



Feel free to contact me:

- Join a Zoom meeting during an officially scheduled lecture
- E-mail me

Assignment

You can find the “Simple Static Web” assignment in ReCodEx. Your goal is to create a simple static web, the specification of this assignment in ReCodEx is quite detailed. You will be rewarded by one point if your solution meets the required criteria.

Except for ReCodEx, the assignment has to be approved by myself. Your web page has to look like a real web page, it is not enough to just stack the mandatory elements with no concept. You will find my comments to your solution directly in ReCodEx. If your solution is approved, I will reward you one extra point.

You need to get 1+1 points to be able to take the “Introduction to networking” test.

WWW – World Wide Web

- WWW is the most used Internet service
- Started as an experiment in CERN
- Last year's description
 - Platform for information and data exchange
 - Environment for applications, that are accessible from anywhere
- This year
 - Social interactions
 - Shopping
 - Culture
 - Studying
 - Entertainment ...

facebook

rohlik
.CZ

zoom

DRAMOX

NETFLIX

World Wide Web and Internet are not the same. Web is one of the Internet's (most used) services.

Ancient history

- 1945 – Vannevar Bush
 - Human brain works with associations
 - hypothetical electromechanical device Memex – "enlarged intimate supplement to one's memory", bookmark list of static microfilm pages
- '60s – Theodore Nelson first used the word hyper-text, i.e. text linked with associations
 - Project Xanadu
 - System for sharing information
 - Implemented as prototype

Vannevar Bush was an American engineer, inventor and MIT professor. During his career, Bush patented a string of his own inventions. He is known particularly for his engineering work on an analog computer that could solve differential equations, and for memex. The memex, which he began developing in the 1930s, was a hypothetical adjustable microfilm viewer with a structure analogous to that of hypertext. The memex and Bush's 1945 essay "As We May Think" influenced generations of future computer scientists. Bush wanted the memex to emulate the way the brain links data by association. It was intended to be sort of extension to human memory:

"[A] device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

In 1963 Ted Nelson coined the terms "hypertext" and "hypermedia". He was strongly influenced by the work of Vannevar Bush.

From 1960 on he worked on Project Xanadu with the goal of creating a computer network with a simple user interface. Nelson dedicated most of his productive life to this project. Projekt Xanadu is still running. According to its administrators: "The World Wide Web trivialises our original hypertext model with one-way ever-breaking links and no management of version or contents."

History (Middle Ages)

- Tim Berners-Lee – inventor of World Wide Web
 - 1980 – while in CERN he proposed a project based on the concept of hypertext to facilitate sharing and updating information among researchers
 - 1989 – he saw an opportunity to join hypertext with the Internet and implemented a server-client communication
 - He published the very first web page
<http://info.cern.ch/hypertext/WWW/TheProject.html>
- NCSA Mosaic
 - 1993 – web browser by Marc Andreessen and Eric Bina
 - 1995 => Internet Explorer



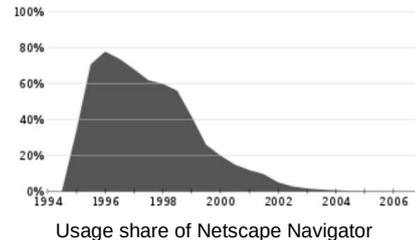
Tim Berners-Lee designed and implemented the first web browser which could be used as web editor as well (the software was called WorldWideWeb). He is also the creator of HTTP protocol and the first web server – CERN HTTPd (Hypertext Transfer Protocol daemon).

- British scientist sir Tim Berners-Lee is the director of the World Wide Web Consortium (W3C) which oversees the continued development of the Web.
- Berners-Lee published the first web site, which described the WorldWideWeb project itself, 30 years ago, on 20 December 1990.
- WWW software included WYSIWYG editor, Berners-Lee didn't expect web developers to know HTML.
- “Creating the web was really an act of desperation, because the situation without it was very difficult...” (*Tim Berners-Lee*)

Mosaic is one of the first web browsers. It was instrumental in popularizing the World Wide Web by integrating multimedia such as text and graphics. Starting in 1995, Mosaic lost market share to Netscape Navigator. In 1995 Microsoft licensed Mosaic to create Internet Explorer.

History (Modern age)

- 1996 – the war of browsers
 - Internet Explorer vs. Netscape Navigator
- 1997 – HTML 4 published as a W3C Recommendation
- 2002 – The first idea of “Web 2.0”
 - The content of web is created by users
- 2004~2006 – introduction of AJAX applications
- 2010 – HTML5
 - Beginning of an end of Flash (Adobe will no longer update or maintain Flash after December 2020)



In the mid-'90s Netscape Navigator became a synonym to a web browser. With Netscape, people using dial-up connections could begin reading the text of a web page within seconds of entering a web address, even before the rest of the text and graphics had finished downloading. This made the web much more likeable to the average user.

By the end of the decade, Netscape's web browser had lost dominance over the Internet Explorer. Netscape became buggy and Microsoft marketing strategy was better.

Web 2.0 – refers to websites that emphasize user-generated web content, as opposed to a static content. This term doesn't refer to a technical aspects of web, but rather to the change in the way the web sites are designed and used.

AJAX applications – web is becoming much more interactive (AJAX will be explained later).

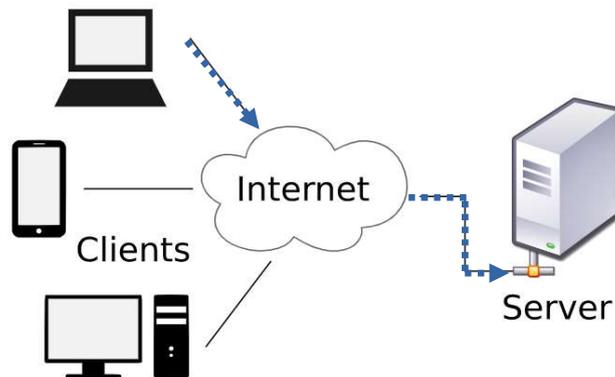
Adobe Flash is a multimedia software platform used for production of animations, desktop applications, mobile apps, mobile games, and embedded web browser video players. Flash uses ActionScript as its programming language.

Flash is on its decline, due to the reliance on Adobe for decoding Flash, security concerns, the lack of support on Android and in iPhones and insufficient accessibility.

Flash functionality is being replaced by HTML5 features.

Displaying a web page - **client**-server architecture

← → 🔄 google.cz/search?q=kotatko



Introduction to networking (2020)

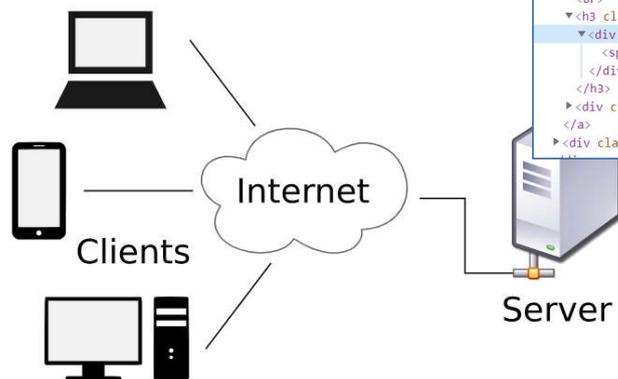
6 / 32

This figure shows what needs to be done to display a web page in your web browser. To describe this process we use the client-server concept:

- Web pages are located on servers (by a server we mean a computer that is connected to the internet at the static IP address)
- In the first step, user enters an address (to be more precise – a URL) into the browser's address bar. Web browser (running on any device) takes the role of a client.
- The request travels through the internet until it reaches the requested server.

... (to be continued)

Displaying a web page - client-server architecture



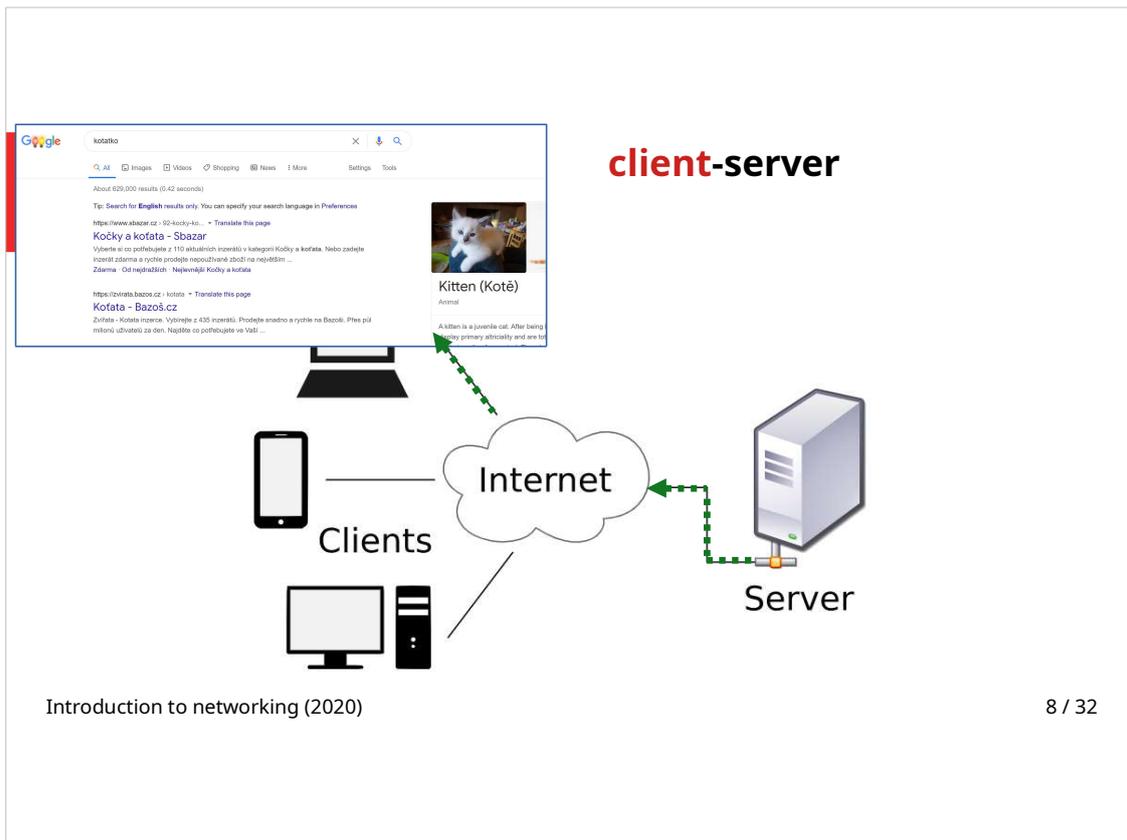
```
<a href="https://www.sbazar.cz/92-kocky-kotata/?url?sa=t&source=web&rct=j&url=https://www.sbazar.cz/92-kocky-kotata&ved=2ahUKewi3h5H5v7DUKHagyBZAQFjAAegQIARAC">  
<br>  
<h3 class="LC201b DKV0Md">  
  <div class="ellip"> == $0  
    <span>Kocky a kotata - Sbazar</span>  
  </div>  
</h3>  
> <div class="TbwUpd NJjxre">...</div>  
</a>  
> <div class="B6fmyf">...</div>
```

Introduction to networking (2020)

7 / 32

- When the request reaches the server, the specified HTML file (or multimedia content) is located (or generated by the server) and retrieved (this process will be explained later)

... (to be continued)



- HTML source code (and other requested files) are delivered to the client.
- Source code is interpreted by the browser and displayed in a user-friendly way.

Web page



```

<!DOCTYPE HTML>
<html>
<head>
  <title>Google</title>
  <link rel="stylesheet"
        type="text/css"
        href="styles.css">
</head>

<body>
Text contents...
...

<source src="movie.mp4">
</video>

```

```

styles.css
body {
  Font: 12pt Calibri;
}
p {
  Margin: 10px;
}

```




Introduction to networking (2020) 9 / 32

Web pages are written in HTML markup language. HTML source code describes the semantic meaning of the single elements.

CSS (Cascading Style Sheets) specify what will the elements look like (font, color, layout).

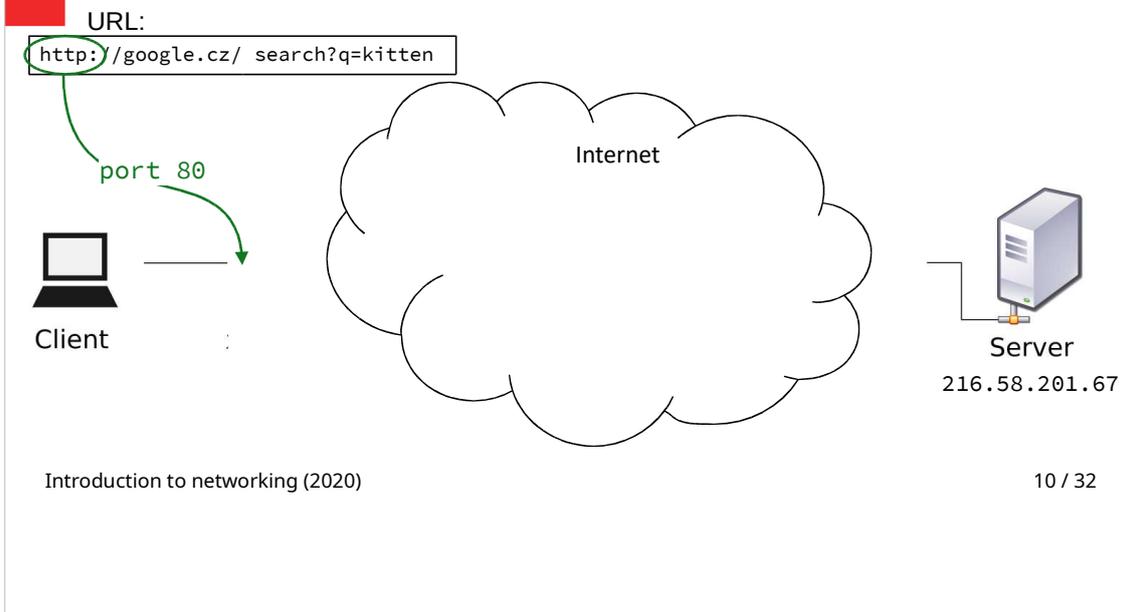
A web page can contain images and other multimedia, server sends these files to a client along with the HTML source code.

HTML can contain links to other web pages or multimedia that can be located on a different server.

Note:

- Web browsers try to (somehow) display even the source code that is not 100% correct.
- The way that clients interpret HTML can slightly differ, especially when there are errors in the code.

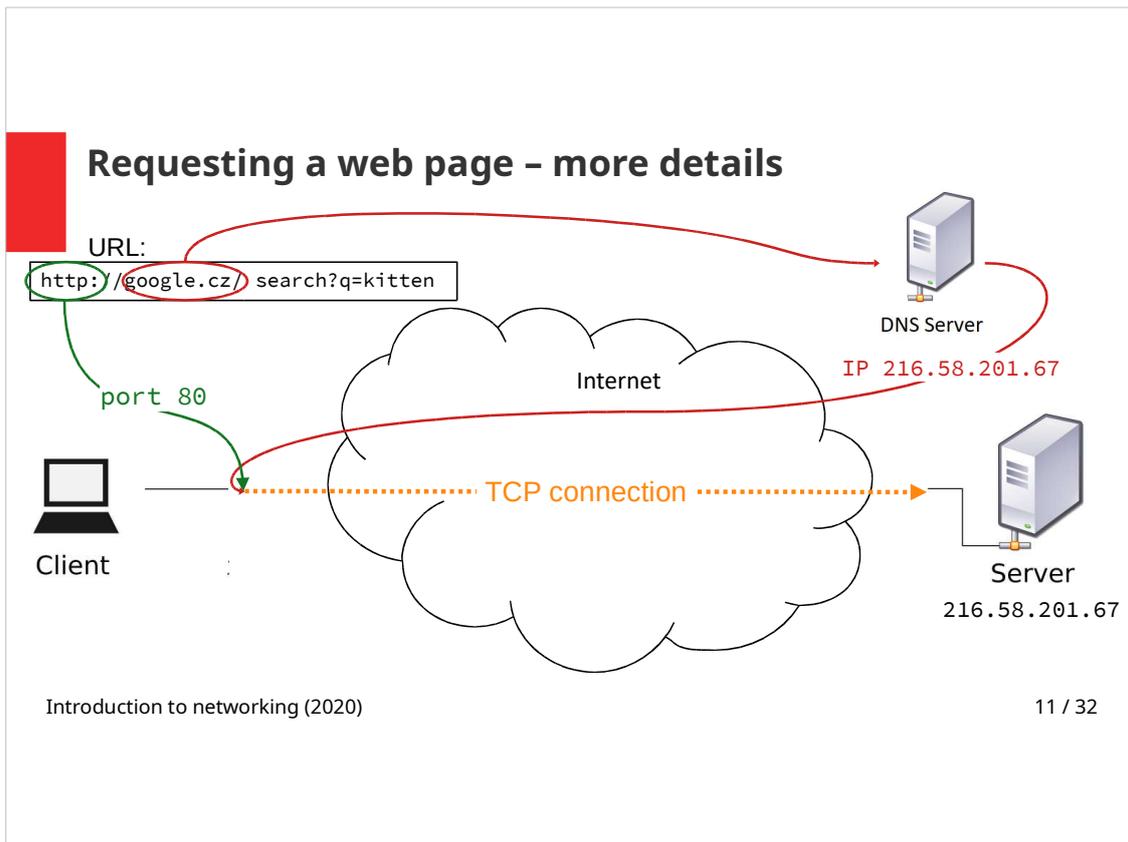
Requesting a web page - more details



We will describe the whole process once again, this time with several more technical details:

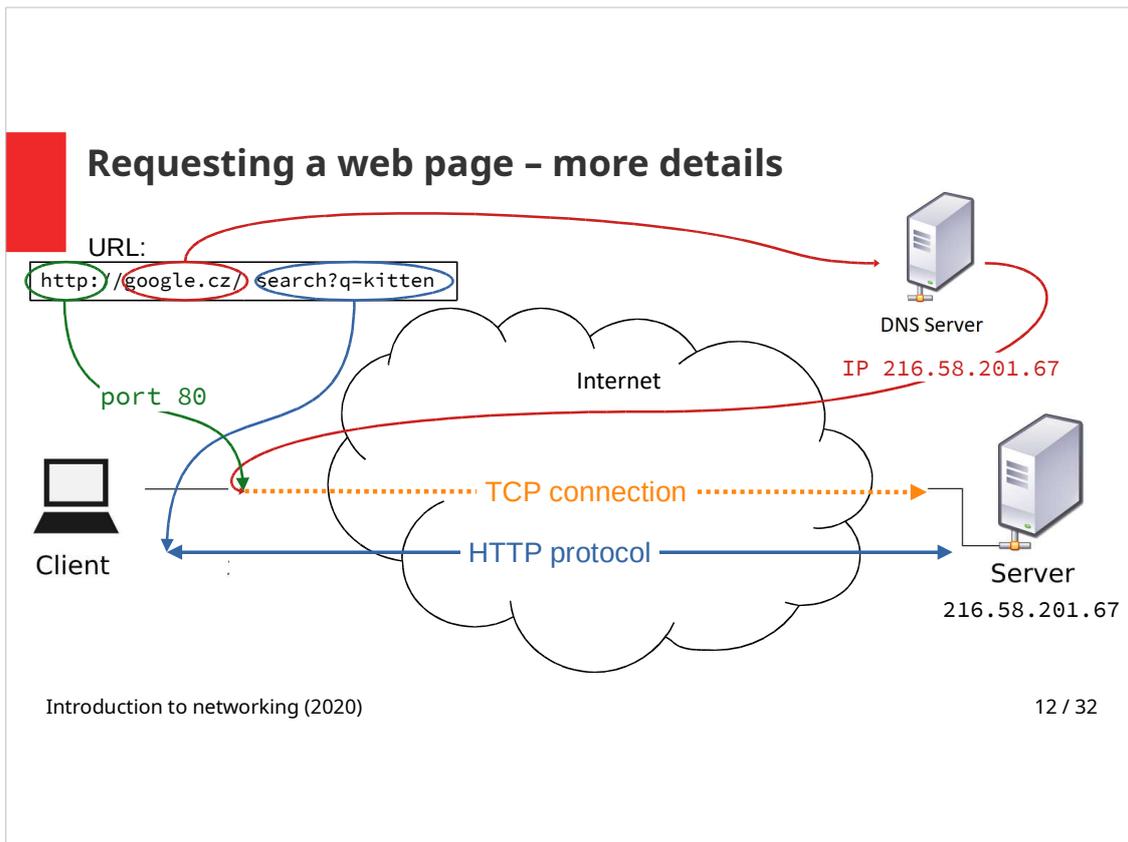
- After a user types a URL to an address bar, it is “translated”:
- If the port is not explicitly specified, the port number is set according to a used protocol, e.g. HTTP protocol uses port 80

... (to be continued)



- Domain name is translated to an IP address:
 - DNS (Domain Name System) client (which is a part of an operating system) allows a browser to send a request to a DNS server to translate the domain name
 - DNS server responds with the IP address
- TCP connection between client and server with the obtained IP address is established.

... (co be continued)



- After the TCP connection is established, the HTTP protocol is used for communication between client and server.
- The remaining part of the URL (after the domain name) is used in HTTP Request Headers. This part tells the server the location of the document within the server. Also the parameters for the server-side script can be set in this part of URL.

URL (web site address)

- Uniform Resource Identifier (URI)
 - Identification string of the following format:
`<schema>:<hierarchical_part>?<query>#fragment`
 - `query` and `fragment` parameters are optional
- Uniform Resource Locator
 - URI that describes a location of a resource
`protocol://username:password@domain:port/path
?parameter=value¶meter2=value2#element_id`
 - e.g.:
`http://is.cuni.cz/stud/schedule/teacher.php?teacherID=13270`

URL is a special case of URI.

Some parts of URL are mandatory, some are optional and some of them are usually left out:

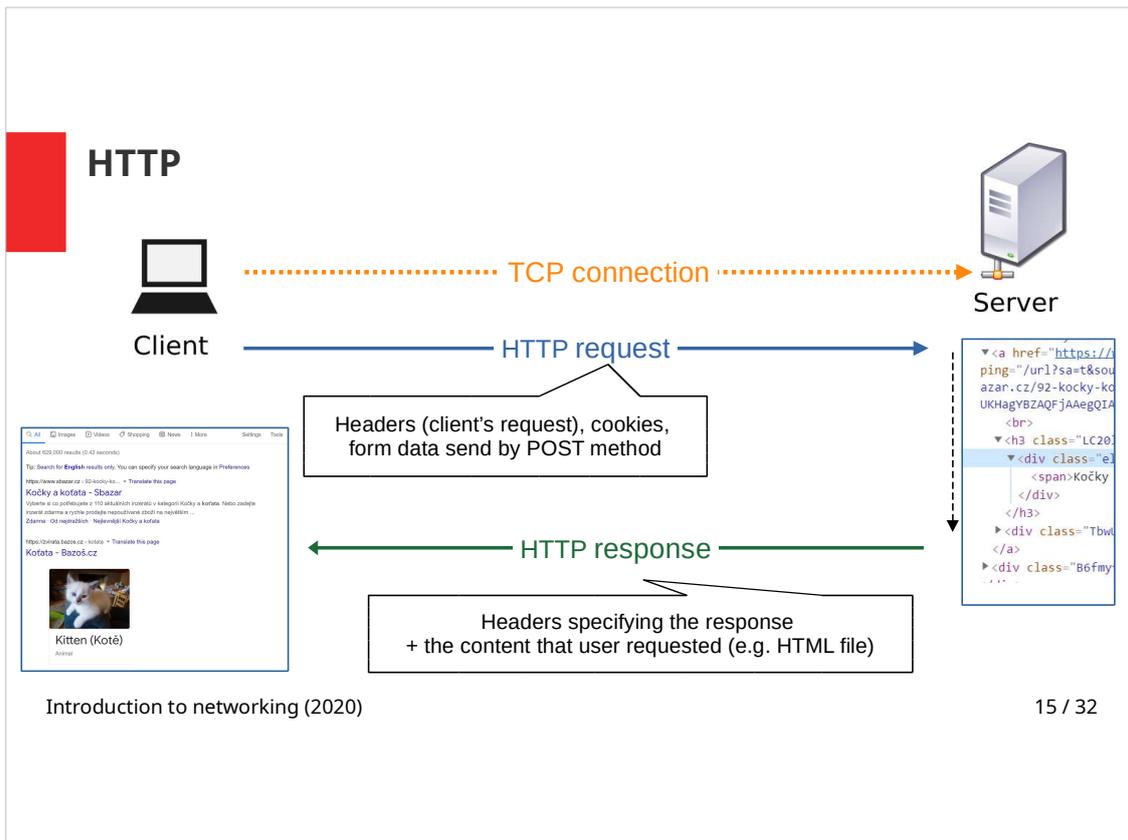
- **Protocol:** mandatory, although the nowadays browsers usually try to hide it
- **username:password@:** this part of URL is rarely used, it can be used for HTTP authentication
- **Domain:** mandatory
- **Port:** in most of the cases there is no need to set port explicitly, the value is set according to the protocol used
- **Path:** almost always used, it refers to the server directory hierarchy (unless URL masking is used)
- **Parameter-value pairs:** are used to set parameters of the server-side script in case that the web page is dynamically generated
- **Element_id:** specifies a HTML element that is at the top of the displayed part of page (the page can automatically scroll down to a specified element)

HTTP – Hyper-text transfer protocol

- Simple text-based protocol
- Uses TCP channel
- Designed for data retrieval
 - Originally only plain text data
 - Extended to support any data type and encoding (MIME)
- HTTP communication protocol
 - user (client) sends a HTTP Request
 - Specifying the details of the requested content
 - server replies with HTTP Response
 - (usually) containing the requested data

MIME (Multipurpose Internet Mail Extensions) – standard that extends the format of plain-text messages to support other character sets than ASCII and transfer of binary data. Standard was originally designed for electronic mail but is used by other communication protocols, including HTTP.

HTTP Response – contains the requested data if no errors occur.



- First, a TCP connection between client and server must be established.
- Client sends HTTP Request to a server
 - Request Headers specify what exactly is requested, more data can be included in the Request Body (Cookies, POSTed data).
- Server loads or creates requested data.
- The result is sent to the client as a HTTP Response
 - Example of a Response content is a HTML file.
- Once a TCP connection is established, more HTTP Request/Response pairs can be exchanged.

Static web page vs web application

- Static web page – server returns the requested document
 - WWW as [Web of Documents](#)
 - URLs are unique global identifiers of documents
 - Documents are linked by hyperlinks
 - HTML is a format for representing documents published on WWW
- Web applications – server generates the document dynamically based on user's request
 - HTML and CSS – application's UI presentation format

A web page can be either static or dynamic according to a way it was created.

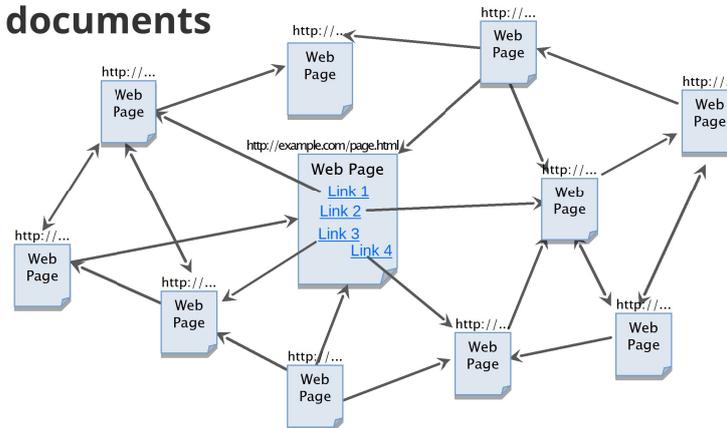
The content of a static web page stays always the same – the author of the web page can update or change the page, but if two (or more) users display the page in the same moment in time they get the same result. Static web page consists of a HTML document that is located on a server along with its multimedia content. Static web page has a unique global identifier that specifies its location on a server (URL).

Web applications interact with their users, they use WWW as an environment where they operate and thus they are accessible from anywhere where there is an internet connection available.

Web applications use HTML and CSS as a user interface – to get input from users and to display the results of the requested actions. In this case, the displayed HTML does not exist in advance, instead it is generated on demand.

Web applications range from simple web sites that dynamically generate the content of their pages after user clicks on a menu item, to very complex ones, that contain number of form controls, like e-banking applications or content management systems. These applications provide same features as a desktop application could, in addition thanks to being displayed in browsers, they are naturally multi-platform and can be accessed from anywhere within the internet. Web applications often use a database running on server to store both data and web pages contents.

Web of documents



- Static web pages – HTML and CSS
- To view a static web page, you can use your browser, server is needed to publish the page on the internet

Static pages can be viewed as a Web of Documents, i.e. static pages with static contents that are interconnected by links. Each page has a unique static address (URL).

Hyperlinks (or links):

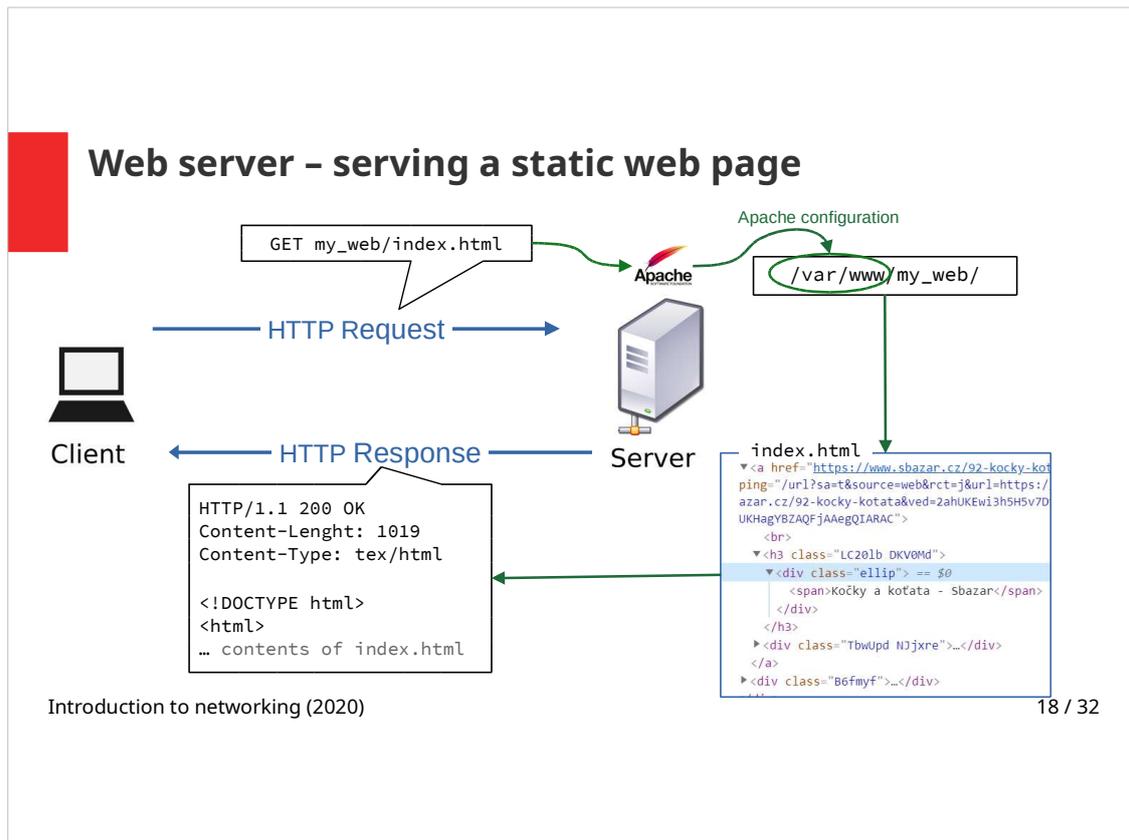
Link is a HTML element that contains URL of a web page. The page can be connected (or linked) with another web page (note that the connection is one-way only, the other web page doesn't have to include link to the first page).

When a user clicks on the link, web browser follows the link and displays the new page, i.e. it sends a HTTP GET Request to a given server. The original page is stored in browser's history so that the user can go "back".

Note:

As static pages use HTML and CSS only, they can be displayed directly in a web browser. This can be useful during development of static web pages and/or while working on assignment for this course.

To publish the page, it is necessary to upload it to a server; however there is no need for the server to run e.g. PHP or database.



In case of a static web page, the web page that is requested by a user is a HTML document that was previously uploaded on a server and is there waiting for users to view it.

- Client sends HTTP Request (GET type) that contains address (path) of the requested page within the given server.
- A web server (software, e.g. open-source HTTP server Apache) running on the server (hardware) receives the HTTP Request.
- A path to the requested document in the Request is relative to the directory that is made accessible from web by a web server.
- The requested document is located on the server and HTML file is retrieved in HTTP Response Body.
- If no file name is specified, i.e. URL ends with / after a directory name, index.html file is displayed by default.

Note:

Term **server** is used for both the hardware (server = a computer) and software that uses HTTP protocol (a web server).

Web application

- Interactive elements
 - Hyperlinks – a way to change an application view
 - Form controls – user input data
 - Can be submitted and processed at the server
- Dynamic web pages
 - Server-side
 - The page is generated on demand
 - Client-side
 - Script running inside a browser can handle user events and modify the loaded HTML document

Web application features user controls, that allow a user to communicate with the application. An example of such controls is HTML form and its elements or hypertext links.

Based on user input, the web page is generated dynamically. It is customized to user's requests.

Web pages can be dynamically generated either on server (server-side) or directly in web browser (client-side). In the first case server generates HTML, that is send to a client as a new page. In case of client-side dynamic web page, there is a script running in a browser. This script modifies a downloaded HTML page due to user's actions without the need to communicate with a server.

HTML Form

- Component of web page composed of UI controls
 - Text fields of various types (input)
 - Select boxes
 - Radio buttons
 - Check boxes
 - File uploads
 - Buttons
- Form components are a user-friendly version of hyperlinks
 - Sends user data and receives new web page
 - HTTP method can be selected (GET, POST)
- Data are sent when a submit button is hit

The image displays several examples of HTML form components. At the top right, there is a login form with two text input fields: 'Jméno: Klára' and 'Heslo:', followed by an 'Odeslat' button. Below this is a search bar with a magnifying glass icon and the text 'Search..'. In the center, there is another text input field with an 'Odeslat' button. At the bottom, a validation message is shown: a yellow warning icon followed by the text 'Please fill out this field.' To the right of the main text, there is a preview of a form with the label 'Jméno' and two input fields containing 'Klára' and 'Příjmení'.

Web forms provide a way for a user to communicate with a web application. By entering data into a form, user provides input for the application and can set its parameters.

Forms are composed of various HTML elements, their appearance can be modified by CSS styles.

Form data are sent to a server all at once, once a submit button is hit. Data are sent as parameter-value pairs either in URL (when GET method is used) – in this case they are visible to everyone – or as a HTTP Request Body (POST method).

Web server – serving dynamic web page

- Common Gateway Interface (CGI)
 - One of the first standards for generating dynamic web content
 - Interface specification for web servers to execute programs like console applications that generate web pages dynamically
 - Interface specification only
 - The console application may be written in any programming language
 - Important information and headers are set as environment variables, POST body is directed to std. input
 - Response is taken from the std. output

Generating a dynamic web page is based on CGI concept. CGI is an interface specification for web servers (meaning software) to execute programs like console applications. User requests are delegated to an application that processes the request and provides the server with the dynamically generated, now static, web page.

CGI specifies only a user interface, command-line program can use any programming language to process the request and generate HTML.

CGI was replaced by more effective solutions like FastCGI or integrated scripting modules (see below).

Web server – serving dynamic web page – three approaches

- FastCGI
 - CGI processes are already started and waiting for requests
 - More effective interface
- Scripting language module
 - A special module is integrated into HTTP server
 - The module interprets script files and serves their outputs
- Application with embedded HTTP server (as a library)
 - Dedicated web server designed for the specific application

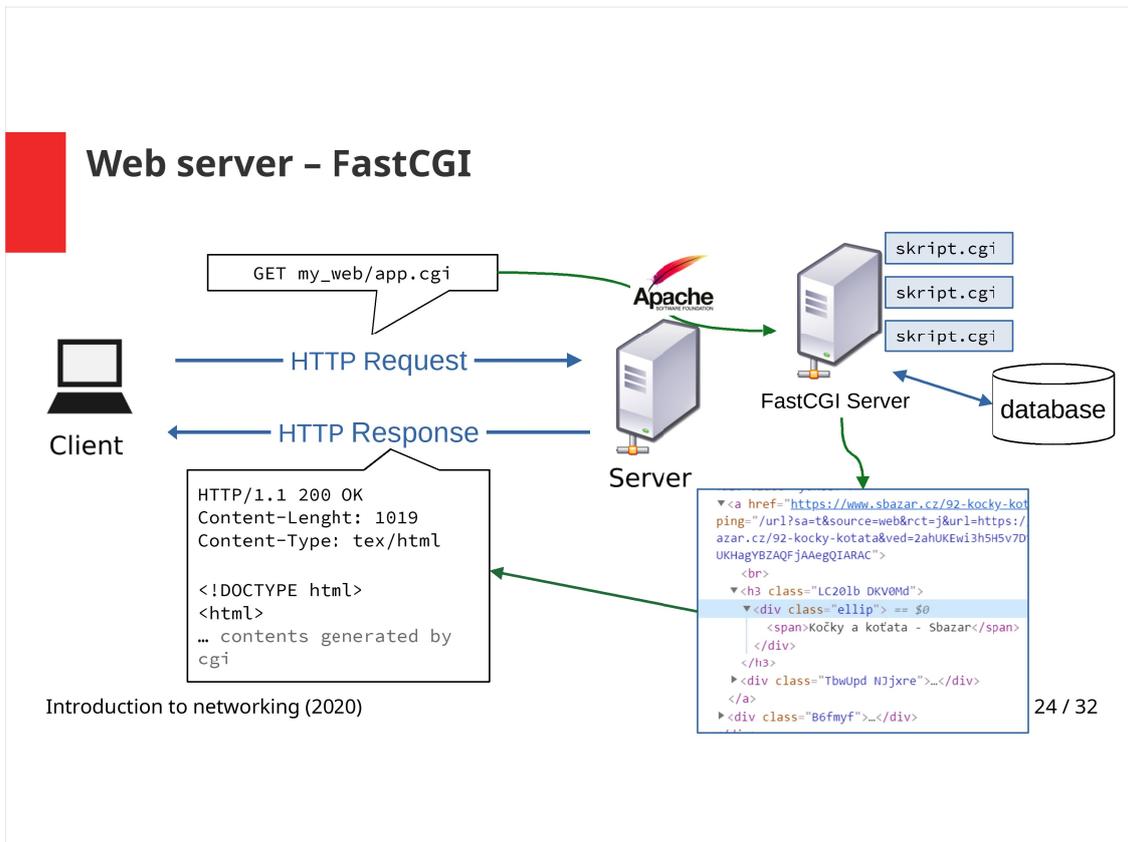
There are three approaches to the server-side dynamic web pages:

1) FastCGI – improved CGI that reduces the overhead related to interfacing between web server and CGI programs, allowing a server to handle more web page requests per unit of time.

2) Use a programming language (PHP, Python) as a module integrated into HTTP server (Apache).

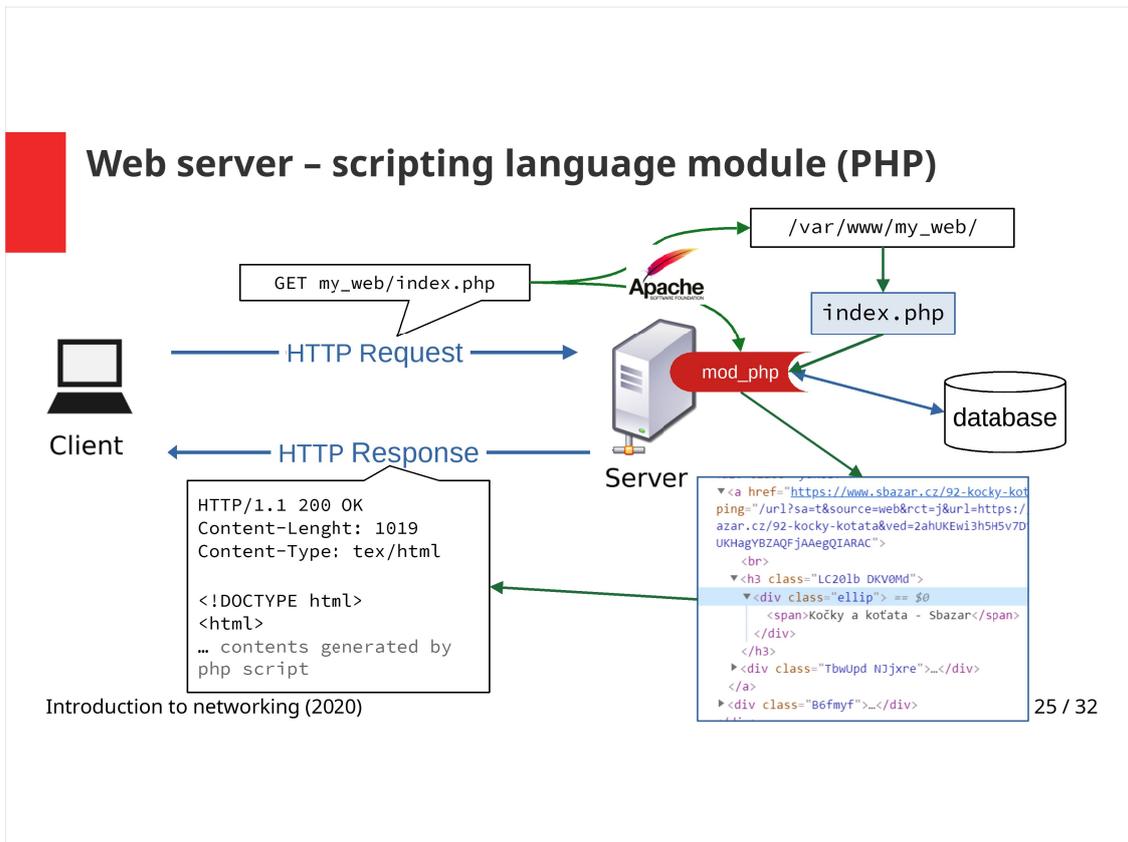
3) Implement your own customized web server, that is designed for your application. Example of such solution can be a server for live streaming Wowza.

These three approaches will be described in the following slides with an emphasis to the difference between the approach at hand and serving static web pages.



On a hardware server there is a FastCGI server running aside from a web server. FastCGI server handles requests for dynamic web pages. Separating web applications from the web server has many advantages, e.g. pre-application security, web server and application processes can be restarted independently. Different types of incoming requests can be distributed to specific FastCGI servers which have been equipped to handle those types of requests efficiently.

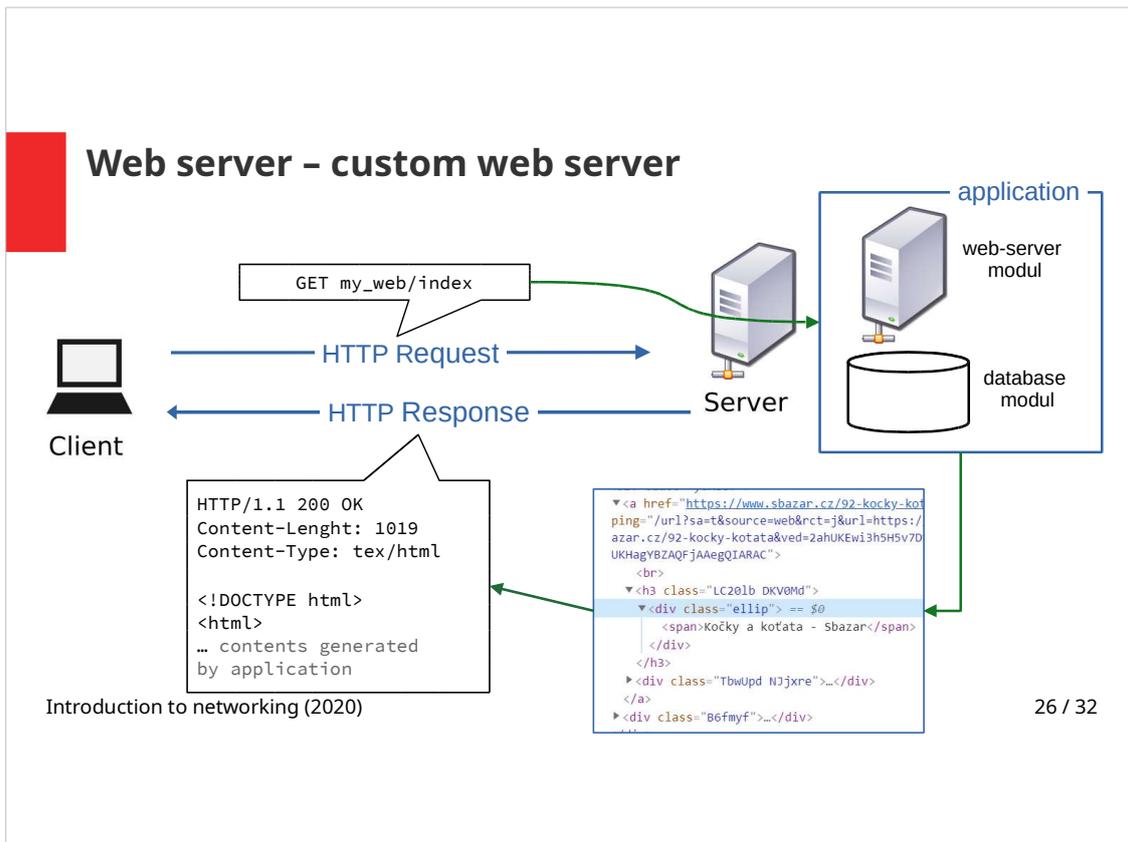
- A web server sends HTTP Request to a FastCGI server, where CGI scripts take care of dealing with the request.



In this approach, scripting language module is a part of a web server, e.g. `mod_php`, PHP language interpreter is a module of Apache web server. This integration enables faster communication between web server and a program that processes the dynamic page request.

- After a HTTP Request is received, the `.php` file is located on the server.
- This PHP file is interpreted by `mod_php` module that is a part of the web server.
- Module generates HTML source code. When the database is involved, it is the module that connects to the database.

If no file name is specified, i.e. URL ends with `/` after a directory name, `index.php` file is displayed by default.



Complex web applications can have such specific requirements on a web server, that it is worth to develop a custom web server, that is designed for the specific application. HTTP server is a part of this application.

An example of such solution is a server for live streaming, that uses dynamic web pages as user interface to set parameters of live streams.

Client – dynamic page

- JavaScript
 - Developed by Brendan Eich in Netscape as a scripting language for web browser (early '90s)
 - c-like syntax
 - Dynamic, weakly typed
 - Object-oriented with prototyping and functional aspects
 - Named after Java for marketing reasons

Dynamic web pages can be generated client-side – in a web browser. In early '90s, JavaScript was designed to serve this purpose.

JavaScript is a programming language that can run in other environments apart from a web browser. It is used as a scripting language for Open Office, Adobe supports running JavaScript in .pdf files. JavaScript can also run server-side, e.g. its Node.js or Rhinola implementations.

JavaScript is not Java, it is not even similar to Java, on the contrary, the two languages are distinct and differ greatly in design.

Client - dynamic page

- JavaScript in a browser
 - Sandboxed, limited interactions with user
 - DOM (Document Object Model) API
 - Allows manipulating loaded HTML document and CSS
 - Modifications are immediately visualized
 - Event model for handling user actions
 - Other APIs
 - Browser window control, history, location (URL)
 - Animations, 3D graphics, sounds
 - Data storage (inside browser)
 - Networking (HTTP requests, WebRTC)

JavaScript has access to an object model of HTML page that is loaded into browser. The web page responds to user's actions without interacting with a server.

An example of dynamic changes in a loaded web page can be an alert window, HTML form validation or a pop-up window with a full sized image after clicking on its thumbnail.

JavaScript can change the structure of HTML, attributes of HTML elements and their content. It is also possible to change the design of HTML elements and their layout within the page by changing their CSS properties. JavaScript has control over browser window, it can follow a link, work with animations and more.

JavaScript can also send HTTP Requests, or use WebRTC (Web Real-Time Communication) – the API that provides support for phone calls, video calls, or video players.

AJAX (/ˈeɪdʒæks/) – Asynchronous JavaScript and XML

- A technique that combines three technologies
 - JavaScript
 - Asynchronous HTTP client API integrated in browser
 - XML or other semi-structured data format
- Script invokes HTTP transfer – without the need to refresh the web page
- Requests are asynchronous, response data are processed using callbacks

AJAX is a way of combining several technologies to get data from server without the need of reloading the whole page.

The page changes while a user is browsing it, without his or her contribution and often even without his or her knowledge. As one can imagine this can cause troubles and/or expose the user to security risks.

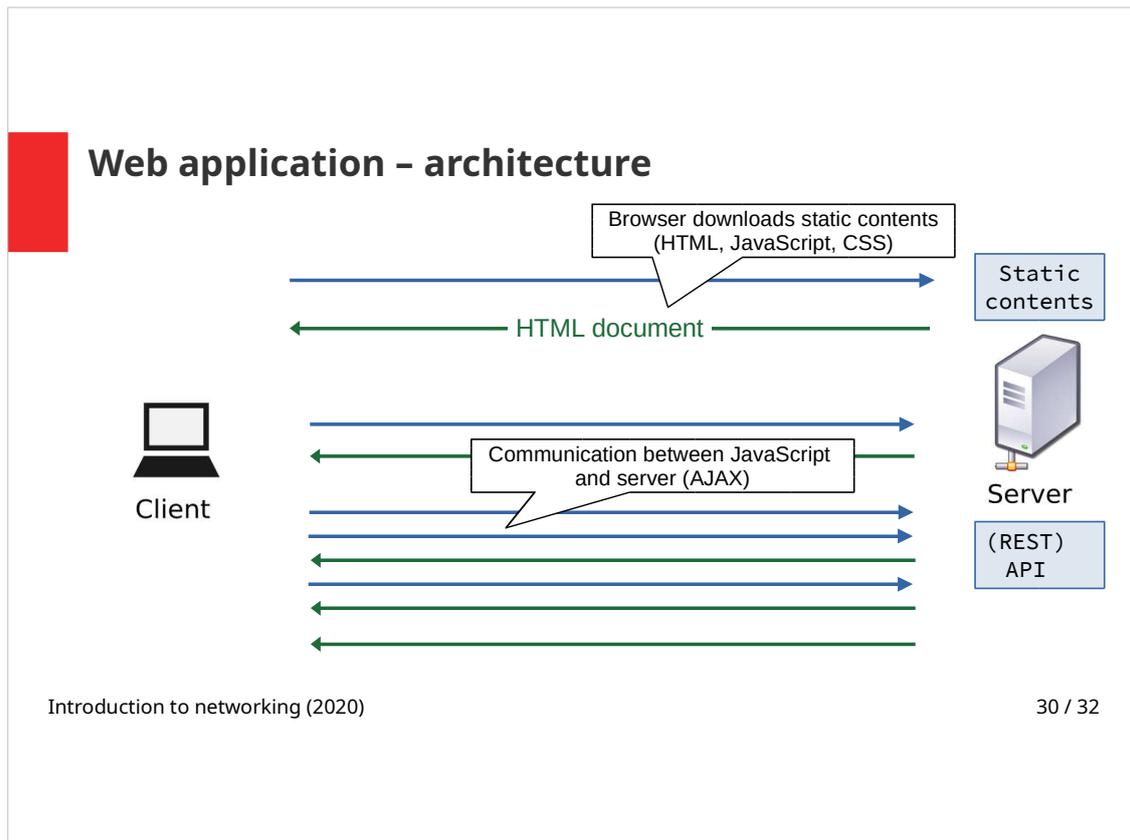
Asynchronous requests

Have you visited a page that at first sight looks like an empty page, but after a while the contents are being loaded? (E.g. there seem to be 0 messages at first, but they are loaded after a while.) This is caused by JavaScript's asynchronous request to get the data from server.

With synchronous operations a program is waiting for the action to finish, then it continues to execute its source code. On the outside it may seem that the application (or its user interface) is “frozen”.

When asynchronous requests are sent, the program doesn't wait for the result, it continues in its execution. In the meantime, while the request is being dealt with, user can continue to work with the application. Once the result of the asynchronous request is ready, callback takes care of the delivered data.

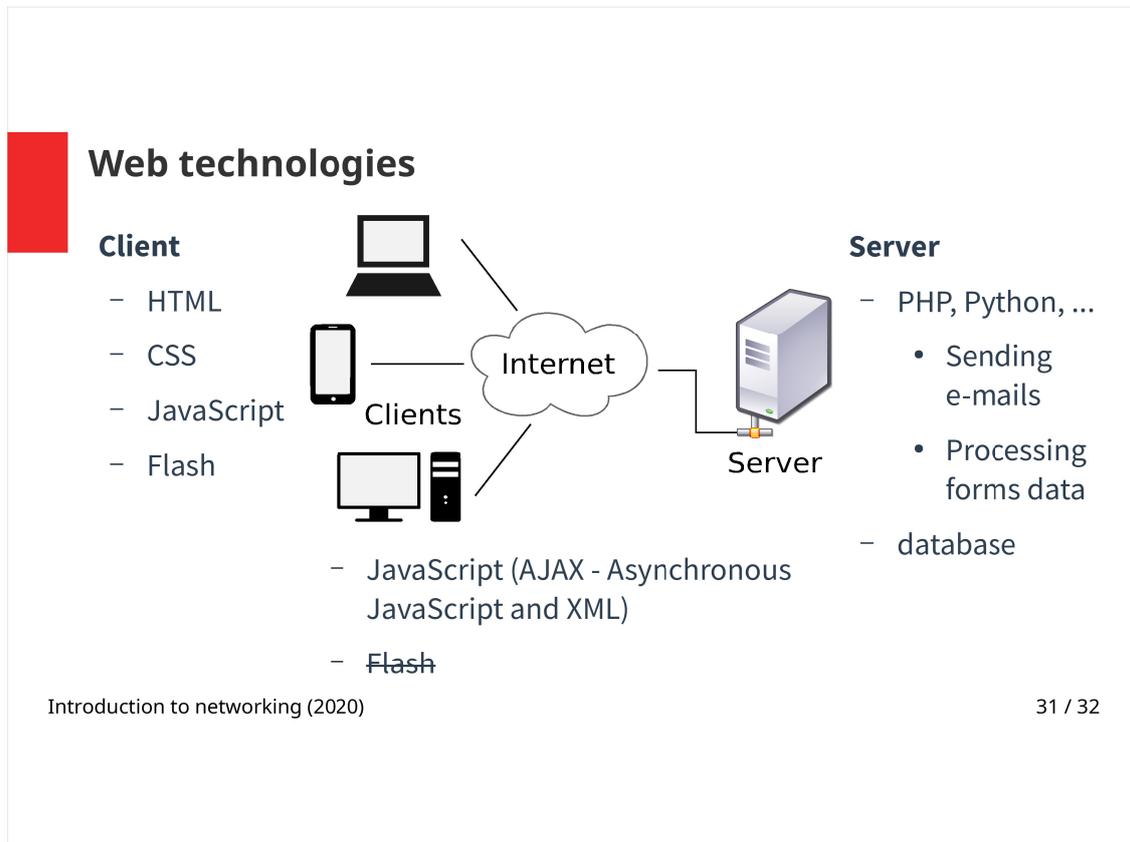
Originally, the data format used was XML (a markup language similar to XML but more general), nowadays most of the applications use JSON format instead.



- At first AJAX application loads the static content.
- Then it uses asynchronous HTTP Requests (using JavaScript functions) to get the data from a server.

Applications often use API for the communication (data requests and responses) with the server. Namely web applications often use REST (REpresentational State Transfer) API.

REST API is designed for client-server architecture. It is state-less, the state of the application is captured in a **resource**. Resource can be represented by XML, HTML, JSON, SVG, PDF, ...



A summary of web technologies that were mentioned in previous slides:

- HTML is a markup language that defines the semantics of web page contents, e.g. heading, paragraph, ... HTML source code is processed by a browser.
- CSS defines the design of single HTML elements and their layout within the page. CSS is also processed by a browser.
- JavaScript – a programming language that can change the displayed web page. It runs in the browser.
- Flash – is not supported any more
- PHP, Python, Java ... – server-side programming languages, that dynamically generate HTML page on user's request
 - They can be used to process HTML forms, send e-mails, ...
 - They can connect to a database
- AJAX runs in a browser, but can request and receive data from server without reloading the web page
- Flash could do that as well



Test yourself

- Explain how a web page gets to your browser.
- What is a URL? What parts does it consist of? What is their meaning?
- Explain a communication between client and server. What protocol do they use to communicate?
- What is the difference between static and dynamic web page?
- What are the two approaches to create a dynamic web page?
- Which web technologies run in a client and which are server-based?
- What is the most important feature of applications using AJAX?