APPLICATION SERVER

SERVER PAGES (JSP, ASP, PHP, etc)

APPLICATION LOGIC

DATABASE (MSSQL, MySQL, Oracle)

REQUEST

HTTP

REPLY

BROWSER CLIENT

Structured content

Visual presentation

HTML

CSS

JavaScript

JQuery

Behavior: Modifying DOM structure, Adding Interactivity

Video / Audio support, Local data storage, and CANVAS!

Focus of this tutorial

PlugIn that makes Javascript much easier

HTML 5

function call

return value

JavaScript function call return value

Behavior: Modifying DOM structure, Adding Interactivity

Focus of this tutorial

PlugIn that makes Javascript much easier

HTML 5

function call

return value
### Three layers of front-end web

<table>
<thead>
<tr>
<th><strong>Behavior</strong></th>
<th><strong>JavaScript</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Event Handling (click, drag&amp;drop, etc)</td>
</tr>
<tr>
<td></td>
<td>• Modifying <em>Presentation</em>, <em>Content</em> dynamically</td>
</tr>
<tr>
<td></td>
<td>• Communicate with server (no page reload)</td>
</tr>
<tr>
<td></td>
<td>• Complex functions (e.g. playing game)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Presentation</strong></th>
<th><strong>CSS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Defines how content will appear</td>
</tr>
<tr>
<td></td>
<td>• Dimension, Color, Font, Align, Image, Style, etc</td>
</tr>
<tr>
<td></td>
<td>• <strong>Tag</strong>, <strong>ID</strong> and <strong>CLASS</strong> match content to presentation</td>
</tr>
<tr>
<td></td>
<td>• Different CSS for different browser, platform</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Content</strong></th>
<th><strong>HTML</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Defines information (text, image) to show</td>
</tr>
<tr>
<td></td>
<td>• Use <code>&lt;DIV&gt;</code>, <code>&lt;SPAN&gt;</code>, <code>&lt;P&gt;</code>, <code>&lt;FORM&gt;</code>, <code>&lt;UL&gt;</code>, <code>&lt;OL&gt;</code>, <code>&lt;LI&gt;</code>,</td>
</tr>
<tr>
<td></td>
<td>• <strong>DO NOT USE</strong> <code>&lt;TABLE&gt;</code>, <code>&lt;FONT&gt;</code>, <code>&lt;BR&gt;</code>, style tags (&lt;B&gt;, &lt;I&gt;, &lt;U&gt;), <code>style</code>, <code>align</code>, <code>color</code>, <code>bgcolor</code> attributes.</td>
</tr>
</tbody>
</table>
DOM Document Object Model

- DOM is a **hierarchical structure** (map of nodes) used to **access**, **traverse** and **manipulate** HTML and XML documents.

  - Use FireBug (firefox plugin) to explore DOM structure of simple HTML file.
DOM Document Object Model

- DOM API is a **hierarchical structure** (map of nodes) used to **access**, **traverse** and **manipulate** HTML and XML documents

- Window.document represents the entire document.

- Let’s explore DOM structure using Firebug or Chrome Element inspector
CSS Cascading Style Sheet

- What CSS can do?
  - Everything about Look&feel.
  - Use Firebug or Chrome *element Inspector* to explore CSS
  - Visit CSS Zen Garden [css zen garden](#)
  - Visit [w3schools.com](http://w3schools.com) and try *Try it yourself>*

- more selectors on [w3school ref](http://w3school.com)
CSS Where to put my style

**Inline style**

```
<p style="color:sienna; margin-left: 20px">This is a paragraph.</p>
```

- Suitable for quick edit
- Hardly sustainable

**Internal Style Sheet**

```
<head>
<style type="text/css">
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>
```

**External Style Sheet**

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

- Best for modularity
- Faster Loading
- Hard to start from scratch
  - You can use CSS frameworks

What if multiple styles conflict?

**inline > internal** (lower the stronger) **> external**
CSS layout demo
1. `<div>` only

```
<Tak Yeon Lee
  my picture here
  about.
  I am a PhD candidate in Computer Science department at University of Maryland, College Park, and a member of the Human-Computer Interaction Lab. As an interaction designer, HCI(Human-Computer Interaction) researcher and Computer Scientist, my research interests are computational thinking, machine learning, infovis, and (of course) Human-Computer Interaction. contact.
  +1 (202) 330 2506, tylee@umd.edu interest.
  I enjoy listening modern folk, atmospheric, and world music.
</Tak Yeon Lee>```
2. `<div>` with class and id

- **id**: unique. one for each object only
- **class**: multiple objects. each object can have multiple classes
  - `<div class="column_3 standOut launchGallery">` _classes for style and click event_

- Semantic vs. non-semantic class name
  - `<div class="title">` _semantic. recommended but not always applicable_
  - `<div class="column_1">` _non-semantic, but every CSS grid framework use this_
  - `<div class="largeText">` _non-semantic. Use ‘standOut’ or ‘confirmedName’ instead._
3. add container div for top-level layout

• If you don’t specify or set any dimension ‘auto’, then something either smart or nasty happens.

Note: Using margin:auto will not work in IE8 and earlier, unless a !DOCTYPE is declared.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. complicated layout using nested div

```html
<html>
  <head>
    <!-- include CSS and JS files here -->
    <!-- define javascript functions here -->
  </head>
  <body>
    <div id="container" style="margin:50px auto; width:400px; border:3px dotted green;">
      <div id="leftColumn" style="float:left; width:30%;">
        <div id="myName">
          Tak Yeon Lee
        </div>
        <div id="myPicture">
          my picture here
        </div>
      </div>
      <div id="rightColumn" style="float:left; width:70%;">
        <div class="paragraphTitle">
          about.
        </div>
        <div class="paragraphText">
          I am a PhD candidate in Computer Science department at University of Maryland, College Park, and a member of the Human-Computer Interaction Lab. As an interaction designer, HCI(Human-Computer Interaction) researcher, and Computer Scientist, my research interests are computational thinking, machine learning, infovis, and (of course) Human-Computer Interaction.
        </div>
        <div class="paragraphTitle">
          contact.
        </div>
        <div class="paragraphText">
          +1 (202) 330 2506, tylee@umd.edu
        </div>
        <div class="paragraphTitle">
          interest.
        </div>
        <div class="paragraphText">
          I enjoy listening modern folk, atmospheric, and world music.
        </div>
    </div>
  </body>
</html>
```
Basic CSS layout technique using **position, float, clear**

No layout

```
.container {
  ...
}
.leftColumn {
}
.rightColumn {
}
```
Floating element and wrapping text

when floating, unspecified size will shrink to its content

.container { ... }
.leftColumn { float:left }
.rightColumn { }

Tak Yeon Lee about.
my picture here
I am a PhD candidate in Computer Science
department at University of Maryland, College Park, and a
member of the Human-Computer Interaction Lab. As an
interaction designer, HCI(Human-Computer Interaction)
researcher and Computer Scientist, my research interests are
computational thinking, machine learning, infovis, and (of
course) Human-Computer Interaction.
contact.
+1 (202) 330 2506,tylee@umd.edu
interest.
I enjoy listening modern folk, atmospheric, and world music.
If parent’s width allows, floating elements stacks horizontally.
Clearing floats

```html
<DIV class='leftColumn'>...
<DIV class='rightColumn'>...
<DIV class='footer'>
  this is footer
</DIV>
```

I want my footer be here below all the floating elements
Clearing floats

Now the parent object knows the size of its content.
Absolute positioning

```css
.container {
    position: relative;
}
.leftColumn {
    position: absolute;
    top: 20px; left: 20px;
    z-index: 5;
}
.rightColumn {
    width: 300px;
    position: absolute;
    top: -20px; right: 10px;
}
.footer {
    position: absolute;
    bottom: 10px; left: -10px;
}
```

without this, the reference position will be the entire document

it puts left column on top of all other elements

let’s get back to the main tutorial

about. I am a PhD candidate in Computer Science Department at University of Maryland, College Park, and a member of the Human-Computer Interaction Lab. As an interaction designer, HCI(Human-Computer Interaction) researcher and Computer Scientist, my research interests are computational thinking, machine learning, infovis, and (of course) Human-Computer Interaction.

contact. +1 (202) 330 2506, tylee@umd.edu

interest. I enjoy listening modern folk, atmospheric, and world music.
5. Box Model

why everything look so cluttered?

It needs Margin, Padding!
5. Box Model

Why the right column pushed down?

width:30%

Actual width take by element = width + padding + border + margin
5. Box Model

Box model is inconvenient. Element width should include padding, and I want to set the inner margin!

user `nested div` having margin

```html
<div class='outer'>
  <div class='inner'>
    Lorem ipsum...
  </div>
</div>
```

`width: auto` is applied by default, which takes the maximum possible width within the parent element.

```css
.outer{
  width:200px
}
.inner{
  margin: 10px;
}
```
5. Box Model

we still have 100% width in total

if no fixed-width is needed, just use padding.

wrap all the content with ‘inner’ div
6. Flexible width / height

- **width:auto (or unspecified)**
  1. floating
     - **minimum** to enclose children
  2. no floating
     - **maximum** within its parent
- **width: x%**
  - takes x% of parent’s width
- **overflow: visible / hidden / scroll**

- **height: x%**
  - takes x% of parent’s height
- **height: auto** (no auto option for height)
Doctype

- appears at the top of HTML page
- tells browser to use different legacy modes

- Standard mode with cutting edge validation (HTML5)
  ```html
  <!DOCTYPE html>
  ```

- Almost standard mode
  ```html
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
  ```

- Standard mode, but stick to legacy validation
  ```html
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://
  www.w3.org/TR/html4/strict.dtd">
  ```

- Quirks mode (don’t use this unless you care about IE6)
  ```html
  no doctype needed
  ```
QUESTION??
JavaScript

- Dynamic, Loosely typed, Prototype-based programming language
- Many usage
  - client-side web programming
  - plug-in for IDEs, Browser, PDF
  - Facebook app
  - Other higher-abstraction usage
- Easy to use libraries (Jquery)
JavaScript Basics  >  TYPE

Number

```javascript
// Note: ALWAYS use 'var' to declare a variable:
var leftSide = 100;
var topSide = 50;
var areaOfRectangle = leftSide * topSide; // = 5000
```

String

```javascript
var firstPart = 'Hello';
var secondPart = 'World!';
var allOfIt = firstPart + ' ' + secondPart; // Hello World!
```

Boolean

```javascript
5 === (3 + 2); // = true
// You can assign Boolean values to variables:
var veryTired = true;
// You can test for it like this:
if (veryTired) {
    // Sleep
```

Array

```javascript
// Literal:
var fruit = ['apple', 'lemon', 'banana'];

// Using the Array constructor:
var fruit = new Array('apple', 'lemon', 'banana');

fruit[0]; // Access the 1st item of the array (apple)
fruit[1]; // Access the 2nd item of the array (lemon)
fruit[2]; // Access the 3rd item of the array (banana)
```
JavaScript Basics > TYPE

**Function**

```javascript
// Using the function operator to create a function
function myFunctionName(arg1, arg2) {
    // Function code goes here.
}

// If you omit the function's name then
// you're creating an "anonymous function":
function(arg1, arg2) {
    // Function code goes here.
}

// Running a function is simply a case of referencing it
// and then adding a parenthesis (with arguments):
myFunctionName(); // No arguments
myFunctionName('foo', 'bar'); // with arguments
```

We’ll see this a lot as CallBack functions of Ajax call

**Object**

Contains pairs of key and value

DOM consists of objects

```javascript
// 2 different ways of declaring a new Object.

// Literal (curly braces):
var profile = {
    name: 'Bob',
    age: 99,
    job: 'Freelance Hitman'
};

// Using the Object constructor:
var profile = new Object();
profile.name = 'Bob';
profile.age = 99;
profile.job = 'Freelance Hitman';
```

Similar to HashMap (java) or Dictionary (python)

Directly convertible to JSON