - window.document.getElementById(“id”);
- window.location

Document Nodes are Objects

- Properties:
  - nodeName
  - nodeValue
  - parentNode
  - firstChild
  - style
  - etc...
- Methods:
  - appendChild()
  - removeChild()
  - click()
  - getAttribute()
  - removeAttribute()
  - etc...

JavaScript: With

- The “with” function re-scopes object properties
- Useful for scoping into object hierarchies in DOM
-
```
with (window.document.body.style){
  borderWidth=20;
  borderStyle='solid';
}
```
- window.document.body.style.borderWidth = 20;
  window.document.body.style.borderStyle=‘solid’;

Searching The DOM Tree

- document.getElementById(id)
  - Finds a single element with a given id
  - ids MUST be unique for a document to validate!
- document.getElementsByName(name)
  - Returns an array of elements with the given name.
  - In IE 6 will also include elements with matching ID and only form objects!

JavaScript: Validating Forms

```
function validate_required(field,alertText)
{
  with (field) {
    if (value==null||value=="") {
      alert(alertText);return false;
    } else {
      return true;
    }
  }
}
function validate_form(theForm)
{
  with (theForm) {
    if (validate_required(email,"Email must be filled out!")==false) {
      email.focus();return false;
    }
  }
}
```

```
<form action="submit.html"
onSubmit="return validate_form(this)"
method="post">
Email: <input type="text" name="email" size="30">
<input type="submit" value="Submit">
</form>
```

JavaScript: Validating Forms

```
function validate_email(field,alertText)
{
  with (field) {
    atPos=value.indexOf("@");
    dotPos=value.lastIndexOf(".");
    if (atPos<1||dotPos-atPos<2) {
      alert(alertText);return false;
    } else {
      return true;
    }
  }
}
```

JavaScript

Event Oriented Programming

rije Universiteit amsterdam

JavaScript: More on Events

- Netscape created JavaScript and the event model
- Internet Explorer has a different model
- W3C created the DOM Event Specification
- What a mess!

JavaScript: Event Models

- Differences in Handler Registration
- Differences in event bubbling
- Differences in event capturing

JavaScript: Event Registration

- inline: element.onclick = function() { alert("Hi!");};
  - Supported by all browsers
- attribute: <a onclick="alert("Hi!");">
  - Which element supports which attributes varies by browser

JavaScript: Multiple Registrations

- element.onclick = doSomething;
- element.onclick = doSomethingElse;
- Only the second will run!
- element.onclick = function() {
  doSomething(); doSomethingElse(); };
- What happens if doSomething() throws?
- How to unregister? Messy!

JavaScript: Advanced Event Registration

- W3C
  - element.addEventListener('click', doSomething, false);
  - third parameter determines order
    (More on this in a minute)
  - element.removeEventListener('click', doSomething, false);
- Microsoft Model
  - So broken it is useless!

JavaScript: Event Registration

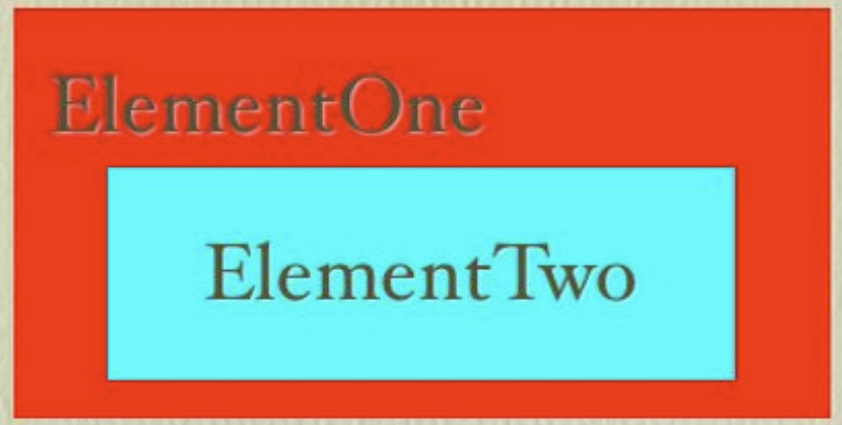
- Most developers use a framework
  - Prototype
  - jQuery (Topic of a future class if time!)

JavaScript: Accessing Events

- element.onclick = doSomething;
- the event is passed as a parameter to your function, except in IE
- IE uses window.event
- Cross browser:

```
function doSomething(e) {
  if (!e) var e = window.event;
}
```

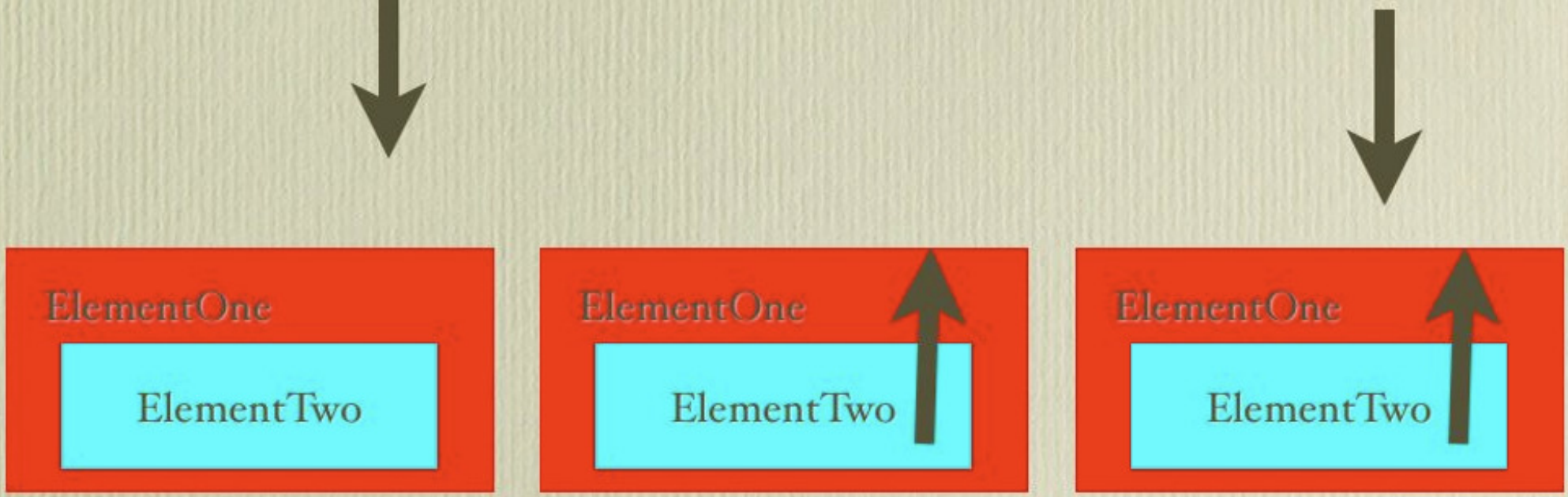
JavaScript: Event Order



- Click on ElementTwo!
  - Which gets the event?
    - Both!
  - In what order?
    - Depends on the browser!

JavaScript: Event Order

- Capture (Netscape)
- Bubble (IE)
- W3C



element.addEventListener('click', doSomething, false);
false = bubble true = capture

JavaScript: Event Order Why care?

- All events end up in the document
- Bubbling means you can implement default behaviors

JavaScript: Stopping Events

- MS Model
  window.event.cancelBubble = true;
- W3C Model
  e.stopPropagation();
- Cross Browser
  function doSomething(e) {
    if (!e) var e = window.event;
    e.cancelBubble = true;
    if (e.stopPropagation) e.stopPropagation();
  }

JavaScript: Setting Cookies

```
function setCookie(c_name,value,expiredays)
{
    var exdate=new Date();
    exdate.setDate(exdate.getDate()+expiredays);
    document.cookie=c_name+ "=" +escape(value)+
        ((expiredays==null) ? ""
         :";expires="+exdate.toGMTString());
}
```

JavaScript: Getting Cookies

```
function getCookie(c_name) {
if (document.cookie.length>0) {
    c_start=document.cookie.indexOf(c_name + "=");
    if (c_start != -1) {
        c_start=c_start + c_name.length+1;
        c_end=document.cookie.indexOf(";",c_start);
        if (c_end == -1) c_end=document.cookie.length;
        return unescape(
            document.cookie.substring(c_start,c_end));
        }
    }
return "";
}
```

JavaScript: ActionScript

- ActionScript is language used in Flash
- Very similar to JavaScript
- Learning JavaScript makes learning Flash very easy!