# Moving on

**Code: moving-on**

## Author and version

- Daniel K. Schneider
- Email: Daniel.Schneider@tecfa.unige.ch
- Version: 0.5 (modified 6/5/07 by DKS)

## Objectives

- Understand the architecture of Internet
- Web servers and server-side scripting (GET and POST HTTP requests)
- HTML forms
- Scripting languages overview (PHP)
- AJAX and web services (demonstration of "webtops/webOSs etc.")
- SVG and SMIL formats

# Prerequisites

- Editing XML (being able to use a simple DTD)
- XML namespaces (some)
- HTML and CSS (some)

# Warning

- We don’t cover any details, this is just "introductory information"

# Disclaimer

- There may be typos (sorry) and mistakes (sorry again)
- Please also consult a textbook !

# Contents

1. Internet and the WWW 4

1.1 Internet history 4

1.2 Architecture of the Internet 5

A.The Physical Layer 5

B.The Data link layer 5

C.The Network layer 5

D.The Transport Layer 6

E.The Application Layer 6

1.3 HTTP 7

2. Server-side scripting 9

2.1 Common Gateway Interface (CGI) 9

2.2 HTML Forms 10

2.3 Server-side scripting with PHP 12

3. AJAX 15

4. Multimedia XML formats 16

4.1 SVG - Scalable Vector Graphics 17

5. Next steps 18

5.1 Reading 18

5.2 Homework 18

5.3 Final exam 18

# 1. Internet and the WWW

## 1.1 Internet history

- Internet is now almost 40 years old ...

1. In 1962, package switching was invented (the basis of modern networks)
2. In December 1969, the first version of Arpanet (Internet) went online.
3. In 1972 Ray Tomlinson (BBN) created the first e-mail program
4. 1973 - 1978 TCP/IP was developed (the low-level Internet communications protocol)
5. The 80’s: several different networks existed (UseNet, BitNet, BBS, etc.)
6. 1992 -Tim Berners-Lee et al. invented the World Wide Web with its two main components: HTTP and HTML.
7. 1995 - Internet goes commercial. Also, Microsoft enters the game.

# 1.2 Architecture of the Internet

- Internet can be described in terms of layers

## A. The Physical Layer

- Various sorts of conducts that connect computers (mostly optical fiber cables but also WiFi and telephone lines)

## B. The Data link layer

- This is the layer which transfers data between adjacent network nodes in a wide area network or between nodes on the same local area network segment. E.g.
- Ethernet (mostly at work, but also in some homes)
- DSL (e.g. ADSL) over telephone lines, mainly used to connect a Home network [edit]

## C. The Network layer

- is responsible for end to end (source to destination) packet delivery, whereas the data link layer is responsible for node to node (hop to hop) packet delivery.
- The best known protocol is the Internet Protocol (IP). It breaks down a message to packets and can send them over through several nodes over a heterogeneous network.
- IP provides an unreliable service, i.e. data can arrive corrupt, out of order, be lost etc. Errors must be repaired at the next level, e.g. with TCP.

# D. The Transport Layer

- The best known layers are TCP and UDP.
- The Transmission Control Protocol (TCP) can create connections between two computers, over which they can exchange streams of data using so-called Stream Sockets. The protocol guarantees reliable and in-order delivery of data from sender to receiver. It can distinguish data for applications, e.g. you can at the same time surf on the Web, receive email and be connected to a virtual world. HTTP (World Wide Web) servers use TCP and IP.
- The User Datagram Protocol (UDP), also called Universal Datagram Protocol or Unreliable Datagram Protocol can send short messages sometimes known as datagrams. It does not provide the reliability and ordering that TCP does. However, this makes UDP faster and more efficient for many lightweight or time-sensitive purposes, e.g. video streaming.

# E. The Application Layer

- ***Many*** standards that define messages and data formats understood by specific applications running at each end of the communication. E.g.
- ***Hypertext Transfer Protocol*** (HTTP), the underlying communication protocol of the World Wide Web that specifies how a web server and a navigator talk to each other.
- ***File Transfer Protocol*** - FTP
- ***Simple Mail Transfer Protocol*** - SMTP (e-mail transmissions).
- ***Real-time Transport Protocol*** (for video streaming, .e.g. for windows media player).
- Simple Object Access Protocol (SOAP), one of the protocols that define how servers can talk to each other by sending XML-based messages.

# 1.3 HTTP

- Hypertext Transfer Protocol (HTTP) is a method used to transfer or convey information on the World Wide Web
- Browsers can sent requests to an URN (URL) using methods like GET, POST, etc.

### Browser requests consist of the following:

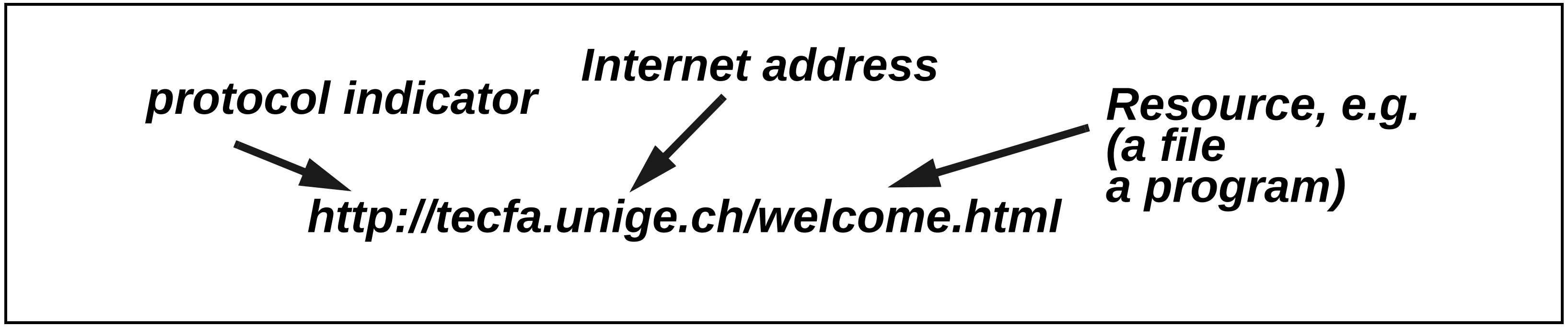
- Request line, such as GET /images/logo.gif HTTP/1.1, which requests the file logo.gif from the /images directory
- Headers that will give the server information about your browser, cookies etc.
- An empty line
- An optional message body (e.g. POST contents from a form)

Example:

```
GET /index.html HTTP/1.1
Host: www.example.com
```

### Universal Resource Locators” (URLs)

```
URL = <protocol>://<address>/<resource>
```



## Server replies (simplified)

- Firstly some response lines that will not display,
- ... followed by a blank line and then contents that the browser will display
- The first line includes a status code (e.g. "200 OK" means "here is what you asked")

Example:

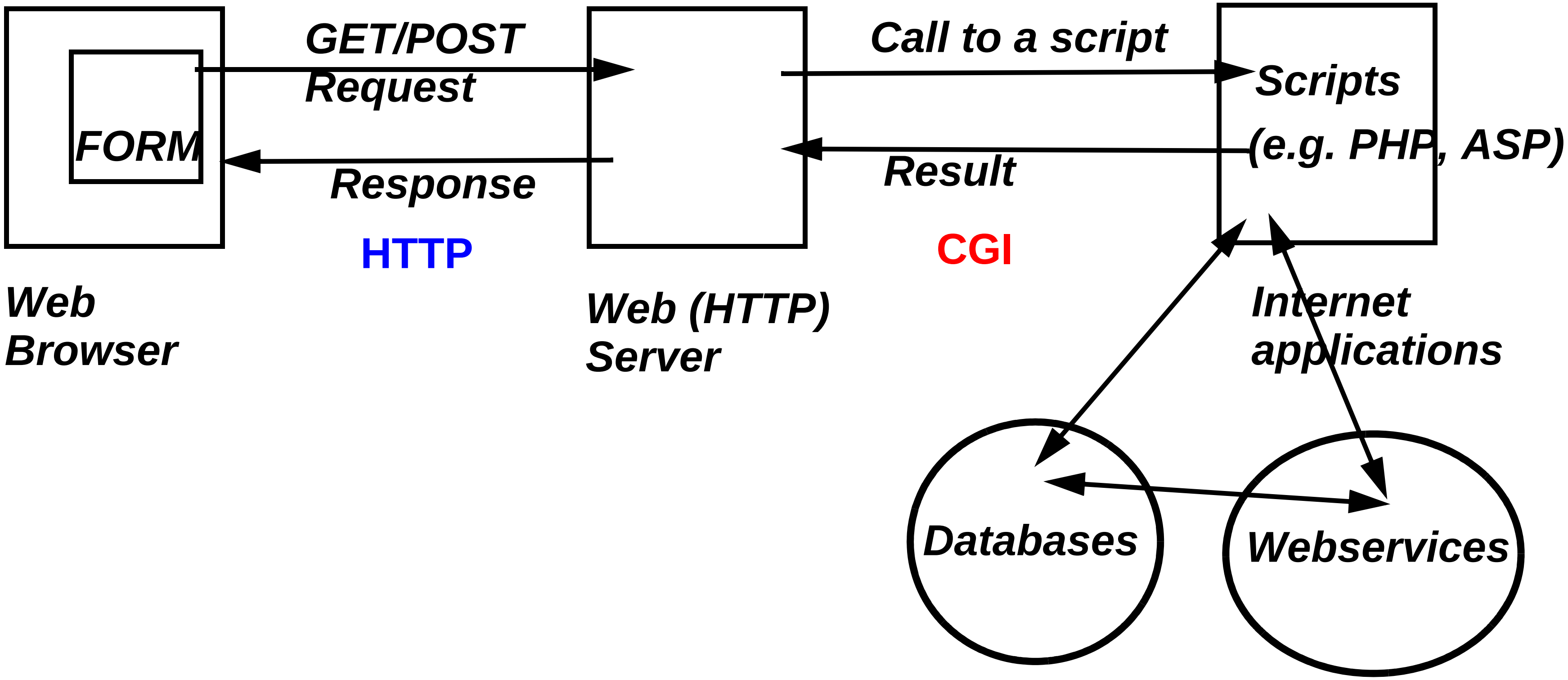
```
HTTP/1.1 200 OK
Date: Sun, 06 May 2007 19:13:32 GMT
Server: Apache/2.2.3 (Win32) PHP/5.1.5
X-Powered-By: PHP/5.1.6
Content-Length: 252
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

(contents will be appended here)
```

# 2. Server-side scripting

- Your browser talks to a web server who in turn talks to a computer program ...

## 2.1 Common Gateway Interface (CGI)

- Web server talking to a server-side script ....



## 2.2 HTML Forms

- HTML Forms are the "basic" HTML elements for client or server-side interactivity
- (DHMTL can do more, but includes forms)

### Example

*url:* **test-form.html**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head><Title>HTML Form example</title></head>
  <body>
    <h1>HTML Form example</h1>
    <form method="post" action="test-form.php">

      <h3>Tell us ...</h3>

      <strong>Name</strong><br>
      <input type="text" name="name" size="25"><p/>
      <strong>First Name</strong><br>
      <input type="text" name="first_name" size="25"><p>
      <strong>I need your documentation</strong>
      <input type="checkbox"><p>

      <strong>Country</strong>
      <select name="pays" size=1>
       <option>CH:  Switzerland
       <option>D:   Germany
      <option>F:   France
      <option>US:  USA
      <option>A:   Austria
      <option>I:   Italia
      </select><p><p>
    <strong>Gender</strong>

    <input type="radio" name="gender">Male
    <input type="radio" name="gender">Female
    <h3>Major</h3>
    You may choose more than one<p>
    <input type="checkbox" name="domain" value="Ped">Pedagogy
    <input type="checkbox" name="domain" value="Psy">Psychology
    <input type="checkbox" name="domain" value="CS">Computer Science
    <input type="checkbox" name="domain" value="Other">Other
    <h4>Commentaire</h4>
     Please, describe your interests <p>
    <textarea name="comments" rows=4 cols=60></textarea><p>
    <input type="reset" value="Reset data">
    <input type="submit" value="Submit/register"><p>
    </form>
    <hr>
  </body>
</html>
```

# 2.3 Server-side scripting with PHP

## PHP scripts are put on a web server (like HTML files)

PHP code is inserted between: **`<?php ..... ?>`**

```
<?php echo("I am a php instruction\n"); ?>
```

### Example

*url:* **calculate.html and calculate.php (but these files need to be put on a server)**

The HTML form:

```
<form action="calculate.php" method="post">

What do you know about HTML ?
<input type="radio" name="choice" value="1" checked>little
<input type="radio" name="choice" value="2">some
<input type="radio" name="choice" value="3">everything

<br>
What is your programming experience ?
<input type="radio" name="choice2" value="1" checked>none
<input type="radio" name="choice2" value="2">some
<input type="radio" name="choice2" value="3">good
<P>
<input type="submit" value="See result!">

</form>
```

## Contents of the Php file

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head><title>Simple test with</title></head><body>
    <h1>Simple test with PHP</h1><hr>
<?php

// Get variables from the form
$choice = $_POST['choice'];
$choice2 = $_POST['choice2'];

// Compute the score
$score = $choice + $choice2;
// Display the score
echo "<h3>Your score is " . $score . "</h3>";
if ($score < 3) {
  echo "<p>You are a beginner</p>";
} elseif ($score < 5) {
  echo "<p>You have some knowledge</p>";
} else {
  echo "<p>You are an expert !</p>";
}
?>
<hr>DKS - 2007
</body></html>
```

- No need to understand CGI details, PHP makes it quite easy ....

# Your future as web designers

- is not creating web pages
- **is configuring skins for portals and web services ...**
- is working together with applications developers ...
- therefore you might want to acquire a minimum of PHP, ASP, etc.
  ... just enough to able to edit CSS and template files

# To play on your personal computer, install a WAMP package

- Windows, the operating system;
- Apache, the Web server;
- MySQL, the database management system
- PHP

*url:* **http://edutechwiki.unige.ch/en/WAMP (also includes WAMPs running on a memory stick)**

*url:* **http://www.wampserver.com/ (currently most recommended by DKS)**

*url:* **http://www.chsoftware.net/en/useware/wos/wos.htm (best WAMP for memory sticks)**

# 3. AJAX

- AJAX is the latest trend in interactive web pages

The Ajax technique uses a combination of:

- XHTML (or HTML) and CSS, for marking up and styling information.
- The DOM accessed with a client-side scripting language, usually ECMAScript (JavaScript)
- The XMLHttpRequest object is used to exchange data asynchronously with the web server.
- XML is sometimes used as the format for transferring data between the server and client.

## Webtops

- Webtops or so-called "Ajax homepages" are popular AJAX applications
- You can configure your personal access point to various Internet services

*url:* **http://edutechwiki.unige.ch/en/Webtop (introductory article)**

*url:* **http://edutechwiki.unige.ch/en/Virtual_office (introduction for on-line tools)**

## Hands-on activity:

*url:* **http://www.netvibes.com/ (a popular webtop)**

# 4. Multimedia XML formats

- Some of the Web’s future lies in XML multimedia formats
- Most of these will be embedded in XHTML
- Free and commercial authoring tools exist (so you do not need to hand code !)

## Most important multimedia XML formats

- SVG
- SMIL and various profiles (e.g. XHTML+SMIL, SVG+SMIL)
- MathML
- X3D (3D)

## Current situation

- IE explorer integrates XHTML+Smile (called HTML+Time)

  *url:* **http://msdn2.microsoft.com/en-us/library/ms533112.aspx (tutorials, reference and examples)**
- Firefox implements SVG and XHTML+SVG (DOM scripting is supported)
- All browsers can launch helper applications or plugins for these formats (e.g. Adobe SVG, Real Player, QuickTime, etc.)

# 4.1 SVG - Scalable Vector Graphics

- SVG is a language for describing two-dimensional graphics and graphical applications in XML.
- SVG 1.1 is a W3C Recommendation and forms the core of the current SVG developments
- SVG 1.2 is the specification currently being developed as is available in draft form
- The SVG Mobile Profiles: SVG Basic and SVG Tiny are targeted to resource-limited devices and are part of the 3GPP platform for third generation mobile phones.
- For more, see: **http://en.wikipedia.org/wiki/Svg**

## Example activity

*url:* **test.svg (open with Firefox)**

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "file:/C:/Program Files/
Exchanger XML Lite 3.2/types/SVG/Validation/svg10.dtd">

<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">
    <rect x="0" y="0" width="100" height="100" fill="yellow"></rect>
    <circle cx="100" r="50" fill="red" cy="100" opacity="0.5"></circle>
    <line x1="10" x2="100" y1="10" y2="100" stroke="black"
          stroke-width="5"></line>
</svg>
```

- Activity: Open this file in XML Exchanger lite and change a few attribute values. You also may add other elements ....

# 5. Next steps

## 5.1 Reading

Deitel Textbook Chapter 21, Web Servers (IIS and Apache) - optional reading !

## 5.2 Homework

Find some extra information about these topics on wikipedia.org

## 5.3 Final exam

- Topics
  - HTML, DHTML and CSS principles
  - XML (material covered in slides)
  - XSLT (material covered in slides)
  - "Moving on" topics covered in these slides
- Paper-based: short questions and MCQs
- If you did a little homework, little preparation is needed ...
- You may use slides, textbooks and personal notes (no Internet)