Secure Programming
—— Introduction of the Web

胡天磊, Dr. HU Tianlei

Associate Professor

College of Computer Science, Zhejiang Univ.

htl@zju.edu.cn

Course Outline

- Introduction of HTTP
- Introduction of Front-End Web Language
- Introduction of Server Framework and Language
- Introduction of SQL & Database

Hyper Text Transfer Protocol

HTTP - Hyper Text Transfer Protocol

On top of TCP (default port: 80)

Two main versions

Version 1.0 defined in RFC 1945

Version 1.1 defined in RFC 2616

Client

Opens TCP connection to the server

Sends an HTTP request

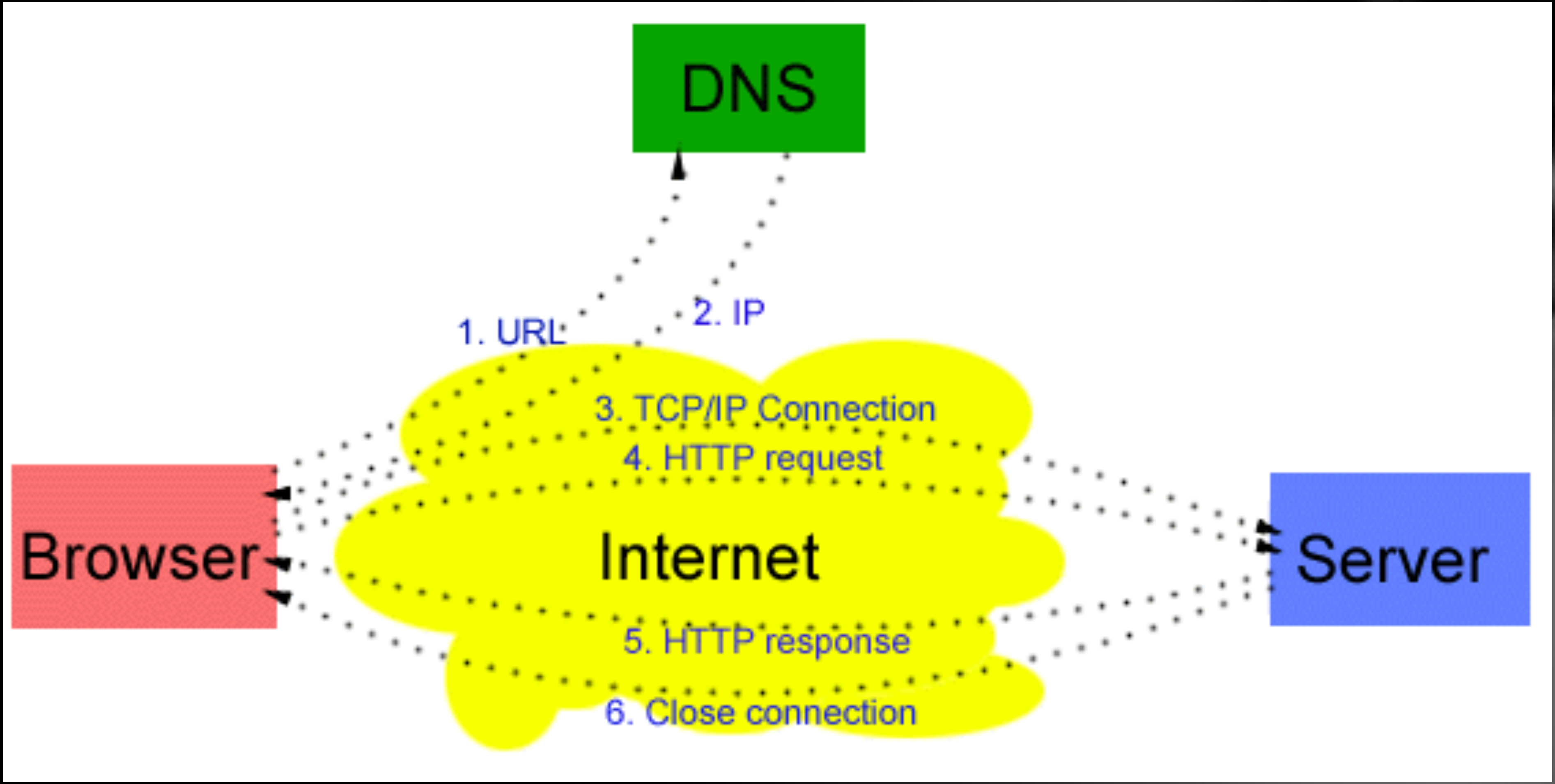
Server

Accepts the TCP connection from the client

Processes the HTTP request

Sends an HTTP reply

How the HTTP works



From one "click" to one Webpage

Short Intro to HTTP

Each request and response has three parts:

- the request or response line
- a header section
- the entity-body

The client initiates a transaction as follows:

- GET /index.html?param=value HTTP/1.0

After the request and headers, the client may send additional data

- This data is mostly used by CGI programs using the POST method
- Note that for the GET method, the parameters are encoded into the URL

HTTP Request

An HTTP request consists of the following pieces:

- The method must be one of a set of legal actions;
- The Universal Resource Identifier(URI) is the name of the information requested (Berners-Lee 1994);
  - Compare with URL, the Universal Resource Locator
- The protocol version;
- (optionally) Other information to modify or supplement the request.

HTTP Request

Example:

```
GET /stuff/Funny/silly.html HTTP/1.0
User-agent: NCSA Mosaic
Accept: text/plain
Accept: text/html
Accept: application/postscript
Accept: image/gif
```

- The first line is the method, the URI, and the version of HTTP;
- The second line identifies the type of browser/client (the User-agent) that is sending the request;
  - The User-agent may be a human browser or an automated agent, i.e.
    - Safari on iPad: Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405
    - Search engine bot by Google: Googlebot/2.1 (+http://www.google.com/bot.html)
- The following lines indicate the data types (in terms of MIME spec) that the client is prepared to receive.

HTTP Request

Method GET is used in the above example.

HTTP also has some other methods, which are listed as follows:

Method	Explain
GET	Return the object.
HEAD	Return only information about the object, not the object itself.
POST	Send information to be stored on the Server.
PUT	Send a new copy of an existing object to the server.
DELETE	Permanently delete the object.

HTTP Response

An HTTP response consists of the following piece:

- A status line, which indicates the success or failure of the request;
- A description of the data in the response. This is information about the data, which is called meta-information;
- Blank line;
- The actual information requested.

HTTP Response

Example:

```
HTTP1.0 Status 200 Document follows
Server: NCSA/1.4
Date: Tue,4 Jul,1996 19:04 GMT
Content-type: text/html
Content-length: 5280
Last-modified: Wed,1 Jan 1996 01:00:02 GMT

The contents of the document requested.
```

- 1st line indicates the version of HTTP that the server uses and the request succeeded, and the information will be returned.
- 2nd line indicates the HTTP server software ;
- 3rd line indicates the date and time of the request;
- 4th line indicates this is an HTML document, which is of MIME-type text/html;
- 5th line indicates the document is 5280 bytes long;
- 6th line indicates the last modified time of the document;
- The blank line is used as a separating line, indicating the end of meta-information.

URLs

Syntax: <scheme>://<authority><path>?<query>

Scheme: a string specifying the protocol/framework

Examples:

- **ftp**://ftp.ietf.org/rfc/rfc1808.txt
- **http**://www.iseclab.org/~dbalzarotti/
- **mailto**:nobody@iseclab.org
- **telnet**://127.0.0.1

URLs

Syntax: <scheme>://<authority><path>?<query>

Authority: a namespace that qualifies the resource

- Generally in the form: username@hostname:portnumber
- The hostname can be either a name or an IP address

Examples:

- **ftp://ftp.ietf.org/rfc/rfc1808.txt**
- **http://www.iseclab.org/~dbalzarotti/**
- **mailto:nobody@iseclab.org**
- **telnet://127.0.0.1**

URLs

Syntax: <scheme>://<authority><path>?<query>

Path: a / separated path of the requested resource

Examples:

- ftp://ftp.ietf.org/rfc/rfc1808.txt
- http://www.iseclab.org/~dbalzarotti/
- mailto:nobody@iseclab.org
- telnet://127.0.0.1

URLs

Syntax: <scheme>://<authority><path>?<query>

Query: application-specific piece of information

Examples:

- http://host/login.php?user=foo&pwd=bar

HTTPs

Encrypt the communication

- Protect against eavesdropping
- Protect from a man in the middle attacks (provided that the certificate is trusted)
- Used to protect the user authentication
- By-pass IDS / IPS

The trust in HTTPS is inherited from major certificate authorities that come pre-installed in browser software.

- Do not protect against the majority of attacks aimed at the web application.

Web Languages

——HTML/CSS/JS

Web Languages

Three languages all web developers MUST learn:

- HTML to define the content of web pages
- CSS to specify the layout of web pages
- JavaScript to determine the behavior of web pages

What is HTML?

HTML is a language for describing web pages.

- HTML stands for HyperText Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

HTML Example

```
<!DOCTYPE html>
< html>
< body>

< h1>My First Heading</h1>

< p>My first paragraph.</p>

< /body>
< /html>
```

Example Explained

- The DOCTYPE declaration defines the document type
- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between <p> and </p> is displayed as a paragraph

HTML Tags

HTML markup tags are usually called HTML tags.

- HTML tags are keywords (tag names) surrounded by angle brackets like <html>
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, and the second tag is the end tag
- The end tag is written like the start tag, with a forward slash before the tag name
- Start and end tags are also called opening tags and closing tags

<tagname>content</tagname>

HTML Elements

"HTML tags" and "HTML elements" often describe the same thing.

- But strictly speaking, an HTML element is everything between the start and end tags, including the tags.

<p>This is a paragraph.</p>

HTML Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading.

```
<h1>This is a heading.</h1>
<h2>This is a heading.</h2>
<h3>This is a heading.</h3>
```

Headings Are Important

- Use HTML headings for headings only. Don't use headings to make text BIG or bold.
- Search engines use your headings to index the structure and content of your web pages.
- Since users may skim your pages by their headings, it is important to use headings to show the document structure.
- H1 headings should be used as the main headings, followed by H2 headings, then the less important H3 headings, and so on.

HTML Paragraphs

Paragraphs are defined with the <p> tag.

```
<p>This is a paragraph.</p>
<p>This is another paheading.</p>
```

Use the <br> tag if you want a line break (a new line) without starting a new paragraph.

```
< p>This is<br>a para<br>graph with line
breaks.</p>
```

HTML Text Formatting

This text is bold

This text is italic

This is $_{\text{subscript}}$ and $^{\text{superscript}}$

HTML Text Formatting

Tag	Description
<b>	Defines bold text
<em>	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
<strong>	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML Comments

Comment tags <!-- and --> are used to insert comments in HTML.

<!-- Write your comments here -->

The browser does not display comments, but comments can help document your HTML. With comments, you can place notifications and reminders in your HTML.

HTML Links

The HTML <a> tag defines a hyperlink.

- A hyperlink (or link) is a word, group of words, or image you can click to jump to another document.
- When you move the cursor over a link on a Web page, the arrow will turn into a little hand.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

```
<a href="url">Link text</a>
```

HTML Head

The <head> element is a container for all the head elements.

- The following tags can be added to the head section: <title>, <style>, <meta>, <link>, <script>, <noscript>, and <base>.

```
<head>
   < title>Title of the document</title>
   <base href="http://www.w3schools.com/images/"
      target="_blank">
   <link rel="stylesheet" type="text/css" href="mystyle.css">
   <style type="text/css">
       body {background-color:yellow;}
       p {color:blue;}
    </style>
< /head>
```

HTML Head Related Tags

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script
<style>	Defines style information for a document

HTML Tag Reference

More Tags

There are also other tags, like CSS, Images, Tables, Lists, Forms, Blocks, Layout, and so on.

More Info

CSS Tutorial: http://www.w3schools.com/css/default.asp

HTML DOM

What's DOM

DOM is a W3C (World Wide Web Consortium) standard.

- Short for "Document Object Model."
- "The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The W3C DOM standard is separated into three different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

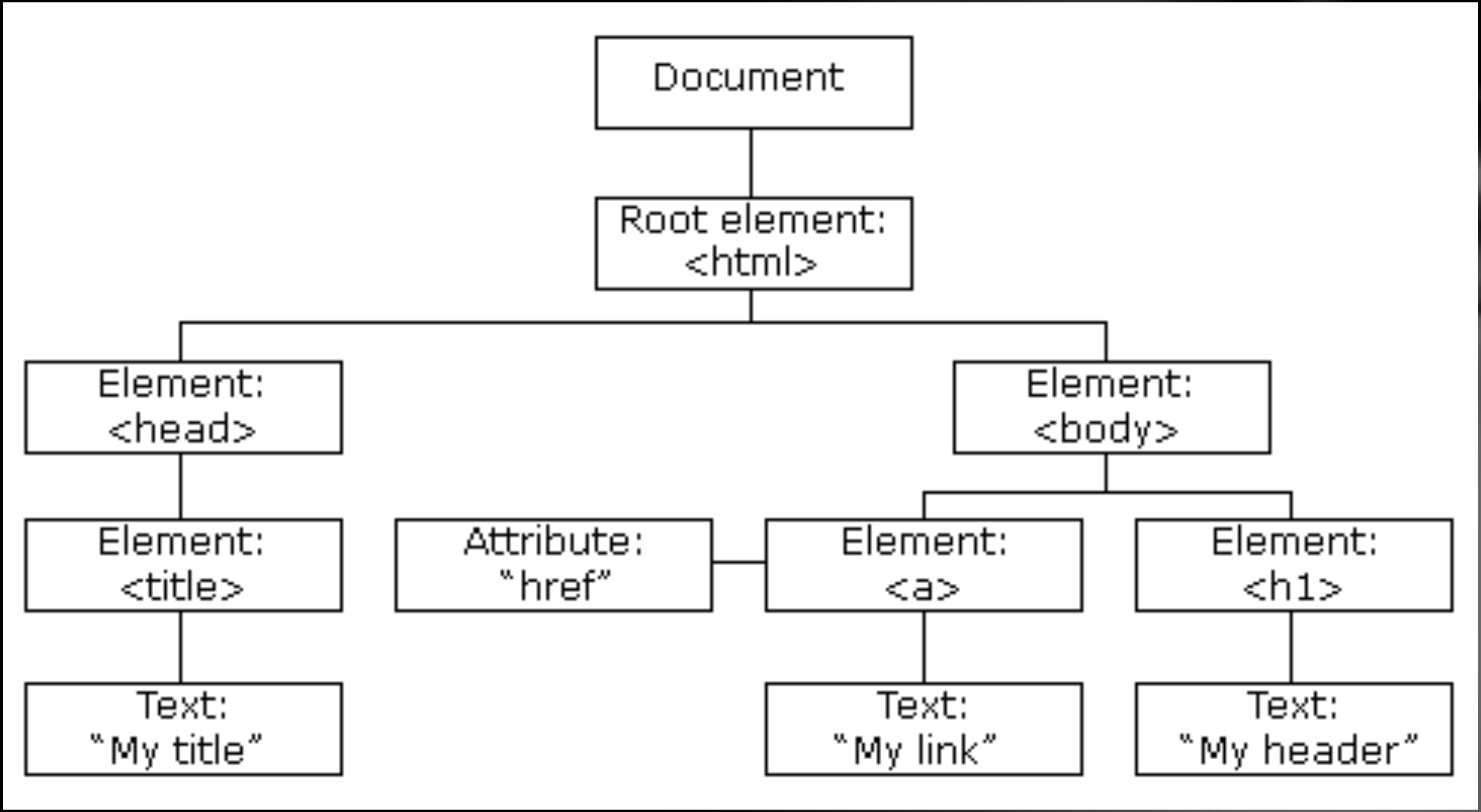
What's HTML DOM

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

HTML DOM


Document
Root element: <html>
Element: <head>
Element: <body>
Element: <title>
Attribute: "href"
Element: <a>
Element: <h1>
Text: "My title"
Text: "My link"
Text: "My header"

Web Languages

——CSS

CSS

What's CSS

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements

CSS can be added to HTML in the following ways:

- Inline - using the style **attribute** in HTML elements
- Internal - using the <style> **element** in the <head> section
- External - using an external CSS **file**

The preferred way to add CSS to HTML is to:

Put CSS syntax in separate CSS files.

CSS Solves a Big Problem

HTML was never intended to contain tags for formatting a document.

- HTML was intended to define the content of a document, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

- When tags like <font> and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers.
  - The development of large websites, where fonts and color information were added to every page, became a long and expensive process.
  - The World Wide Web Consortium (W3C) created CSS to solve this problem.
- In HTML 4.0, all formatting could be removed from the HTML document and stored in a separate CSS file.

All browsers support CSS today.

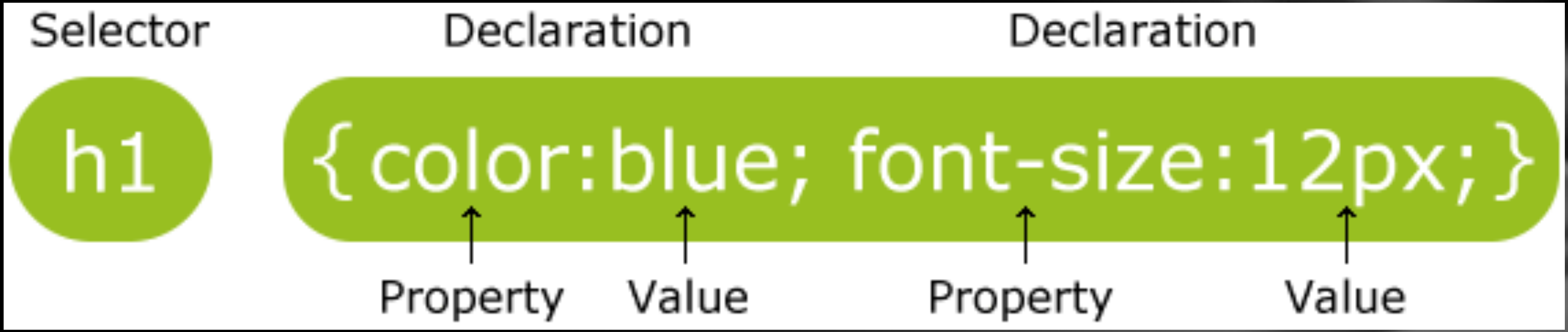
CSS Saves a Lot of Work

CSS defines HOW HTML elements are to be displayed.

- Styles are normally saved in external .css files.
- External style sheets enable you to change the appearance and layout of all the pages on a Web site just by editing one single file!

CSS Syntax

- A CSS rule set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

CSS Example

- A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:

```
p {color: red; text-align: center;}
```

- To make the CSS more readable, you can put one declaration on each line like this:

```
p
{
color: red;
text-align: center;
}
```

CSS Example

- A CSS comment starts with /* and ends with */:

```
/*This is a multiple
lines comment*/
p
{
color: red;
/*This is another comment*/
text-align: center;
}
```

How To …

When a browser reads a style sheet, it will format the document according to it.

Three Ways to Insert CSS

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

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

```
hr {color: sienna;}
p {margin-left:20px;}
body { background-image:
        url("images/background.gif");}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:

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

Inline Styles

An inline style loses many advantages of style sheets by mixing content with presentation. Use this method sparingly!

- To use inline styles, you use the style attribute in the relevant tag.
- The style attribute can contain any CSS property.
- The example shows how to change the color and the left margin of a paragraph:

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

Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the h3 selector:

```
h3
{
   color: red;
   text-align: left;
   font-size:8pt;
}
```

Multiple Style Sheets

And an internal style sheet has these properties for the h3 selector:

```
h3
{
   text-align: right;
   font-size: 20pt;
}
```

If the page with the internal style sheet also links to the external style sheet, the properties for h3 will be:

```
color: red;
text-align: right;
font-size: 20pt;
```

The color is inherited from the external style sheet, and the internal style sheet replaces the text-align and the font-size.

Cascading order

What style will be used when more than one style is specified for an HTML element?

Generally speaking, we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number 4 has the highest priority:

1. *Browser default*
2. *External style sheet*
3. *Internal style sheet (in the head section)*
4. *Inline style (inside an HTML element)*

Web Languages ——JavaScript

JavaScript

- Developed by Netscape as a light-weight scripting language with object-oriented capabilities
  - The current version is standardized as ECMA 357
  - Most popular scripting language on the Internet
  - Works with basically all browsers
- Designed to add interactivity to HTML pages
  - usually embedded directly into HTML pages (<script>tags)
  - can access and add elements to HTML pages (DOM tree)
  - can react to events
- JavaScript is a scripting language
  - dynamic, weak typing
  - interpreted language
  - The script executes on a virtual machine in the browser

JavaScript

- JavaScript is quite different from Java
  - Originally, JavaScript was named LiveScript
    - marketing department made developers change the name
  - Java is more complex and powerful
  - static, strong typing

- Design decisions (Brendan Eich)
  - make it easy to copy and paste snippets of code
  - tolerate “minor” errors (missing semicolons)
  - simplified event handling
  - choose some powerful, often-needed primitives

JavaScript

- Syntax quite similar to Java
  - control statements, exception handling
- No classes, but object-based
  - uses objects with properties (name-value pairs)
- No input/output facilities per se
  - must be provided by embedding the environment
- The scope of variables is either global or function-local
- Code can be generated at run-time and executed on-the-fly
  - eval() function

Where To Write Code

In HTML, JavaScripts must be inserted between <script> and </script> tags, where they can be put in the <body> and the <head> sections of an HTML page.

Example

```
<script>
function myFunction()
{
   document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
```

The browser will interpret the code between the <script> and </script> tags as JavaScript.

Concepts in Javascript

JavaScript Functions and Events

- Most often, JavaScript code is written to be executed when an event occurs, like when the user clicks a button.
- If we put JavaScript code inside a function, we can call that function when an event occurs.

JavaScript in <head> or <body>

- You can place an unlimited number of scripts in an HTML document.
- Scripts can be in the <body> or the <head> section of HTML or in both.
- It is a common practice to put functions in the <head> section or at the bottom of the page.
- Separating HTML and JavaScript by putting all the code in one place is always a good habit.

JavaScript in <head>

Example

```
<!DOCTYPE html>
<html><head>
<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
</head>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

JavaScript in <body>

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>

</body>
</html>
```

External JavaScript

Scripts can also be placed in external files. External files often contain code to be used by several different web pages.

- External JavaScript files have the file extension .js.
- To use an external script, put the name of the script file in the source (src) attribute of the <script> tag.

```
<!DOCTYPE html>
<html>
<body>
<script src="myScript.js"></script>
</body>
</html>
```

JavaScript Output

- JavaScript does not have any print or output functions.
- In HTML, JavaScript can only be used to manipulate HTML elements.
- To use src) attrib

Mani

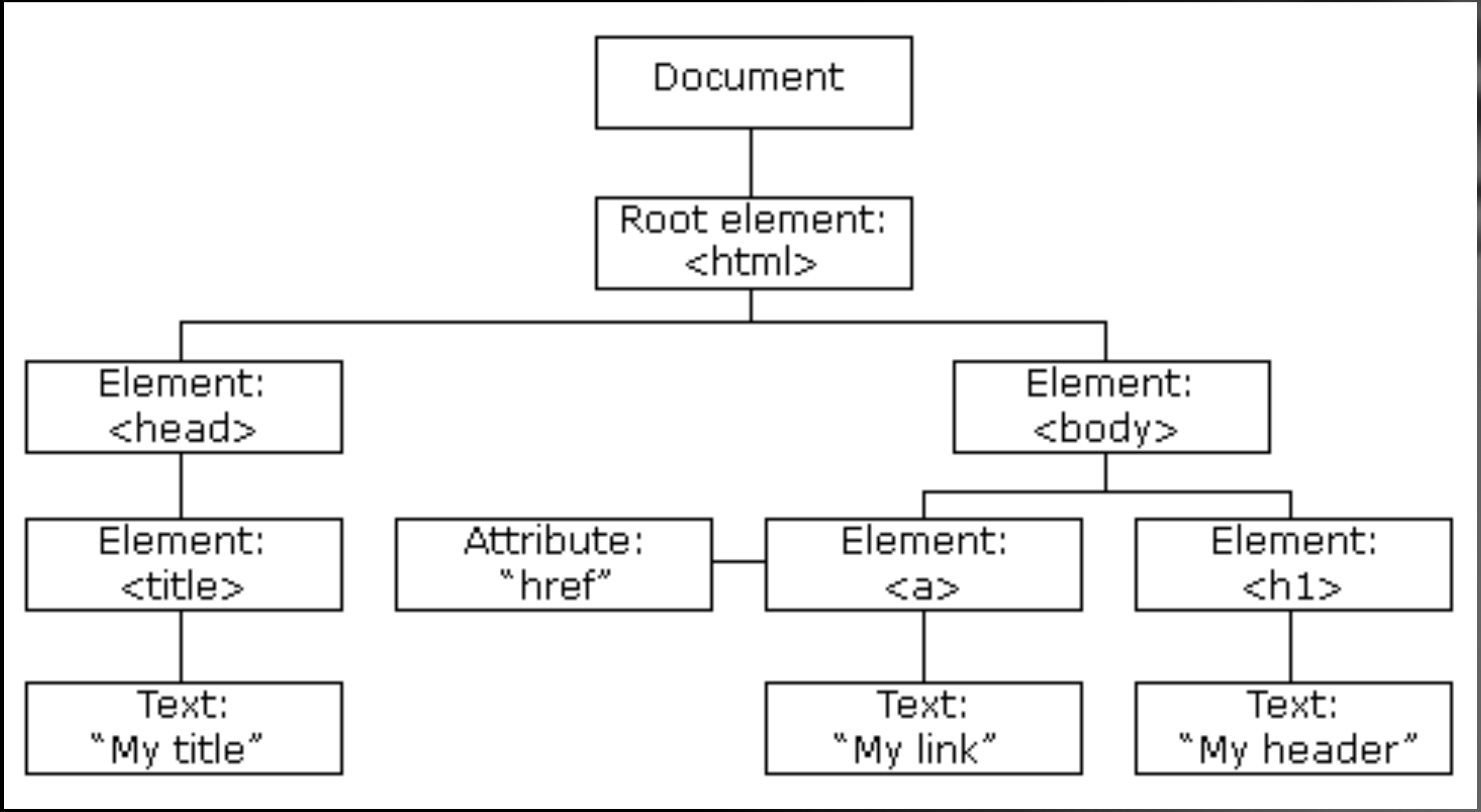
```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<p id="demo">My First Paragraph</p>

<script>
elem = document.getElementById("demo");
elem.innerHTML = "My First JavaScript";
</script>

</body>
</html>
```

HTML DOM

When a web page is loaded, the browser creates a Document Object Model of the page. The HTML DOM model is constructed as a tree of Objects:



What the JavaScript can DO

JavaScript can change all the HTML elements on the page

- JavaScript can change all the HTML attributes on the page
- JavaScript can change all the CSS styles on the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events on the page
- JavaScript can create new HTML events on the page

The DOM Programming Interface

The HTML DOM can be accessed with JavaScript (and other programming languages).

- In the DOM, all HTML elements are defined as objects.
- The programming interface is the properties and methods of each object.
  - A property is a value you can get or set (like changing the content of an HTML element).
  - A method is an action you can do (like adding or deleting an HTML element).

Example

```
<html>
<body>
<p id="intro">Hello World!</p>

<script>
   var txt=document.getElementById("intro").innerHTML;
   document.write(txt);
</script>

</body>
</html>
```

In the example above, *getElementById()* is a method, while *innerHTML* is a property.

The DOM Programming Interface

The getElementById() method

- The most common way to access an HTML element is to use the id of the element.
- In the example above, the getElementById() method used id="intro" to find the element.

The innerHTML property

- The easiest way to get the content of an element is by using the innerHTML property.
- The innerHTML property is useful for getting or replacing the content of HTML elements.

The Methods

Finding HTML Elements

Method	Description
document.getElementById()	Finding an element by element id
document.getElementsByTagName()	Finding elements by tag name
document.getElementsByClassName()	Finding elements by class name
document.forms[]	Finding elements by HTML element objects

The Methods

Changing HTML Elements

Method	Description
element.innerHTML=	Changing the inner HTML of an element
element.*attribute=*	Changing the attribute of an HTML element
element.setAttribute*(attribute,value)*	Changing the attribute of an HTML element
element.style.*property=*	Changing the style of an HTML element

The Methods

Adding and Deleting Elements

Method	Description
document.createElement()	Create an HTML element
document.removeChild()	Remove an HTML element
document.appendChild()	Add an HTML element
document.replaceChild()	replace an HTML element
document.write(*text*)	Writing into the HTML output stream

Adding Events Handlers

Method	Description
document.getElementById(*id*).onclick=function(){*code*}	Adding event handler code to an onclick event

HTML DOM Navigation

DOM Nodes

According to the W3C HTML DOM standard, everything in an HTML document is a node:

- The entire document is a document node
- Every HTML element is an element node
- The text inside HTML elements are text nodes
- Every HTML attribute is an attribute node
- All comments are comment nodes

With the HTML DOM, all nodes in the node tree can be accessed by JavaScript. New nodes can be created, and all nodes can be modified or deleted.

HTML DOM Navigation

N

Th
ot

Th

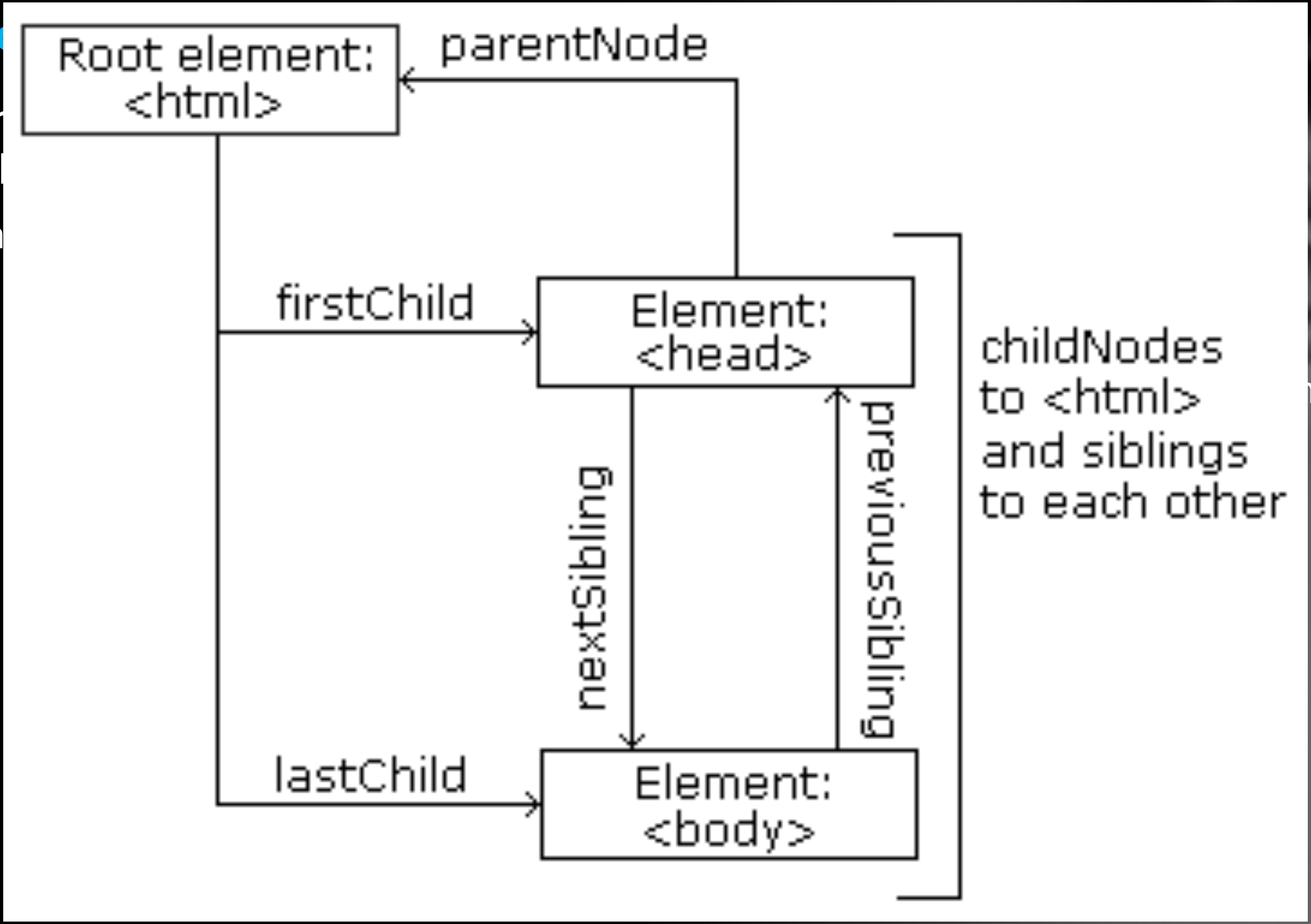
•

•

•

•

nt)



HTML DOM Navigation

Example

```
<html>

  <head>
   <title>DOM Tutorial</title>
  </head>

  <body>
   <h1>DOM Lesson one</h1>
   <p>Hello world!</p>
  </body>

</html>
```

HTML DOM Navigation

Example

From the HTML above you can read:

- <html> is the root node
- <html> has no parents
- <html> is the parent of <head> and <body>
- <head> is the first child of <html>
- <body> is the last child of <html>

and:

- <head> has one child: <title>
- <title> has one child (a text node): "DOM Tutorial"
- <body> has two children: <h1> and <p>
- <h1> has one child: "DOM Lesson one"
- <p> has one child: "Hello world!"
- <h1> and <p> are siblings

HTML DOM Navigation

Navigating Between Nodes

You can use the following node properties to navigate between nodes with JavaScript:

- parentNode
- childNodes[nodenumber]
- firstChild
- lastChild
- nextSibling
- previousSibling

HTML DOM Navigation

Child Nodes and Node Values

In addition to the innerHTML property, you can also use the childNodes and nodeValue properties to get the content of an element.

The following code gets the value of the <p> element with id="intro":

```
<html>
<body>
<p id="intro">Hello World!</p>
<script>
var
txt=document.getElementById("intro").childNodes[0].node
Value;
document.write(txt);
</script>
</body>
</html>
```

HTML DOM Navigation

DOM Root Nodes

There are two special properties that allow the access to the full document:

- document.documentElement - The full document
- document.body - The body of the document

```
<html>
<body>
<p>Hello World!</p>
<div>
<p>The DOM is very useful!</p>
<p>This example demonstrates the <b>document.body</b> property.</p>
</div>
<script>
alert(document.body.innerHTML);
</script>
</body>
</html>
```

HTML DOM Navigation

The nodeName Property

The nodeName property specifies the name of a node.

- nodeName is read-only
- nodeName of an element node is the same as the tag name
- nodeName of an attribute node is the attribute name
- nodeName of a text node is always #text
- nodeName of the document node is always #document

Note: nodeName always contains the uppercase tag name of an HTML element.

HTML DOM Navigation

The nodeValue Property

The nodeValue property specifies the value of a node.

- nodeValue for element nodes is undefined
- nodeValue for text nodes in the text itself
- nodeValue for attribute nodes is the attribute value

HTML DOM Navigation

The nodeType Property

The nodeType property returns the type of node. nodeType is read-only. The most important node types are:

Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

HTML DOM Elements (Nodes)

Creating New HTML Elements (Nodes)

To add a new element to the HTML DOM, you must first create the element (element node) and then append it to an existing element.

```
<div id="div1">
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
</div>

<script>
var para=document.createElement("p");
var node=document.createTextNode("This is new.");
para.appendChild(node);
var element=document.getElementById("div1");
element.appendChild(para);
</script>
```

HTML DOM Elements (Nodes)

Creating new HTML Elements - insertBefore()

The appendChild() method in the previous example appended the new element as the last child of the parent.

If you don't want that, you can use the insertBefore() method:

```
<div id="div1">
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
</div>
<script>
var para=document.createElement("p");
var node=document.createTextNode("This is new.");
para.appendChild(node);
var element=document.getElementById("div1");
var child=document.getElementById("p1");
element.insertBefore(para,child);
</script>
```

HTML DOM Elements (Nodes)

Removing Existing HTML Elements

To remove an HTML element, you must know the parent of the element.

```
<div id="div1">
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
</div>

<script>
var parent=document.getElementById("div1");
var child=document.getElementById("p1");
parent.removeChild(child);
</script>
```

HTML DOM Elements (Nodes)

Replacing HTML Elements

To replace an element to the HTML DOM, use the replaceChild() method.

```
<div id="div1">
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
</div>

<script>
var para=document.createElement("p");
var node=document.createTextNode("This is new.");
para.appendChild(node);

var parent=document.getElementById("div1");
var child=document.getElementById("p1");
parent.replaceChild(para,child);
</script>
```

HTML DOM Nodelist

A nodelist is an array of nodes (like an array of all HTML elements).

The getElementsByTagName() method returns a node list. A node list is an array of nodes.

The following code selects all <p> nodes in a document:

```
var x=document.getElementsByTagName("p");
```

The nodes can be accessed by index number. To access the second <p>, you can write:

```
y=x[1];
```

HTML DOM Nodelist

HTML DOM Node List Length

The length property defines the number of nodes in a node list.

You can loop through a node list by using the length property:

```
x=document.getElementsByTagName("p");

for (i=0;i<x.length;i++)
{
document.write(x[i].innerHTML);
document.write("<br />");
}
```

More Info

More Info about JavaScript: http://www.w3schools.com/js/default.asp

Introduction to JS Frameworks: https://hackr.io/blog/best-javascript-frameworks

- Angular / React / Vue / Ember / Meteor / Mithril / Node.js / Polymer / Aurelia / Backbone

Web Framework

A web application framework (WAF) is a software framework designed to support the development of dynamic websites, web applications, web services, and web resources. The framework aims to alleviate the overhead associated with common activities performed in web development.

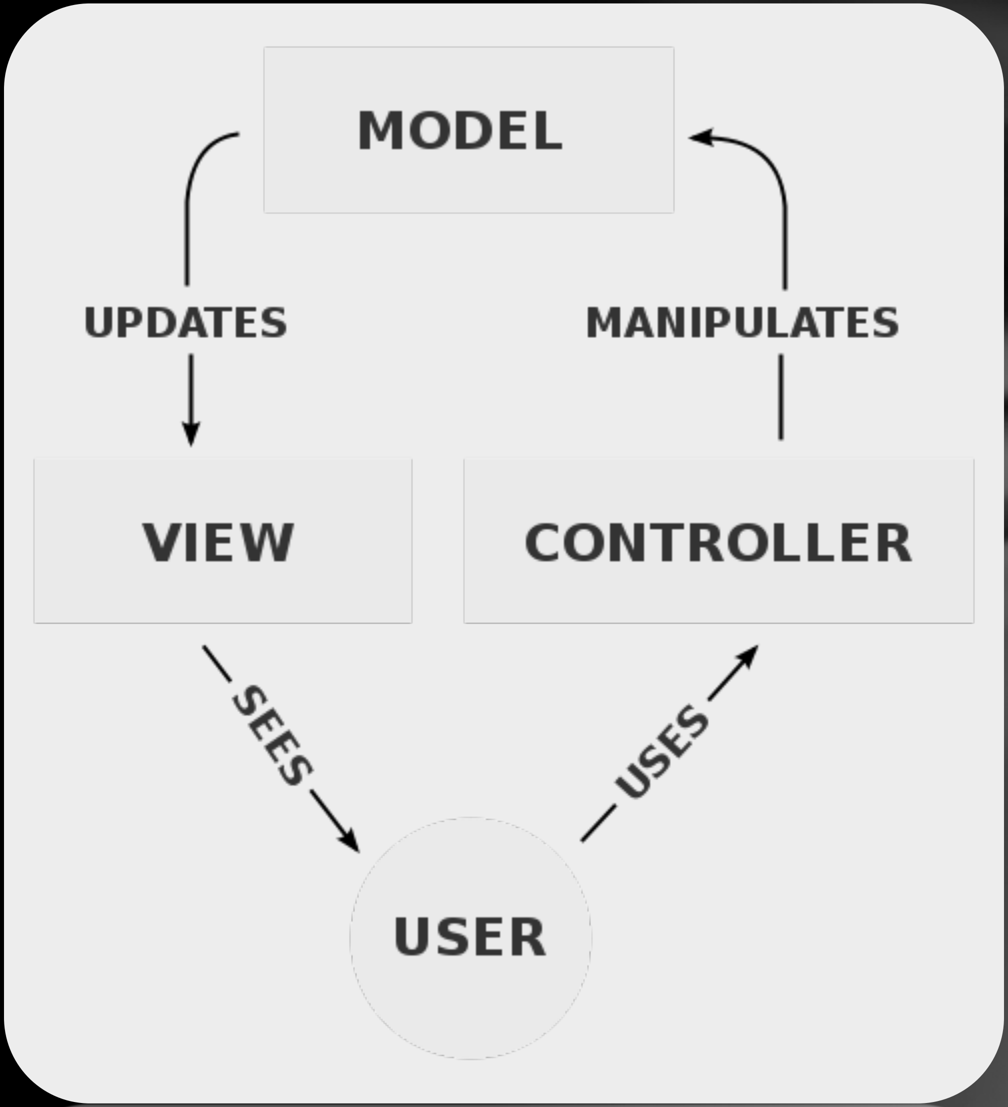
For example, many frameworks provide libraries for database access, templating frameworks, and session management, and they often promote code reuse.

- **Model–view–controller (MVC)**
- **Three-tier organization**

Model–View–Controller (MVC)

Model–view–controller (MVC) is a software pattern for implementing user interfaces. It divides a given software application into three interconnected parts to separate internal representations of information from how it is presented to or accepted by the user.

- The central component, the model, consists of application data, business rules, logic, and functions.
- A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- The third part, the controller, accepts input and converts it to commands for the model or view.


MODEL
UPDATES
MANIPULATES
VIEW
CONTROLLER
SEES
USES
USER

Three-tier architecture

Presentation tier - View

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. (In simple terms, it is a layer that users can access directly, such as a web page or an operating system GUI)

Three-tier architecture

Application tier - Control

Also called business logic, logic tier, or middle tier

The logical tier is pulled out from the presentation tier, and, as its layer, it controls an application’s functionality by performing detailed processing.

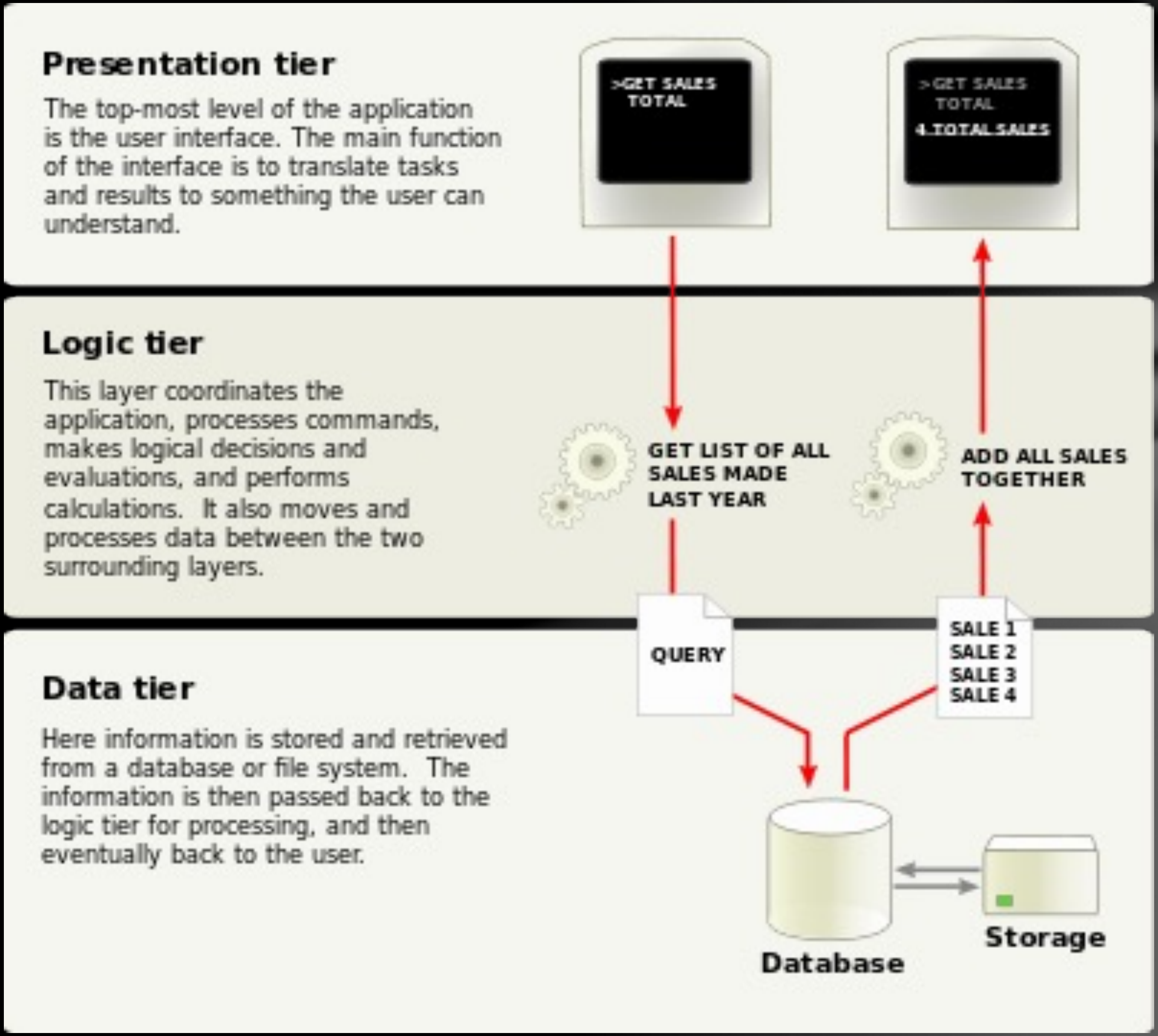
Three-tier architecture

Data tier - Model

The data tier includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data.

The data access layer should provide an API to the application tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms.

Avoiding dependencies on the storage mechanisms allows them to be updated or changed without the application tier clients being affected by or even aware of the change.



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

An example of three layers - SSH

What’s SSH

SSH is an integrated framework of struts + spring + hibernate, a very popular open-source Web application framework.

SSH framework is divided into four layers: presentation layer, logic layer, data persistence layer, and domain module layer to help developers build a clear structure in the short term, can be good reusability and easy maintenance of Web applications procedures.

- Struts infrastructure systems as a whole are responsible for MVC separation, control of the business logic;
- Hibernate is used to support persistence.
- Spring is in charge of management, managing struts and hibernate.

More Info

Introduction to Web Frameworks: https://hackr.io/blog/web-development-frameworks

- Express (with node.js)
- Django (for Python)
- Ruby on Rails
- Laravel (for PHP)
- Spring (for Java)

Introduction to Java Frameworks: https://rollbar.com/blog/most-popular-java-web-frameworks/

- Spring
- JSF (Java Server Faces)
- GWT (Google Web Toolkit)
- Play!
- Struts
- Vaadin
- Grails
- Hibernate (for ORM)

Web Server Language

Server Side Scripting

Many dynamically built web pages are mostly static. CGI, ISAPI, and Servlets make you generate the entire page via your program, even though most of it is always the same.

Server-side scripting environments allow you to include server-side scripts in HTML documents (as well as client-side scripts).

- Server-side scripts are interpreted by the web server and translated into HTML before being sent to the client, so the client's browser doesn't even see the server-side scripts.

Server-side scripts are like CGI programs in that they can access server-side resources such as databases but can't directly interact with the user.

- They can't, for example, directly pop up a message box.

Like ISAPI and Servlets, server-side scripts are typically executed within the same process as the webserver.

Web Server Scripting

Allows easy implementation of complex functionality (also for non-programmers)

- Think: Is this a good idea?
- Example scripting languages: ASP, JSP, PHP, Perl, Python

Scripts are installed on the Web server and return HTML as output that is then sent to the client.

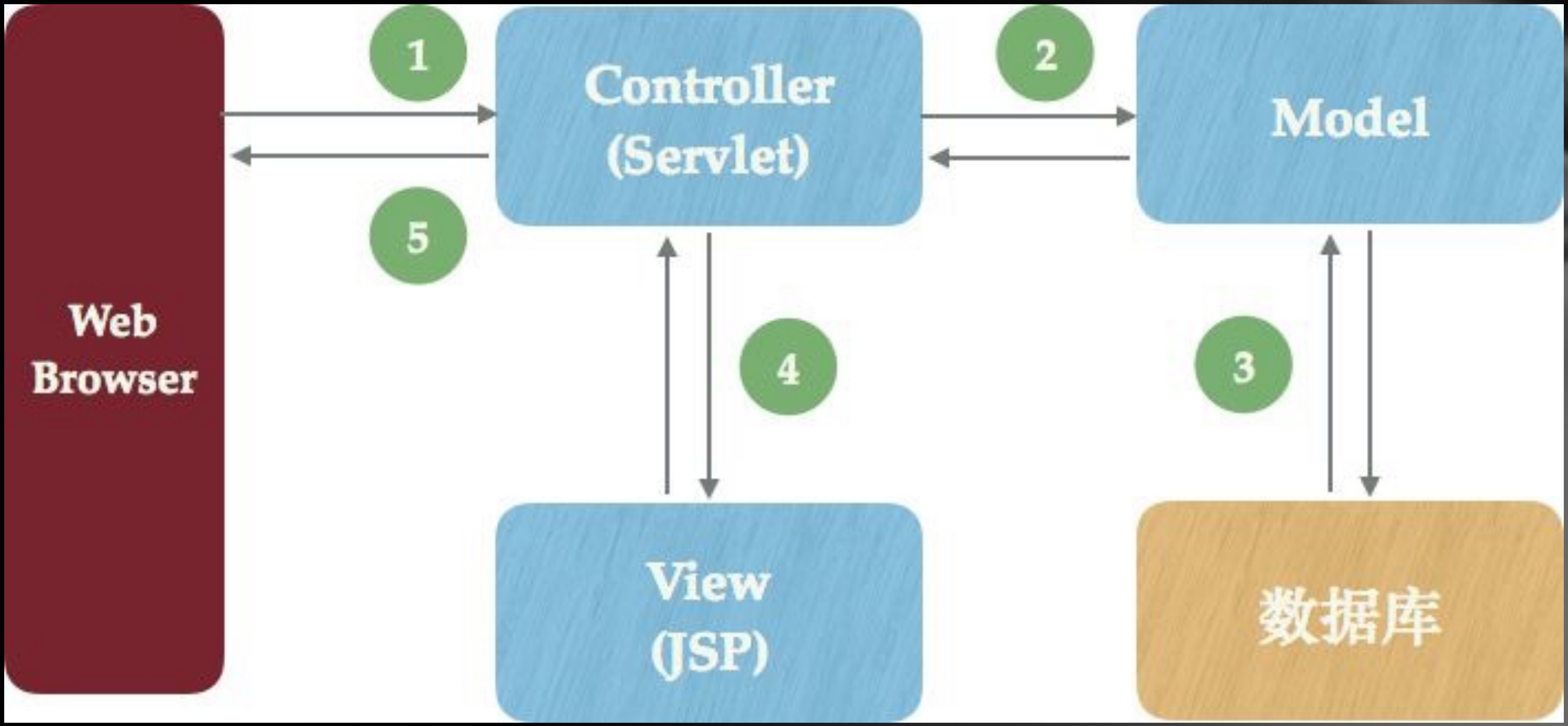
Java Servlet / JSP

What's JSP

The JavaServer Pages is a technology for inserting dynamic content into an HTML or XML page using a Java servlet container.

- In other words, instead of sending HTML pages to web clients that are always the same for everyone, you can send HTML pages that can be different for each client and each time they receive it (using database data, for instance).

Data Flow



Syntax

JSP pages use several delimiters for scripting functions.

- <% … %>, scriptlet, is a fragment of Java code run when the user requests the page.
- <%= … %>, expressions, places an expression to be evaluated inside the java servlet class. Expressions should not be terminated with a semi-colon.
- <%@ … %>, comment, does nothing. It is ignored. It lets you document the file. It differs from an HTML comment as an HTML comment (<! -->) will appear in the generated HTML file.

Example

```
<p>Counting to three:</p>
<% for (int i=1; i<4; i++) { %>
   <p>This number is <%= i %>.</p>
<% } %>
<p>OK.</p>
```

The output displayed in the user's web browser would be:

```
Counting to three:

This number is 1.

This number is 2.

This number is 3.

OK.
```

Directives

JSP directives are added at the top of a JSP page. These directives control how the JSP compiler generates the servlet.

include

The include directive informs the JSP compiler to include a complete file into the current file. It is as if the included file’s contents were pasted directly into the original file. This functionality is similar to the one provided by the C preprocessor. Included files generally have the extension "jspf" (for JSP Fragment)

```
<%@ include file="somefile.jspf" %>
```

Directives

page

```
<%@ page import="java.util.*" %> <%-- example import --%>
<%@ page contentType="text/html" %> <%-- example contentType --%>
<%@ page isErrorPage="false" %> <%-- example for non error page --%>
<%@ page isThreadSafe="true" %> <%-- example for a thread safe JSP --
%>
<%@ page session="true" %> <%-- example for using session binding --%>
<%@ page autoFlush="true" %> <%-- example for setting autoFlush --%>
<%@ page buffer="20kb" %> <%-- example for setting Buffer Size --%>
```

Note: Only the "import" page directive can be used multiple times in the same JSP.

Directives

taglib

The taglib directive indicates that a JSP tag library is to be used. The directive requires a prefix (much like a namespace in C++) and the URI for the tag library description.

```
<%@ taglib prefix="myprefix" uri="taglib/mytag.tld" %>
```

Implicit objects

Object	Description
out	The JspWriter used to write the data to the response stream.
page	The servlet itself.
pageContext	A PageContext instance that contains data associated with the whole page. A given HTML page may be passed among multiple JSPs.
request	The HttpServletRequest object that provides HTTP request information.
response	The HttpServletResponse object that can be used to send data back to the client.
session	The HttpSession object that can be used to track information about a user from one request to another.
config	Provides servlet configuration data.
application	Data shared by all JSPs and servlets in the application.
exception	Exceptions not caught by application code.

JSP actions

jsp:include

Includes a specified JSP into the returned HTML page, but it works differently. The Java servlet temporarily hands the request and response off to the specified JavaServer Page. Once the other JSP has finished, control will return to the current JSP. Using this, JSP code will be shared between multiple other JSPs, rather than duplicated.

```
<html>
 <head></head>
 <body>
  <jsp:include page="mycommon.jsp" >
   <jsp:param name="extraparam" value="myvalue" />
  </jsp:include>
  name:<%=request.getParameter("extraparam")%>
 </body>
</html>
```

JSP actions

jsp:param

It can be used inside a jsp:include, jsp:forward, or jsp:params block. Specifies a parameter that will be added to the request's current parameters.

jsp:forward

They are used to hand off the request and response to another JSP or servlet. Control will never return to the current JSP.

```
<jsp:forward page="subpage.jsp" >
  <jsp:param name="forwardedFrom" value="this.jsp" />
</jsp:forward>
```

In this forwarding example, the request is forwarded to a subpage.jsp

JSP actions

jsp:plugin

Older Netscape Navigator and Internet Explorer used different tags to embed an applet. This action generates the browser-specific tag needed to include an applet. The plugin example illustrates an HTML uniform way of embedding applets in a web page.

```
<jsp:plugin type=applet height="100%" width="100%"
  archive="myjarfile.jar, myotherjar.jar"
  codebase="/applets"
  code="com.foo.MyApplet" >
  <jsp:params>
    <jsp:param name="enableDebug" value="true" />
  </jsp:params>
  <jsp:fallback>
   Your browser does not support applets.
  </jsp:fallback>
</jsp:plugin>
```

JSP References

Tutorials:

http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

http://www.coreservlets.com/

Specification:

http://java.sun.com/products/jsp/download.html

PHP

What’s PHP

- PHP is an acronym for "PHP Hypertext Preprocessor."
- PHP is a widely-used, open-source scripting language
- PHP scripts are executed on the server
- PHP costs nothing; it is free to download and use

PHP (Hypertext Pre-Processor)

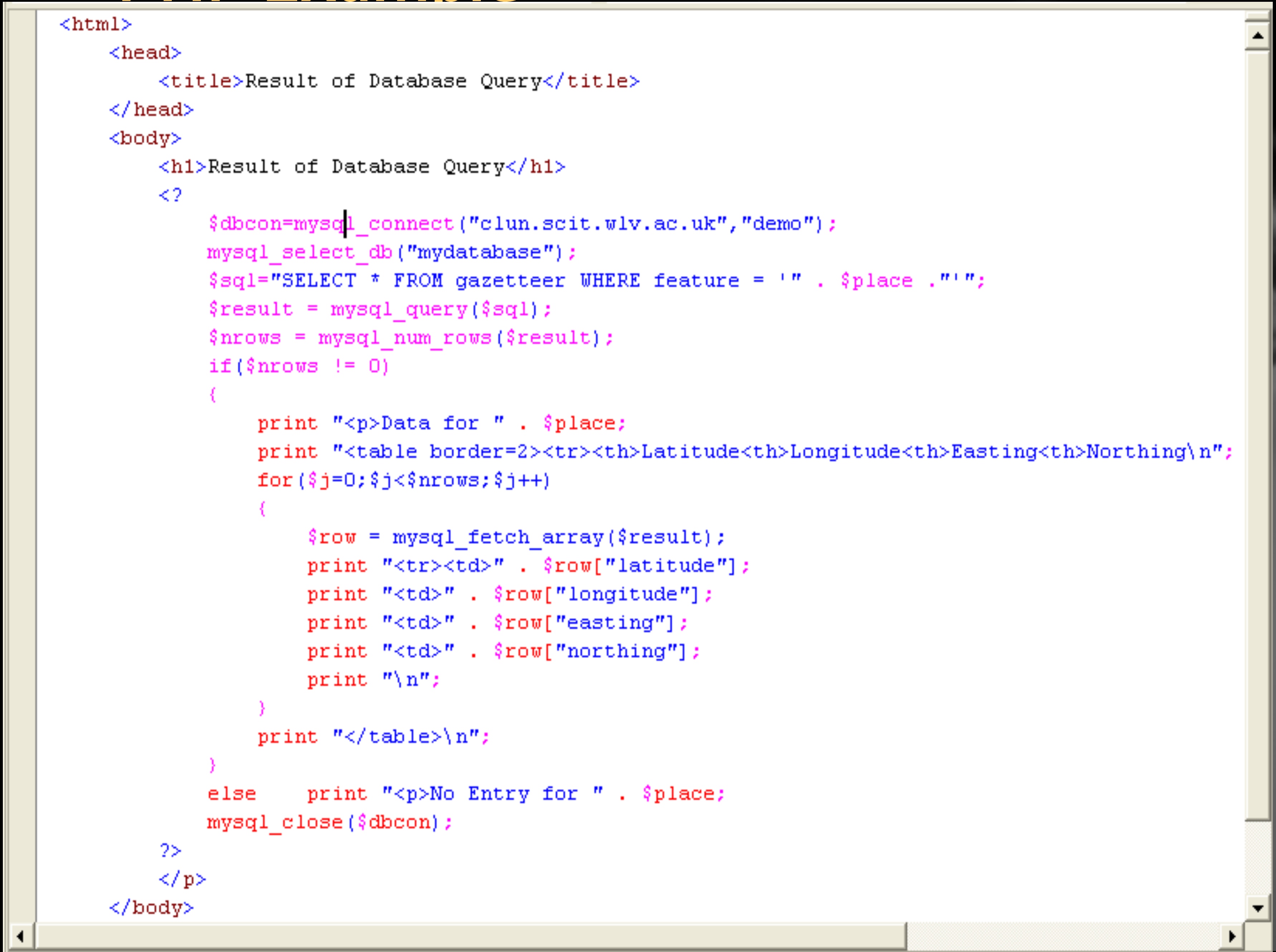
Advantages:

- Cross-platform support (PWS, IIS, and Apache web servers)
- Open Source, developed by Rasmus Lerdorf in 1994.
- The language is specifically designed for the web.
- Typically runs in the process.
- Excellent string processing capabilities (like Perl)
- Tight integration with MySQL (fast)
- Zend optimizing compiler (available commercially)

Disadvantages:

- Quick and dirty ("stubborn function-over-form approach").
- Poor error handling
- "Tedious" objected-oriented programming support.
- Normally interpreted

PHP Example



```
<html>
    <head>
        <title>Result of Database Query</title>
    </head>
    <body>
        <h1>Result of Database Query</h1>
        <?
            $dbcon=mysql_connect("clun.scit.wlv.ac.uk","demo");
            mysql_select_db("mydatabase");
            $sql="SELECT * FROM gazetteer WHERE feature = '" . $place ."'";
            $result = mysql_query($sql);
            $nrows = mysql_num_rows($result);
            if($nrows != 0)
            {
                print "<p>Data for " . $place;
                print "<table border=2><tr><th>Latitude<th>Longitude<th>Easting<th>Northing\n";
                for($j=0;$j<$nrows;$j++)
                {
                    $row = mysql_fetch_array($result);
                    print "<tr><td>" . $row["latitude"];
                    print "<td>" . $row["longitude"];
                    print "<td>" . $row["easting"];
                    print "<td>" . $row["northing"];
                    print "\n";
                }
                print "</table>\n";
            }
            else    print "<p>No Entry for " . $place;
            mysql_close($dbcon);
        ?>
        </p>
    </body>
```

Introduction of SQL Language

SQL

What's SQL

- SQL stands for Structured Query Language
- SQL lets you access and manipulates databases
- SQL is an ANSI (American National Standards Institute) standard

RDBMS

- RDBMS stands for Relational Database Management System.
- RDBMS is the basis for SQL and all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- The data in RDBMS is stored in database objects called tables.
- A table is a collection of related data entries and consists of columns and rows.

A Table in RDBMS

Id	Name	Age	Grade	Comment
1	Alice	18	A	…
2	Bob	20	B	?
3	Calvin	16	F	

Table: Students

Columns: Id, Name, Age, Grade, Comment

Rows: Id = 1, Id = 2 …

SQL Statements

- Most actions you perform on a database are done with SQL statements.
- The following SQL statement selects all the records in the "Customers" table:

```
SELECT * FROM Customers;
```

Keep in Mind That ...

- SQL is NOT case sensitive: select is the same as SELECT.

SQL Commands

Some of The Most Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index.

SQL-Select

The SELECT statement is used to select data from a database.

- The result is stored in a result table called the result set.

SQL SELECT Syntax

```
SELECT column_name,column_name
FROM table_name;
WHERE column_name = value
```

And

```
SELECT * FROM table_name;
```

Navigation in a Result-set

Most database software systems allow navigation in the result set with programming functions like Move-To-First-Record, Get-Record-Content, Move-To-Next-Record, etc.

SQL-Insert Into

The INSERT INTO statement is used to insert new records into a table.

SQL Insert Into Syntax

It is possible to write the INSERT INTO statement in two forms.

The first form does not specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name
VALUES (value1,value2,value3,...);
```

The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

SQL-Update

The UPDATE statement is used to update existing records in a table.

SQL Update Syntax

```
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

Notice

The WHERE clause specifies which record or records that should be updated.
If you omit the WHERE clause, all records will be updated!

SQL - Delete

The DELETE statement is used to delete rows in a table.

SQL Delete Syntax

```
DELETE FROM table_name
WHERE some_column=some_value;
```

Notice

The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

Delete All Data

```
DELETE FROM table_name;
or
DELETE * FROM table_name;
```

More Info

Reference: http://www.w3schools.com/sql/default.asp

Introduction of SQL Database

What’s Database

A database is an organized collection of data. The data are typically organized to model relevant aspects of reality in a way that supports processes requiring this information.

Database management systems (DBMSs) are specially designed software applications that interact with the user, other applications, and the database to capture and analyze data.

A general-purpose DBMS is a software system designed to allow the definition, creation, querying, updating, and administration of databases.

Well-known DBMSs: MySQL, SQL Server, Access, Oracle, Sybase, DB2.

MySQL

MySQL is the world's most widely used open-source relational database management system (RDBMS).

It is named after co-founder Michael Widenius's daughter, My. The SQL phrase stands for Structured Query Language.

Open Source LAMP software

Linux	http://www.linux.org
Apache	http://www.apache.org
MySQL	http://www.mysql.com
PHP	http://www.php.net

Review

- **Introduction of Http**
  - Http
  - URLS
  - Https
- **Introduction of Front-End Web Language**
  - HTML
  - CSS
  - JavaScript
- **Introduction of Server Framework and Language**
  - MVC / Three-tier architecture
  - SSH
  - Java / JSP / PHP
- **Introduction of Database & SQL**
  - Select / Insert-Into / Update / Delete
  - MySQL