WebDev Essentials

Katerina Intzevidou
aintzevi@csd.auth.gr

ACM Student Chapter AuTh

Building a static page

Client-Side

- HTML (structure your content)
- CSS (format content, make it pretty)
- JS (add interactive effects within browser)

Getting started...

...you need to pick an editor

- Notepad++
- Atom (#ShowSomeAtomLove)
- Sublime
- Vim
- JetBrains products (WebStorm, PhpStorm)
- ...

HyperText Markup Language
aka HTML

HTML Basics (and some jargon)

HTML Files - Each page is a file with .html extension

HTML Tags - Keywords inside <brackets>

e.g. open tag `<title>` close tag `</title>`

Tags can have attributes - specified in the opening tag

e.g. src and alt -

```
<img src="http://images2.memedroid.com/image.
  jpeg" alt="meowtain">
```

HTML Basics (and some jargon)

HTML Elements - Opening/closing tag, with their content

e.g. `<title>My First Page</title>`

Void elements (Self-closing)

e.g. `<br/>, <img/>`

In HTML5 the slash (/) is optional, so self-closing tags, are tags without a closing part

`<br>` is the same as `<br/>`

```
<!DOCTYPE html>
<html lang=”en”>
    <head>
        <meta charset="utf-8">
        <title>My First Page</title>
        <!--You can add links here-->
    </head>
    <body>
        <!--Yet Another Useless Comment-->
    </body>
</html>
```

Block and Inline elements

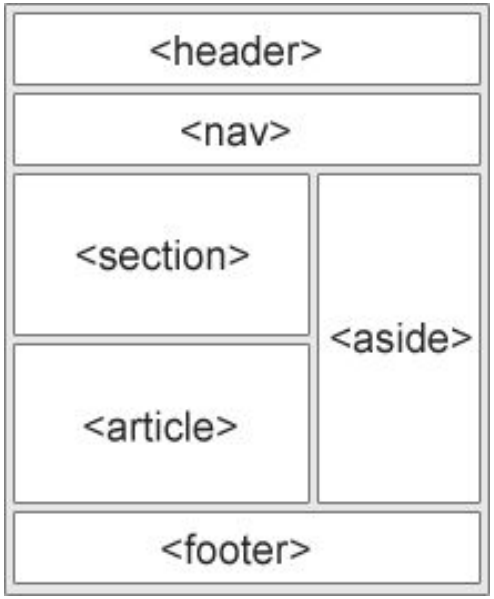
Block element - Starts on new line, takes up full width

`<p>, <div>, <h1> to <h6>, <address>, ...`

Inline element - Does not start on new line, takes up only necessary width

`<span>, <a>, <img>, <em>, <strong>, <small>, ...`

Some new structure tags



Links

- Links using absolute URL - to other sites or our own

```
Press <a href="http://this-is-a-link.com " 
  >HERE</a>
```

- Local links (relative URL)

```
<a href="about.html">About us</a>
```

- Links to the same page

```
<h1 id="here">Desired Point</h1>
<a href="#here">Get to desired point</a>
```

Lists

`<ul></ul>` - unordered list

`<ol></ol>` - ordered list

`<li></li>` - list element

Tables

`<table></table>` - define a table

`<th></th>` - header cell

`<tr></tr>` - table row

`<td></td>` - table cell

Cascading Style Sheets aka CSS

CSS is...

...a style language that defines the layout of HTML documents

It covers fonts, colours, margins, lines, background images, advanced positions, etc

To add CSS to your page...

Create a file with a .css extension in a folder named style

e.g. styles.css

Add this link to your .html file

```
<head>
   <link rel="stylesheet" href="style/styles.css"
  >
</head>
```

CSS Selectors

.class - e.g. .element - Selects all elements with class=" element"

#id - e.g. #element- Selects the unique element with id=" element"

element - e.g. p - Selects all <p> (paragraph) tags

```
element {
    property1: value1;
    /* Comment */
    property2: value2;
}
```

Grouping - p, div, h1 { color: blue; font-size: 12px; }

Nesting - p inside a div - div p { color: red; }

Multiple selectors - class red p - p.red { color: red; }

Child - paragraph with a parent div - div > p { color: red; }

Size

Absolute size

- in (inches), cm (centimeters), (mm) millimeters
- pt (points) = 1/72 inches
- pc (picas) = 1/6 inches
- px (CSS pixels) = 1/96 inches (96 dpi)

Relative size

- % of the parent's width
- em, dependent on parent's text size
- rem, dependent on root element's text size

Display

- block - Display as block-level element
- inline - Display as inline-level element
- inline-block - Elements that are like inline elements but also have width and height
- …

Display vs Visibility

Display sets the type of the element and can take it off the natural flow of the page (display: none;)

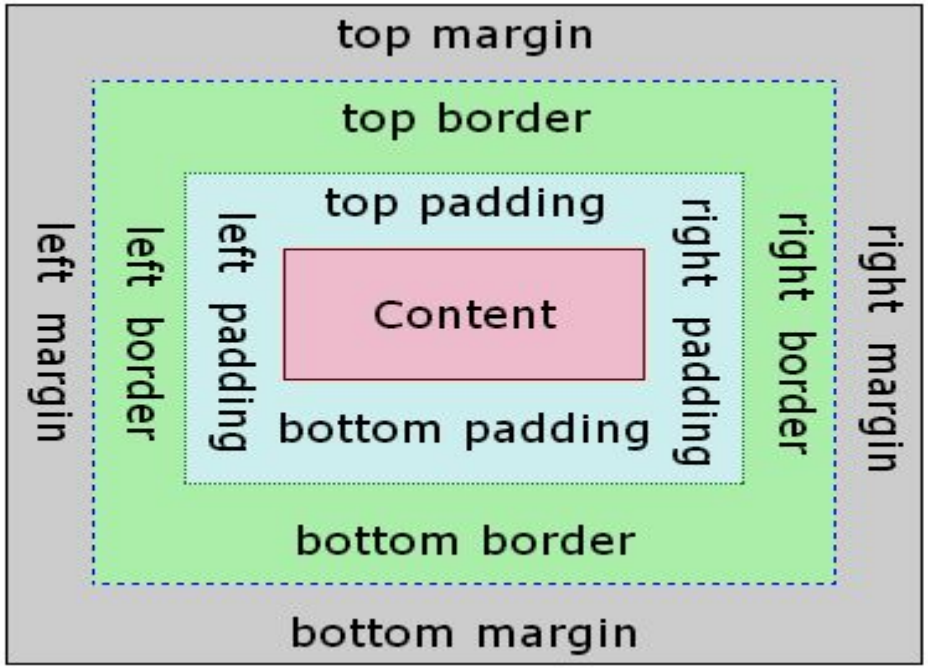
Visibility specifies whether or not an element is visible

Position

- static - Default, according to the page's natural flow
- relative - Positioned relative to its normal position
- absolute - positioned relative to the nearest positioned ancestor
- fixed - Positioned relative to viewport (e.g. browser)

If there is overlapping, the z-index property sets the order

The Box Model



Transition property

A typical CSS transition defines the following:

- Property to apply the transition to
- Duration of the transition
- Timing function to use

e.g. transition-property: width;
transition-duration: 2s;
transition-timing-function: linear;
transition-delay: 1s;

Or transition: width 2s linear 1s;

CSS Preprocessors

Sass is an extension of CSS3, adding nested rules, variables, mixins, selector inheritance, and more.

It's translated to well-formatted, standard CSS

Two syntaxes

- SCSS (more CSS like) and
- Sass (the indented syntax)

No need to do everything from scratch

CSS Frameworks

- Bootstrap
- Foundation
- Semantic UI
- Pure
- Skeleton
- etc

Creating pretty, mobile responsive websites easily!

JavaScript aka JS

JavaScript is...

...a lightweight, client-side interpreted programming language that allows you to build interactivity into static HTML pages.

JavaScript can...

- Validate user input on the web browser - less server interaction
- Change HTML content, attributes and style (CSS)
- …

To add JS to your page...

Create a file with a .js extension in js folder

e.g. scripts.js

Add this script to your .html file like this

```
<script src="js/scripts.js"></script>
```

This can be added anywhere in either the <head> or the <body> of the page

JS Jargon

Objects - Windows, docs, images, tables, …

Properties - Object attributes e.g. document.forms

Methods - Actions applied to particular objects e.g.
document.write(“Hello World!”)

Functions - Named statements that performs tasks

e.g. function foo(p1, p2) { return p1*p2 }

```
3.9                    // numeric literal
"Hello!"               // string literal
false                  // boolean literal
null                   // literal null value
{x:1, y:2}             // Object literal
[1,2,3]                // Array literal
function(x){return x*x;}  // function literal
var example, $number, example_2, Example
example = “Hello World”; // Variable assignment
```

JS is loosely typed - example = 2;

Operators

+, -, /, *, =

tries++; ++tries; tries--; --tries;

“String ” + “concatination”

==,!=, <,>,<=,>=,

===, !== (comparison of both value and type)

&&, ||, !

Statements

- if (c1 < c2) { statement1; statement2 } else {statement}
- switch(n) {case 1: statement1; break; case 2 : statement2; break; default : statement;}

- for (var i=1 ; i < 10 ; i++) { statements; }
- var i = 0;

  while (i < 10) { statements; i++;}
- var i=0;

  do { statements; } while (i < 10);

Arrays

- scores = new Array(2);

  scores[0] = 39;

  scores[1] = 18;
- scores = new Array(39, 18, 95, 45);
- scores = [21, 2];
- test = new Array();

  test[0] = 21;

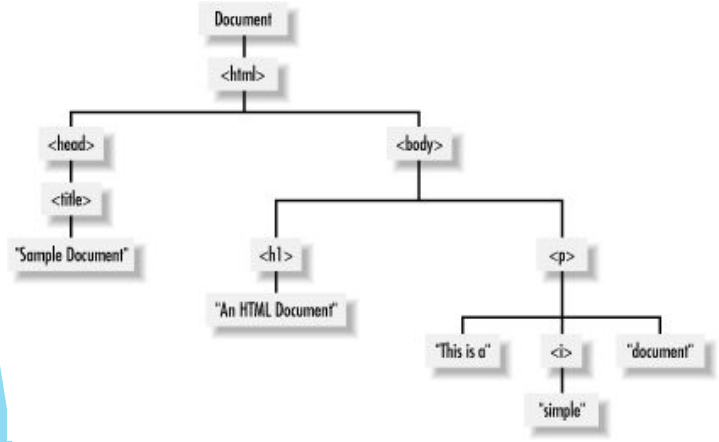
  test[5] = 22;

Functions

```
function foo() {
    alert(“Hello World!”);
}
 var bar = function(name) {
    alert(“Hey” + name);
}
function add(x,y) {
    return x+y;
}
```

DOM (Document Object Model)

Everything is a node, the document itself, HTML elements/attributes, text/comments



BOM (Browser Object Model)

Methods and properties for JavaScript interactivity - no standard implementation

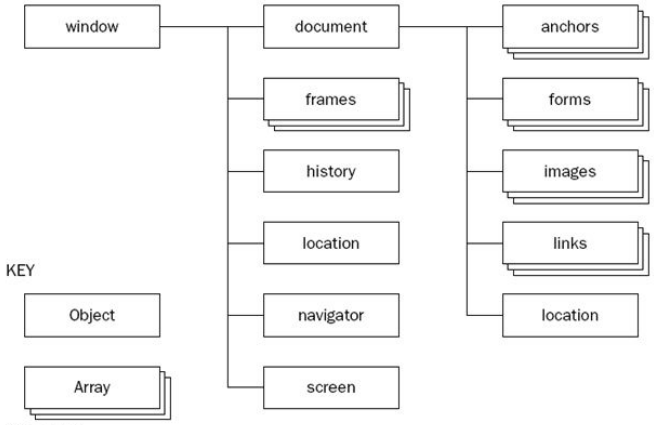


Figure 5-3

The window Object

Represents the browser window

Dialog boxes (not to be used often)

window.alert(String)
window.confirm(String)
window.prompt(String message, [String default])

Finding HTML elements

- Finding HTML elements by id

  e.g. var myElement = document.getElementById("intro");
- Finding HTML elements by tag name

  e.g. var x = document.getElementsByTagName("p");
- Finding HTML elements by class name

  e.g. var x = document.getElementsByClassName("intro");
- Finding HTML elements by CSS selectors

  e.g. var x = document.querySelectorAll("p.intro");
- Finding HTML elements by HTML object collections (like form elements list)

Some Events

Page

- onload

Mouse

- onclick
- ondbclick
- onmouseover
- onmousedown

Forms

- onsubmit
- onreset

Etc

Links and further studying

ACM AuTh Chapter site - http://acm.web.auth.gr/

Editors

- Notepad++ - https://notepad-plus-plus.org/
- Atom - https://atom.io/
- Sublime - https://www.sublimetext.com/
- Vim - http://www.vim.org/
- JetBrains Products - https://www.jetbrains.com/products.html

Validator - https://validator.w3.org/

HTML5 Standards - https://www.w3.org/TR/html5/

Links and further studying

Tutorials

- Codecademy - https://www.codecademy.com/
- Shay Howe - http://learn.shayhowe.com/
- HTML Dog - http://htmldog.com/
- W3Schools - http://www.w3schools.com/

HTML Tags - http://www.w3schools.com/tags/

CSS Properties - http://www.w3schools.com/cssref/

JS Reference - http://www.w3schools.com/jsref/

JS Libraries - https://www.javascripting.com/

Links and further studying

CSS Preprocessors (Sass, SCSS, LESS) -

https://www.rechnerhaus.de/en/difference-between-css-scss-sass

http://www.sitepoint.com/whats-difference-sass-scss/

https://css-tricks.com/sass-vs-less/

Inspiration

- Codepen - http://codepen.io/
- CSS-tricks - https://css-tricks.com/snippets/
- CSSDSGN - http://www.cssdsgn.com/
- Dribbble Designs - https://dribbble.com/

Any Questions?

Katerina Intzevidou
aintzevi@csd.auth.gr