Introduction to Web scraping

Nov 2017

Agenda

- HTML basic
- Use of rvest package and selector gadget under chrome browser to perform basic web scraping
- How to scrap Javascript rendered web content

HTML basic

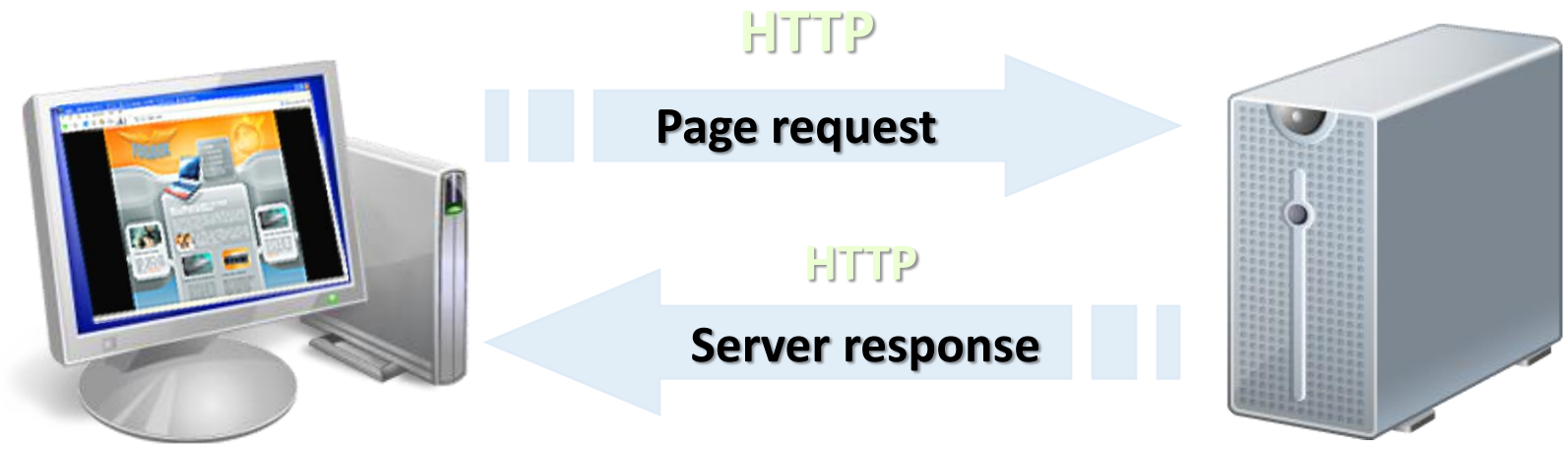
- How web works
- Overall HTML page layout
- Basic of HTML table
- Basic of Javascript
- Basic of CSS

How web works

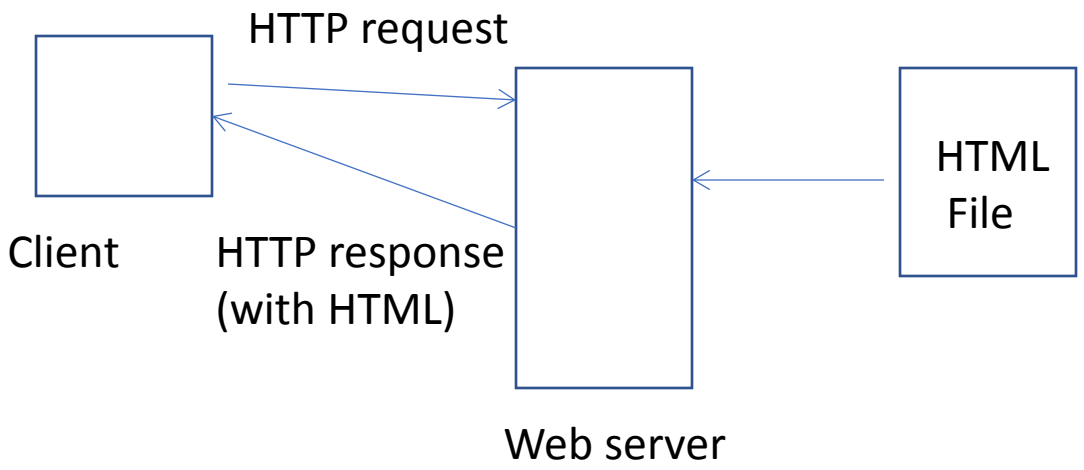
- Use of HTTP protocols
- Static web pages against dynamic web pages
- Server side scripting languages
- Client side scripting language

How the Web Works?

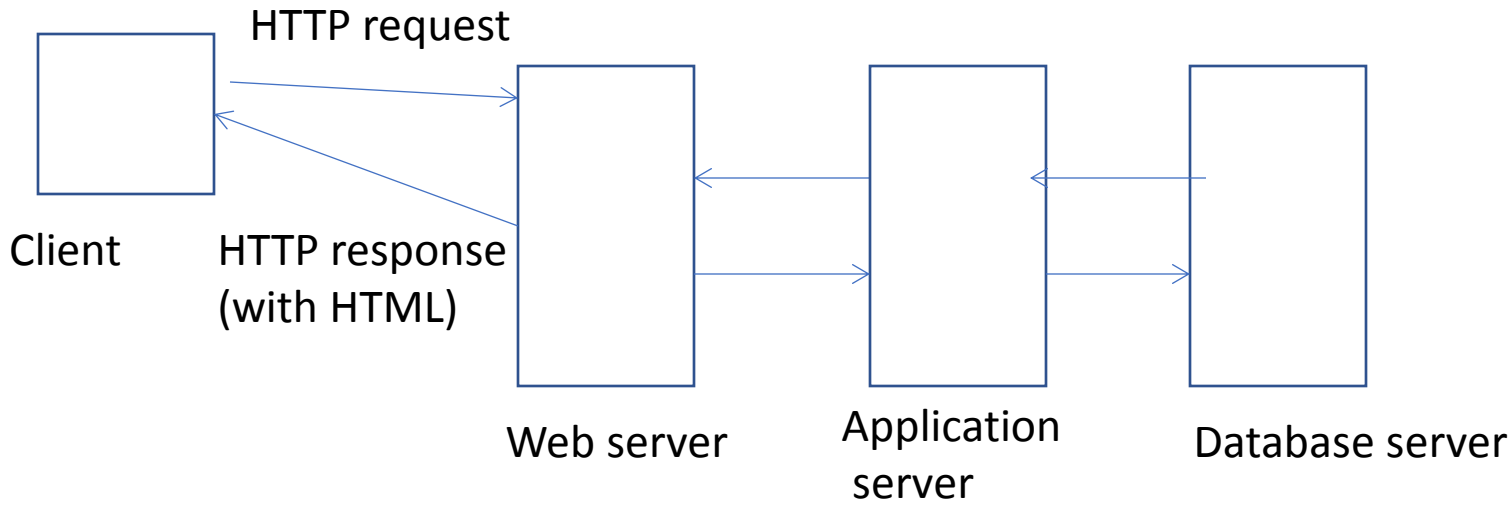
- WWW use classical client / server architecture
  - HTTP is text-based request-response protocol



Static web page



Dynamic web page



- A dynamic web page is a page that's generated by a server-side program or script.
- When a web server receives a request for a dynamic web page, it looks up the extension of the requested file to find out which application server should process the request
- When the application server receives a request, it runs the specified script.

Server-side scripting language

Language	Description
ASP.NET	Runs on a Microsoft IIS web server. It pages have the .aspx extensions
JSP	A free open-source language that is commonly used with Java servlets. It runs on an Apache web server, and its pages have the .jsp extension.
PHP	A free,open-source language that is typically used with an Apache web server. Its page have the .php extension.
Ruby	A free, open-source language that is typically combined with Rails framework to simplify development. Its pages have the .rb extension.
Perl	A free,open-source language that was orginally designed for use at the UNIX command line to manipulate text. Its pages have the .pl extension
Python	A free open-source language that can be used to develop many types of applications besides web applications. Its pages have the .py extension.

Client side script

- Javascript and jQuery
- Javascript is a client-side scripting language that is run by the Javascript engine of a web browser and controls the operation of the browser
- jQuery is a popular Javascript-based library that helps streamline web development.
- Their common use includes:
  - Data validation, date pickers, auto completion and diaglogs
  - Image swaps,image rollovers and slide shows
  - Drop-down menus, tabbed panels and accordions

What is a Web Page?

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
  - Looks (looked?) like:
    - A Microsoft Word document
- The markup tags provide information about the page content structure

Creating HTML Pages

- An HTML file must have an `.htm` or `.html` file extension
- HTML files can be created with text editors:
  - NotePad, NotePad ++, PSPad
- Or HTML editors (WYSIWYG Editors):
  - Microsoft FrontPage
  - Macromedia Dreamweaver
  - Netscape Composer
  - Microsoft Word
  - Visual Studio

HTML Structure

- HTML is comprised of “elements” and “tags”
  - Begins with <html> and ends with </html>
- Elements (tags) are nested one inside another:

```
<html> <head></head> <body></body> </html>
```

- Tags have attributes:

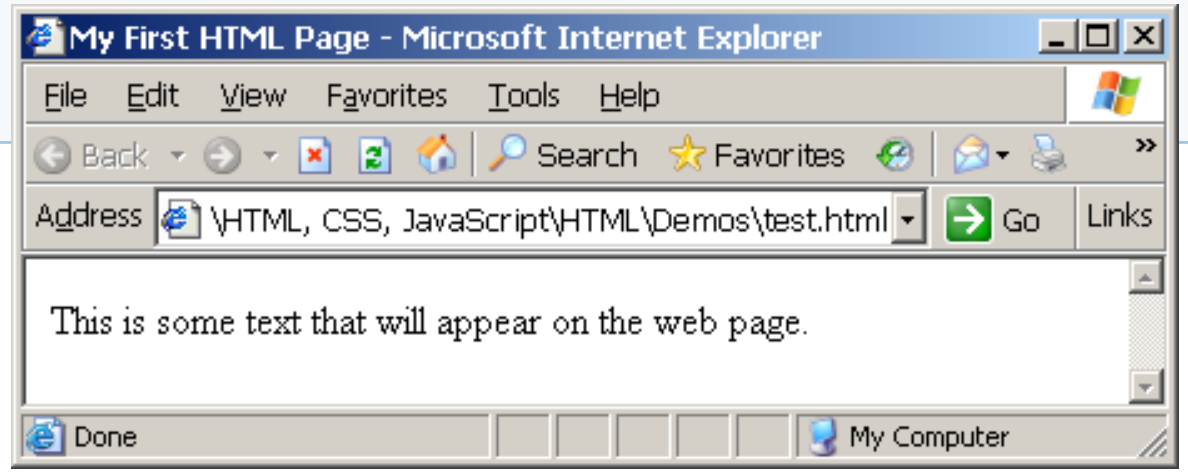
```
<img src="logo.jpg" alt="logo" />
```

- HTML describes structure using two main sections: <head> and <body>

First HTML Page

test.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```



Some Simple Tags

- Hyperlink Tags

```
<a href="http://www.telerik.com/"
  title="Telerik">Link to Telerik Web site</a>
```

- Image Tags

```
<img src="logo.gif" alt="logo" />
```

- Text formatting tags

```
This text is <em>emphasized.</em>
<br />new line<br />
This one is <strong>more emphasized.</strong>
```

Some Simple Tags – Example

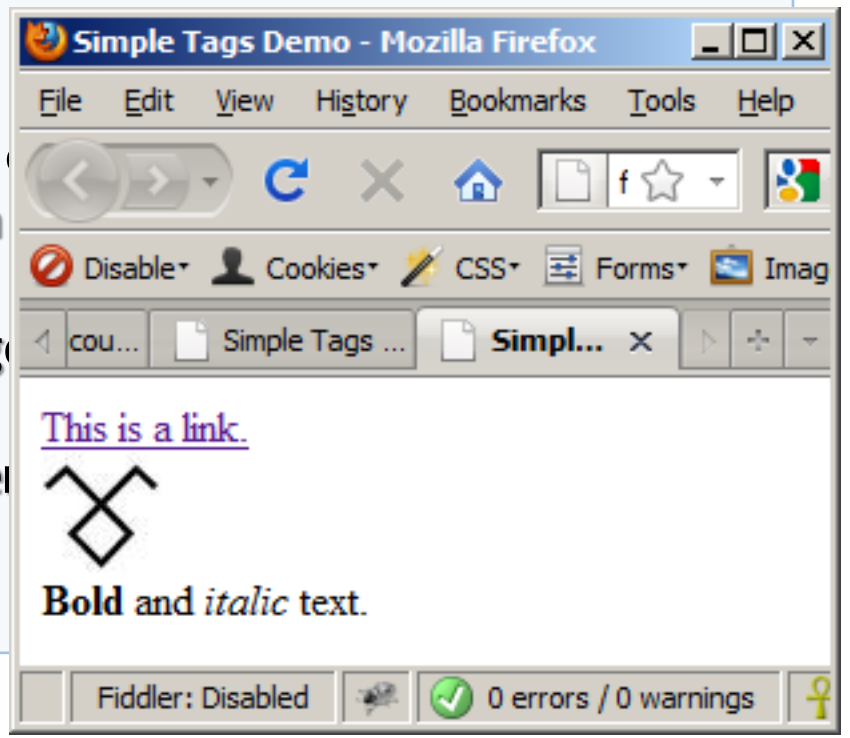
some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.com/" title=
    "Telerik site">This is a link.</a>
<br />
<img src="logo.gif" alt="logo" />
<br />
<strong>Bold</strong> and <em>italic</em> text.
</body>
</html>
```

Some Simple Tags – Example (2)

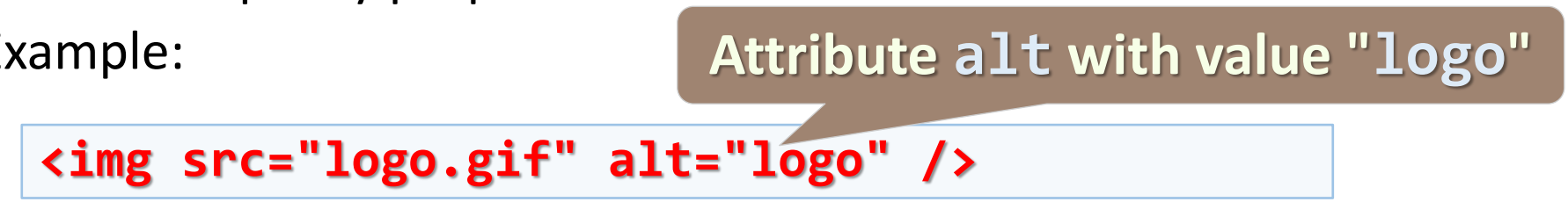
some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.
    "Telerik site">This is a
<br />
<img src="logo.gif" alt="log
<br />
<strong>Bold</strong> and <e
</body>
</html>
```



Tags Attributes

- Tags can have attributes
  - Attributes specify properties and behavior
  - Example:



  - Few attributes can apply to every element:
    - `id`, `style`, `class`, `title`
    - The `id` is unique in the document
    - Content of `title` attribute is displayed as hint when the element is hovered with the mouse
    - Some elements have obligatory attributes

Headings and Paragraphs

- Heading Tags (h1 – h6)

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- Paragraph Tags

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- Sections: `div` and `span`

```
<div style="background: skyblue;">
   This is a div</div>
```

Headings and Paragraphs – Example

headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title></head>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```

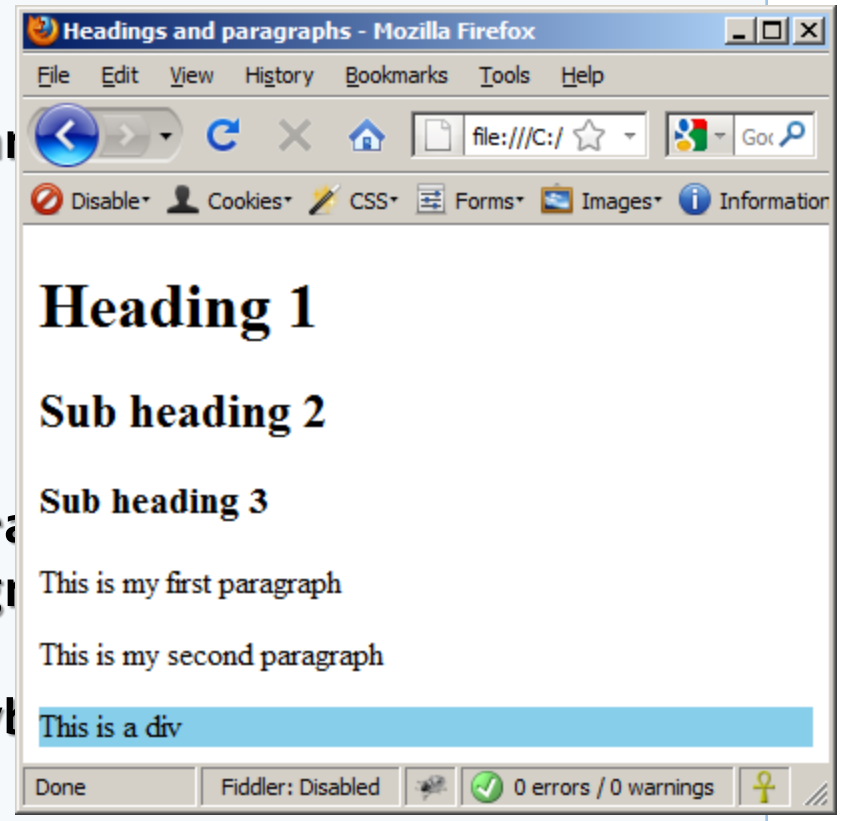
Headings and Paragraphs – Example (2)

headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and pa
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragra
    <p>This is my second paragr

    <div style="background:skyb
      This is a div</div>
  </body>
</html>
```



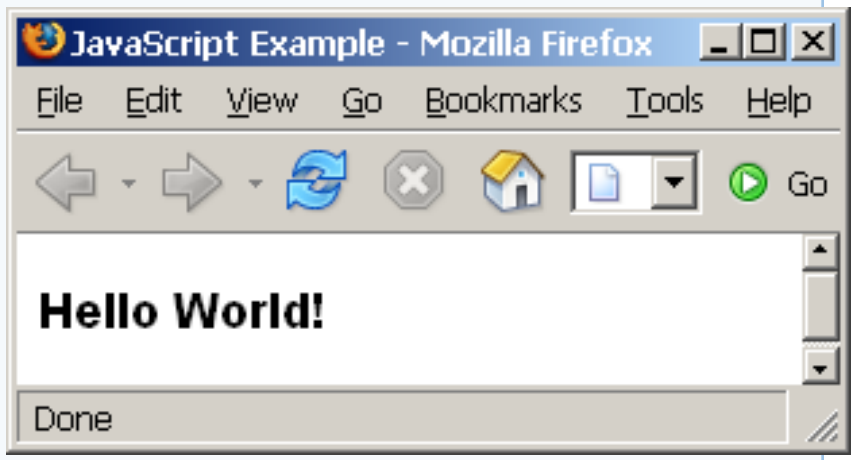
<head> Section: <script>

- The `<script>` element is used to embed scripts into an HTML document
  - Script are executed in the client's Web browser
  - Scripts can live in the `<head>` and in the `<body>` sections
- Supported client-side scripting languages:
  - JavaScript (it is not Java!)
  - VBScript
  - JScript

The <script> Tag – Example

scripts-example.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>JavaScript Example</title>
    <script type="text/javascript">
      function sayHello() {
        document.write("<p>Hello World!<\/p>");
      }
    </script>
  </head>
  <body>
    <script type=
      "text/javascript">
      sayHello();
    </script>
  </body>
</html>
```

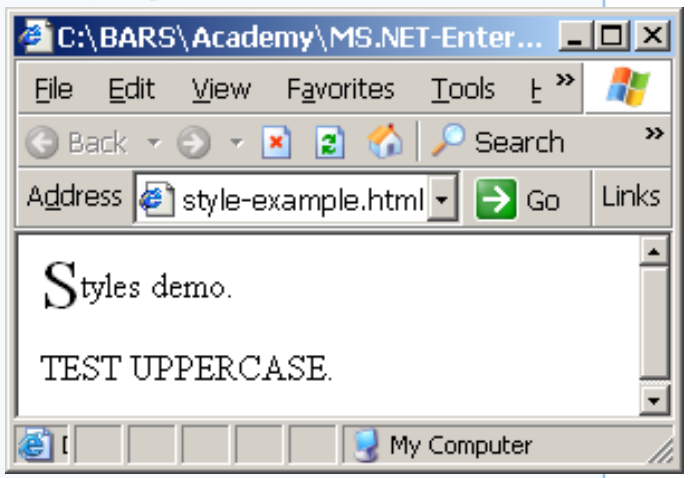


<head> Section: <style>

- The `<style>` element embeds formatting information (CSS styles) into an HTML page

style-example.html

```
<html>
  <head>
    <style type="text/css">
      p { font-size: 12pt; line-height: 12pt; }
      p:first-letter { font-size: 200%; }
      span { text-transform: uppercase; }
    </style>
  </head>
  <body>
    <p>Styles demo.<br />
       <span>Test uppercase</span>.
    </p>
  </body>
</html>
```



Comments: <!-- --> Tag

- Comments can exist anywhere between the `<html></html>` tags
- Comments start with `<!--` and end with `-->`

```
<!-- Telerik Logo (a JPG file) -->
<img src="logo.jpg" alt="Telerik Logo">
<!-- Hyperlink to the web site -->
<a href="http://telerik.com/">Telerik</a>
<!-- Show the news table -->
<table class="newstable">
...
```

<body> Section: Introduction

- The <body> section describes the viewable portion of the page
- Starts after the <head> </head> section
- Begins with <body> and ends with </body>

```
<html>
  <head><title>Test page</title></head>
  <body>
    <!-- This is the Web page body -->
  </body>
</html>
```

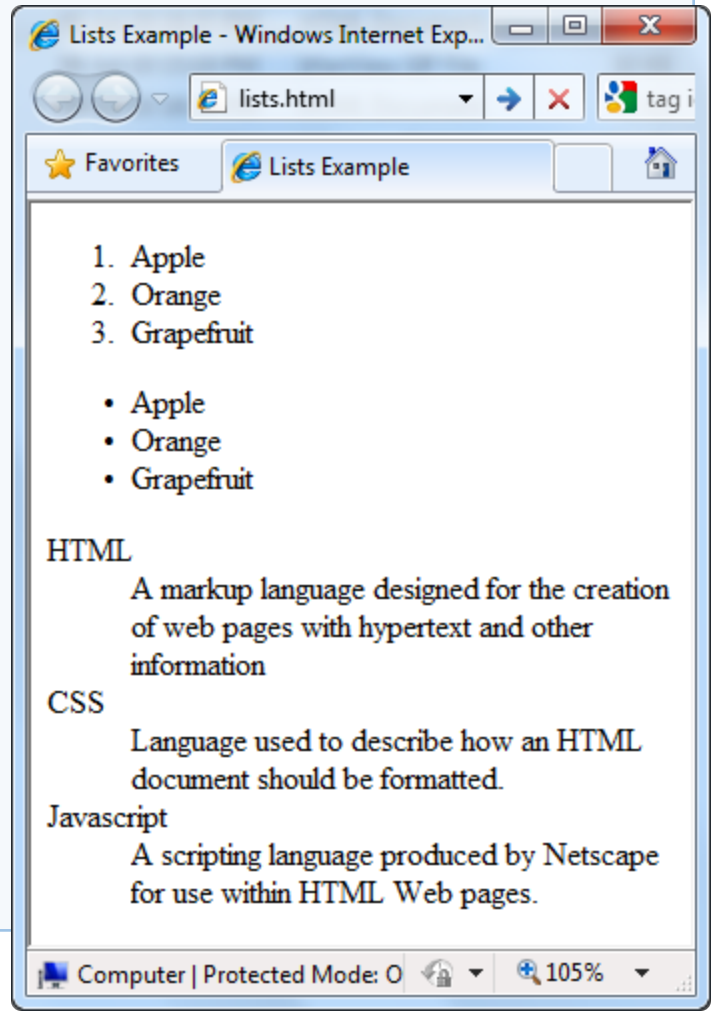
Lists – Example

```
<ol type="1">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ol>

<ul type="disc">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ul>

<dl>
    <dt>HTML</dt>
    <dd>A markup lang…</dd>
</dl>
```

lists.html



HTML Tables

- Tables represent tabular data
  - A table consists of one or several rows
  - Each row has one or more columns
- Tables comprised of several core tags: `<table></table>`: begin / end the table

  `<tr></tr>`: create a table row

  `<td></td>`: create tabular data (cell)
- Tables should not be used for layout. Use CSS floats and positioning styles instead

HTML Tables (2)

- Start and end of a table

```
<table> ... </table>
```

- Start and end of a row

```
<tr> ... </tr>
```

- Start and end of a cell in a row

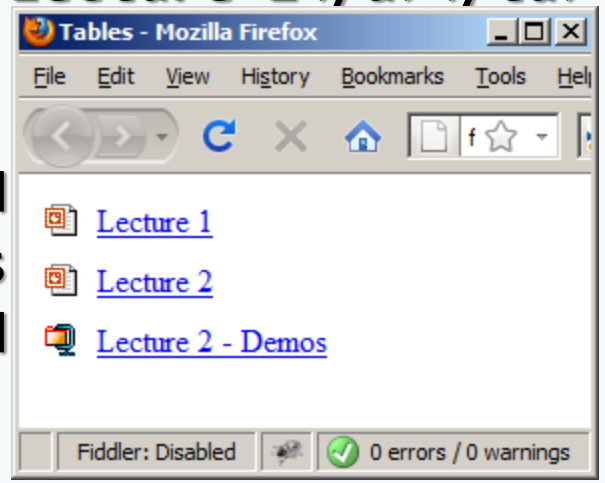
```
<td> ... </td>
```

Simple HTML Tables – Example

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td><img src="zip.gif"></td>
    <td><a href="lecture2-demos.zip">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```

Simple HTML Tables – Example (2)

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td><img src="zip.gif"></td
    <td><a href="lecture2-demos
      Lecture 2 - Demos</a></td
  </tr>
</table>
```

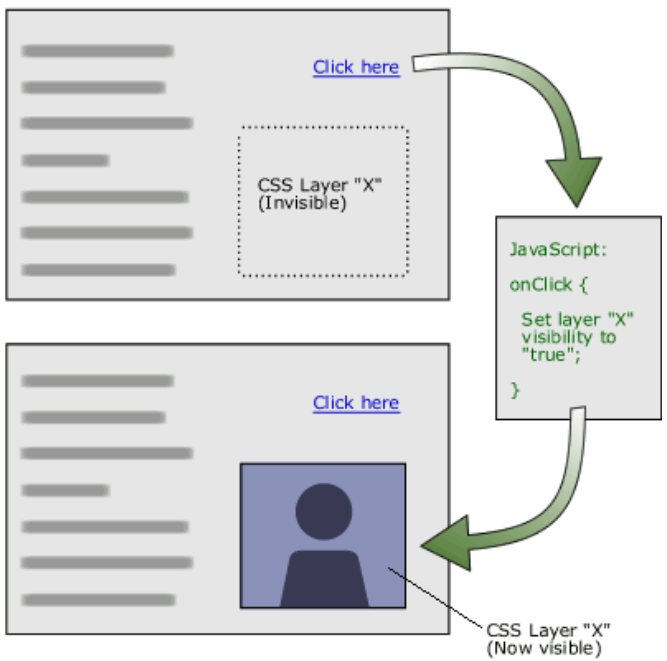


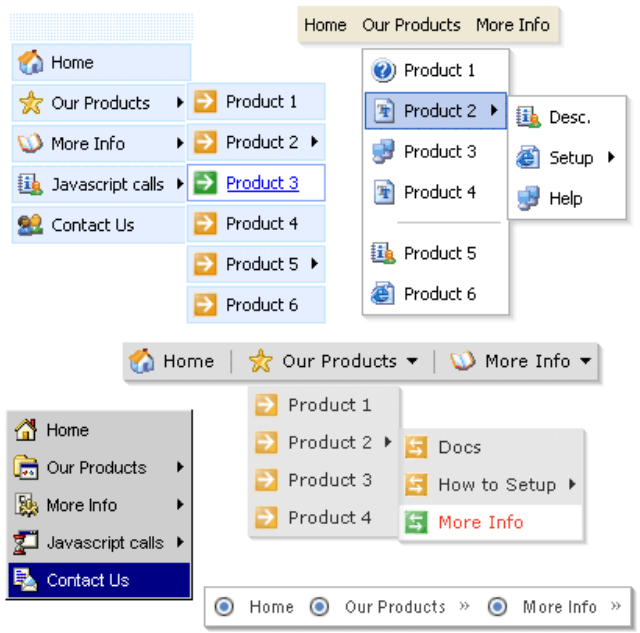
Basic of Javascript

- Introduction to JavaScript
  - What is JavaScript
  - Implementing JavaScript into Web pages
    - In <head> part
    - In <body> part
    - In external .js file

DHTML

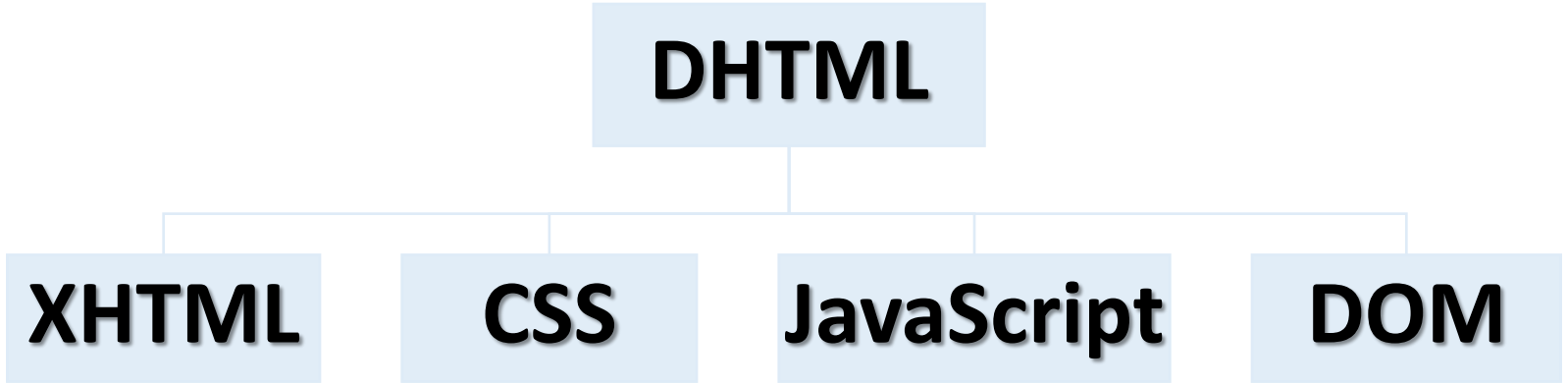
Dynamic Behavior at the Client Side





What is DHTML?

- Dynamic HTML (DHTML)
  - Makes possible a Web page to react and change in response to the user's actions
- DHTML = HTML + CSS + JavaScript



DTHML = HTML + CSS + JavaScript

- HTML defines Web sites content through semantic tags (headings, paragraphs, lists, …)
- CSS defines 'rules' or 'styles' for presenting every aspect of an HTML document
  - Font (family, size, color, weight, etc.)
  - Background (color, image, position, repeat)
  - Position and layout (of any object on the page)
- JavaScript defines dynamic behavior
  - Programming logic for interaction with the user, to handle events, etc.

JavaScript

Dynamic Behavior in a Web Page

JavaScript

- JavaScript is a front-end scripting language developed by Netscape for dynamic content
  - Lightweight, but with limited capabilities
  - Can be used as object-oriented language
- Client-side technology
  - Embedded in your HTML page
  - Interpreted by the Web browser
- Simple and flexible
- Powerful to manipulate the DOM

JavaScript Advantages

- JavaScript allows interactivity such as:
  - Implementing form validation
  - React to user actions, e.g. handle keys
  - Changing an image on moving mouse over it
  - Sections of a page appearing and disappearing
  - Content loading and changing dynamically
  - Performing complex calculations
  - Custom HTML controls, e.g. scrollable table
  - Implementing AJAX functionality

What Can JavaScript Do?

- Can handle events
- Can read and write HTML elements and modify the DOM tree
- Can validate form data
- Can access / modify browser cookies
- Can detect the user's browser and OS
- Can be used as object-oriented language
- Can handle exceptions
- Can perform asynchronous server calls (AJAX)

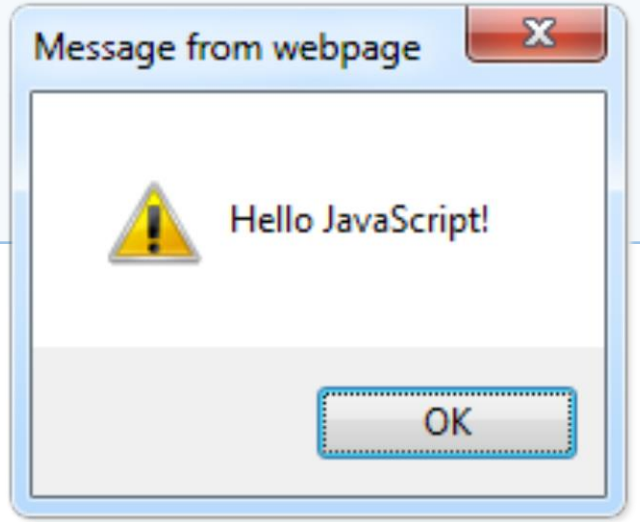
The First Script

first-script.html

```
<html>

<body>
  <script type="text/javascript">
     alert('Hello JavaScript!');
  </script>
</body>

</html>
```



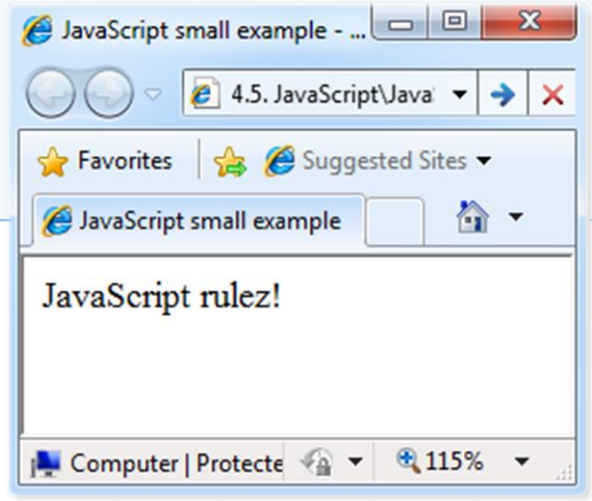
Another Small Example

small-example.html

```
<html>

<body>
  <script type="text/javascript">
     document.write('JavaScript rulez!');
  </script>
</body>

</html>
```



Using JavaScript Code

- The JavaScript code can be placed in:
  - `<script>` tag in the head
  - `<script>` tag in the body – not recommended
  - External files, linked via `<script>` tag the head
    - Files usually have `.js` extension

```
<script src="scripts.js" type="text/javscript">
<!- code placed here will not be executed! -->
</script>
```

    - Highly recommended
    - The `.js` files get cached by the browser

JavaScript – When is Executed?

- JavaScript code is executed during the page loading or when the browser fires an event
  - All statements are executed at page loading
  - Some statements just define functions that can be called later
- Function calls or code can be attached as "event handlers" via tag attributes
  - Executed when the event is fired by the browser

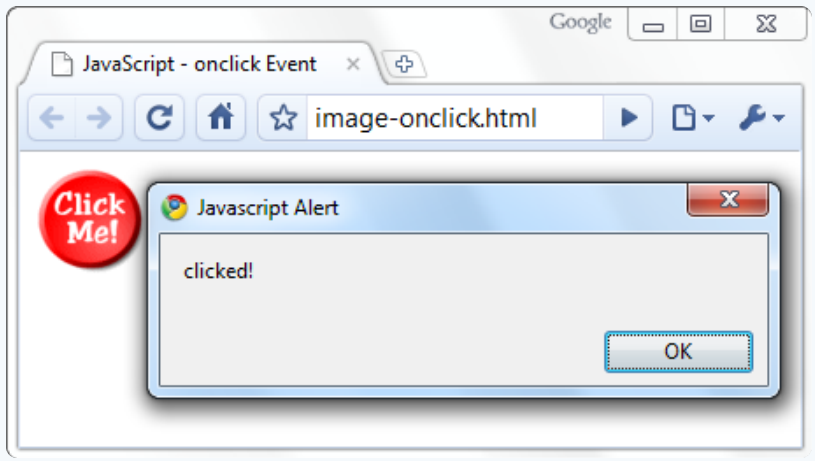
```
<img src="logo.gif" onclick="alert('clicked!')" />
```

Calling a JavaScript Function from Event Handler – Example

image-onclick.html

```
<html>
<head>
<script type="text/javascript">
  function test (message) {
    alert(message);
  }
</script>
</head>

<body>
  <img src="logo.gif"
    onclick="test('clicked!')" />
</body>
</html>
```



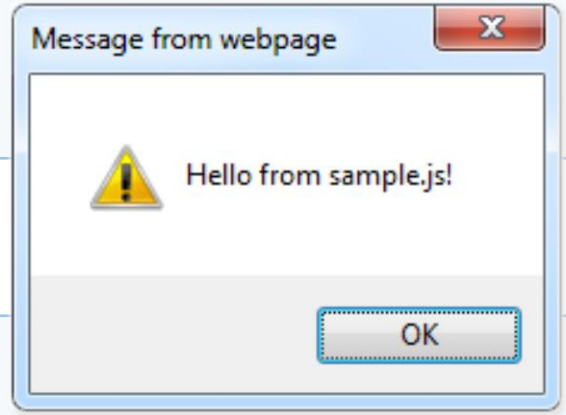
Using External Script Files

- Using external script files:

external-JavaScript.html

```
<html>
<head>
  <script src="sample.js" type="text/javascript">
  </script>
</head>
<body>
  <button onclick="sample()" value="Call JavaScript
    function from sample.js" />
</body>
</html>
```

The <script> tag is always empty.



- External JavaScript file:

```
function sample() {
  alert('Hello from sample.js!')
}
```

sample.js

Download SeletorGadget in chrome browser