

Interreg 
EUROPEAN UNION

Austria-Czech Republic

European Regional Development Fund



INFORMATICS

Web Application Basics



UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA



EUROPEAN UNION

Contents

| | |
|---|----|
| 1. Communication, Networks, Protocols | 1 |
| 1.1. Communication protocols | 1 |
| 1.2. TCP/IP | 2 |
| 1.2.1. Application Layer | 3 |
| 1.2.2. Physical Layer | 3 |
| 1.2.3. Data Link Layer | 3 |
| 1.2.4. Internet Layer | 4 |
| 1.2.5. Transport Layer | 4 |
| 1.3. ICMP | 5 |
| 1.3.1. Packet Size | 5 |
| 1.4. IPv4 | 6 |
| 1.5. IPv6 | 6 |
| 1.6. ICMP | 7 |
| 1.7. TCP | 7 |
| 1.8. UDP | 7 |
| 1.9. SCTP | 8 |
| 2. Languages for presenting web content | 9 |
| 2.1. Introduction to HTML | 9 |
| 2.2. History of HTML | 9 |
| 2.3. Creation of html pages | 9 |
| 2.4. Terms used in HTML | 9 |
| 2.5. HTML | 11 |
| 2.6. CSS | 11 |
| 2.7. Other applications of CSS | 12 |
| 2.8. Syntaxe | 14 |
| 2.9. CSS styles | 15 |
| 2.10. DHTML | 15 |
| 2.11. CSS styles and classes, identifiers and style | 16 |
| 3. Client-side logic - JavaScript | 17 |

| | | |
|--------|---|----|
| 3.1. | What is JavaScript | 17 |
| 3.1.1. | The language characteristics | 17 |
| 3.1.2. | Explanation of script | 18 |
| 3.1.3. | Script writing | 20 |
| 4. | Web architecture..... | 21 |
| 4.1. | Web applications programming..... | 22 |
| 4.2. | Technology of our web applications | 24 |
| 4.3. | What is web architecture | 25 |
| 4.4. | How to create a perfect architecture | 26 |
| 5. | PHP /basics/ | 29 |
| 6. | Processing http request..... | 33 |
| 6.1. | Request methods | 33 |
| 6.1.1. | Request headers..... | 34 |
| 6.1.2. | Response headers..... | 34 |
| 6.2. | Basic features of HTTP protocol..... | 35 |
| 6.3. | HTTP protocol request format | 36 |
| 7. | HTTP protocol request methods | 37 |
| 7.1. | GET method | 37 |
| 7.2. | POST method..... | 37 |
| 7.3. | Using data sources..... | 39 |
| 7.4. | What is ODBC..... | 39 |
| 8. | Data sources | 41 |
| 9. | Database approach..... | 44 |
| 10. | Structured and unstructured data..... | 47 |
| 10.1. | Structured data..... | 47 |
| 10.2. | Unstructured data | 47 |
| 10.3. | Data volumes - structured and unstructured data | 48 |
| 10.4. | Technological characteristics of data warehouses..... | 49 |
| 10.5. | Logical structure of data warehouse | 49 |
| 11. | AJAX | 51 |
| 12. | Frameworks | 55 |

12.1. What is a framework? 57

13. 10 Best PHP frameworks for developers..... 58

I. COMMUNICATION, NETWORKS, PROTOCOLS

I.I. Communication protocols

In the context of computer networks, we often see communication protocols. They precisely define the way of communication to perform a specific function across all levels. There are protocols for sending data, creating secure channels, and searching for IP addresses corresponding to the domain name, or delivering emails, etc. The protocol is known to both communicating parties and precisely describes what content, what sequence and what timing is used for transmission. Any deviation from this structured communication may be interpreted as an error.

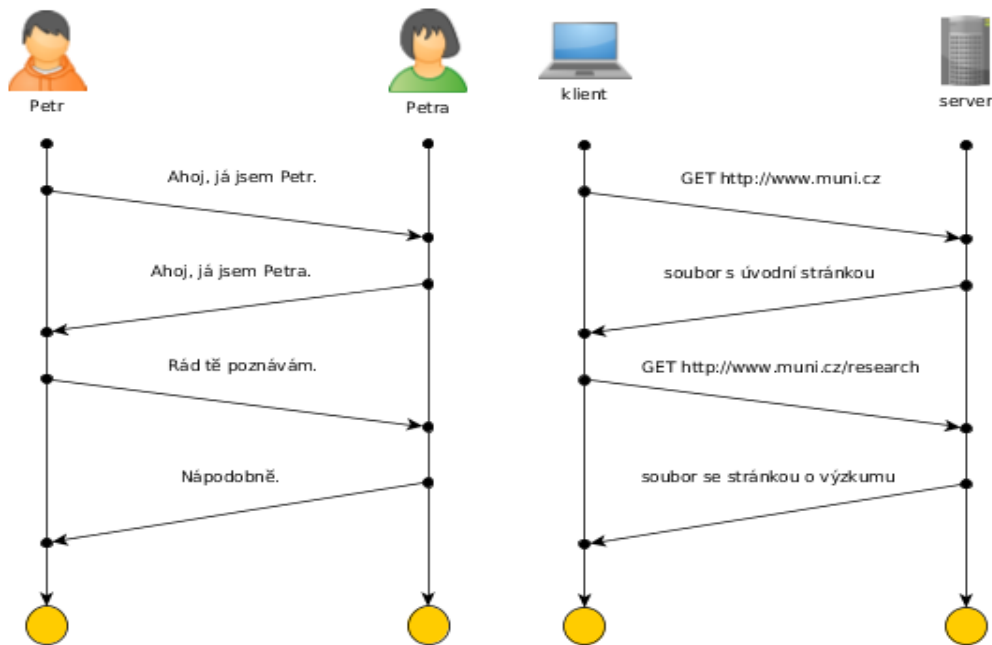


Fig. 1: Communication protocol illustration

1.2. TCP/IP

The essential principle of the computer network architecture is to divide communications into layers by abstraction. Each layer is responsible to describe the transmission starting the application level through physical data link communications. The TCP/IP network model is the cornerstone of all present networks as well as the entire Internet. The name is derived from two major protocols ensuring data routing and transport between nodes. The IP protocol describes the addressing of nodes, data broken down into packets and their routing inside the network.

The communication between the same layers of two different systems is controlled by the communication protocol using a data link created by the adjacent lower layer. The architecture allows to exchange protocols of one layer without impact on the others. The TCP/IP architecture is broken down into four layers (unlike [the OSI reference model](#) with seven layers):

- Application layer
- Transport layer
- Internet layer
- Network interface

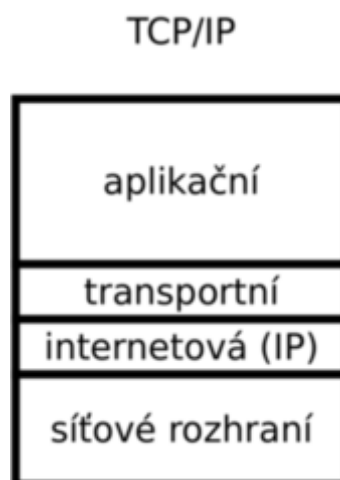


Fig. 2: TCP/IP network model layers

TCP is a transport protocol building a “linked” service on IP transmission: it ensures the control of successful transmission and potential forwarding of missing or damaged parts and selects an optimum transmission speed.

The TCP/IP protocol family which is currently used for the absolute majority of network communications was designed between 70s and 80s with the aim to create a more robust model of network communication, to a certain extent able to cope with losses of parts of

the network. The key (and quite revolutionary at that time) idea is *packet switching*: data are not transmitted as a continuous stream, but in separate blocks (packets) and each network node alone determines which way to send/forward packets.

In practice, this idea is based on a set of protocols implementing the necessary functions. Protocols are typically broken down into several levels (layers). Instead of the abstract ISO/OSI model working with seven layers, a simplified five-layer model is mostly used for TCP/IP interpretation (some protocols in the TCP/IP model implement the function of multiple layers of the ISO/OSI model).

1.2.1. Application Layer

The application layer is the upper layer which refers to application protocol data of the individual network services. It includes network application protocols: electronic mail, HTTP (websites), DNS (domain service) and others. There is a huge number of such protocols, for example HTTP, SMTP, FTP, NTP are the most commonly known. For TCP/IP those are data which must be transmitted to a target recipient. The implementation is up to lower layers.

1.2.2. Physical Layer

The physical layer is the lowest layer in the model. Unlike the upper layers it is not a software layer (protocol) and refers to a specific physical medium used for data transmission. An example can be a twisted pair, cabling in the majority of local Ethernet networks, coaxial cable, optical fibre, or a telephone line. The medium does not need to be material – e.g. in case of wireless networks in the microwave band (Wi-Fi, breezenet) or free-space optical links (FSO).

1.2.3. Data Link Layer

The data link layer is the lowest of software layers. It is the lowest communication protocol used for data transmission on a physical medium. This protocol is most often closely connected with a specific medium, but this correspondence does not need to be 1:1, e.g. Ethernet is not only implemented on twisted-pair cabling, but you can see implementations using coaxial cabling or optical fibres. Another example of a data link layer protocol is PPP which is used for implementation of dial-up connection or computer serial links. The important feature of data link layer protocols is that they only deal with communication between nodes that are *directly* linked (hence the name).

1.2.4. Internet Layer

The *Internet layer* is responsible for global addressing and routing, in practice most often implemented by an IP protocol (in two versions IPv4 and IPv6). Although there exist addresses in link layer protocols and in case of Ethernet MAC addresses, they are (or, at least should be) globally unique, they cannot be used for packet routing because they do not show the target. But IP protocol addresses (IP addresses) are hierarchical so that delegation of individual ranges reflects the network topology. Therefore, you can tell from the target IP address where to further send a packet, that is at least the following hop on the route. In addition to this basic function, IP protocol allows, for example, fragmentation (chopping too big packets into multiple smaller datagrams) or packet designation by type of service (ToS).

1.2.5. Transport Layer

UDP and TCP are the most frequently used protocols of the transport layer. UDP (User Datagram Protocol) is a minimal message-oriented transport layer protocol, only introducing the concept of port as a specific process address (better to say a socket) within the target node. But UDP is still a stateless protocol (not a connection in the real sense of the word), not dealing with the issue of lost packets or their sequence. Still it is often used because it is simple and low overhead, in particular where those issues do not need to be addressed.

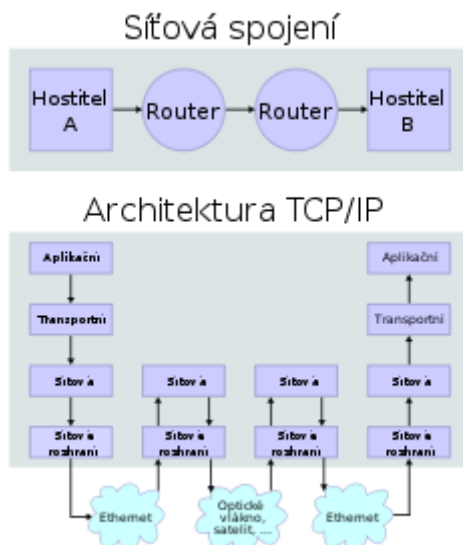
The opposite approach is represented by TCP (Transmission Control Protocol). Unlike UDP it establishes connections between two ports on the end nodes. In the view of client applications, the behaviour of this connection is similar to a communication pipe between two processes (but unlike the pipe TCP is a two-way connection). It is guaranteed that a stream (byte sequence) dispatched from one side is received in the same format on the other side. TCP protocol provides for detection and resending of lost data as well as data rearrangement from packets arriving in the wrong sequence. Since TCP transport protocols ensure fairly high levels of comfort for the application layer, they are currently used for the majority of communication. The disadvantage is a higher overhead and relatively low response to outages, therefore UDP is preferred for some specific purposes (e.g. most DNS queries or VoIP).

1.3. ICMP

ICMP (Internet Control Message Protocol) goes a little beyond the layer structure. Packets (message) of this protocol are transferred directly through an IP protocol. However, ICMP is not considered as a transport protocol because it is not used for transmission of application data. This protocol is used for diagnostics and service purposes. The examples of ICMP applications are packet destination unreachable messages, ICMP echo and echo reply packets, or some service types of messages (redirect).

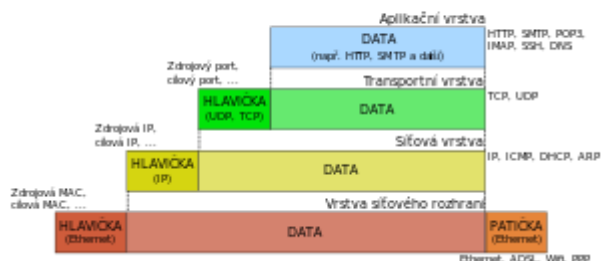
1.3.1. Packet Size

The maximum (theoretical) size of IP packet is 65535 B, but the data link layer is often a limiting factor. Since most of packets go through the Ethernet (or, its equivalent) at least once, the size of packets is often selected according to its limit (1536 B), hence the most typical value is 1500 B. But of course, this is the maximum value and packets are very often much smaller, in particular for interactive applications. The size of IP and TCP headers is 20-60 B (typically near the lower limit), UDP and ICMP headers are 8 B, and Ethernet header is 14 B (in addition 2 B at the end of packet is the checksum).



The TCP/IP layers providing data transfer between two hosts through two routers.

ZAPOUZDŘENÍ DAT V SÍTI TCP/IP



Overview of application data encapsulation on TCP/IP layers.

In the view of problem complexity, the network communications are divided into so-called layers representing the hierarchy of activities. The exchange of information between layers is precisely defined. Every layer uses services of lower layers and provides services to upper layers.

Internet Protocol is the basic protocol of the Internet layer and the entire Internet. It generates datagrams based on network IP addresses contained in their headers. It provides the connectionless network service to upper layers.

Currently, the most widely used is the IP protocol version 4. The new version 6 which addresses the deficiency of addresses in IPv4 and security problems and enhances other properties of the IP protocol is only used by several percent of devices connected to the Internet worldwide, but the number is rapidly rising.

1.4. IPv4

Internet protocol version 4

- 32 bit addresses
- App. 4 billion of IP addresses, insufficient now
- Format: xxx.xxx.xxx.xxx where xxx is an arbitrary number 0 - 255 (8 bits)

1.5. IPv6

Internet protocol version 6

- 128 bit addresses
- Security support
- Mobile device support
- QoS - Quality of Service function
- Packet fragmentation - distribution
- Not compatible with IPv4

The Address Resolution Protocol is used to search for a MAC physical address by a known IP address. If needed, the protocol will send a datagram with the searched IP address information and address it to all stations in the network. The node with the searched address will respond and fill in its MAC address. If the searched node is not in the same segment, the relevant router will respond and return its address.

1.6. ICMP

The Internet Control Message Protocol is used for transmission of **control messages** related to error statuses and special conditions of transmission. It is used, for example, for computer availability testing in the *ping* program, or for packet route tracing to another node by the *traceroute* program.

1.7. TCP

The Transmission Control Protocol creates a virtual circuit between end applications, i.e. **reliable data transmission**. The protocol features are:

- Reliable transport service delivering all data to the addressee without any loss and in the correct sequence.
- Connection service with phases of connection establishment, data transmission and connection termination.
- Transparent transmission of arbitrary data.
- Fully duplex connection, parallel two-way data transmission.
- Distinguishing between applications using ports.

1.8. UDP

The User Datagram Protocol provides a non-reliable transport service for applications which do not need reliability of TCP protocol. It lacks the connection establishment and termination phase and already the first UDP segment contains application data.

1.9. SCTP

A reliable protocol for transmission of datagrams in multiple streams. It is primarily used for the purpose of telecommunications. It has some additional features that TCP does not provide:

- Multihoming – the communicating node can have several IP addresses.
- Breaking down the data stream into datagrams.
- Using multiple data streams – to reduce blocking communication due to a missing data block which may happen in TCP.
- Route selection and tracing – Using an alternative if the primary address has availability problems.

2. LANGUAGES FOR PRESENTING WEB CONTENT

2.1. Introduction to HTML

HTML is a simple markup language for creation of websites, which are just common text files containing a text and several html tags that determine the meaning and appearance of the individual website parts.

Besides html, there are several other languages used for creation of www pages: [css](#), [php](#), [javascript](#), etc.

HTML is a basis in case you want to learn something about [css styles](#) (css styles determine the appearance of web pages) and later about [php \(related to programming\)](#).

2.2. History of HTML

Web appeared in 1989; since then it has been continuously developed. Currently, HTML 5 is being developed.

2.3. Creation of html pages

We can use a simple text editor, notepad or special editors with html code support. These programmes highlight syntax and allow us to switch between a code and preview. Working with such an editor is much more efficient than working with a common notepad. .

2.4. Terms used in HTML.

Tag – is a basic html mark, written as <tag>.

Attribute – is written within a tag and sets its property. It is written as: <tag attribute="value">.

Element – Writing of heading: <h1>Page heading</h1>.

First page.

HTML pages are just common text files with tags

Tags

Tags are written between brackets < >, some of them are paired, some are not.

Writing a non-paired tag:

```
<tag>
```

Writing a paired tag:

```
<tag>some text</tag>
```

In the case of a paired tag, it is important to write a slash, otherwise the explorer would not understand.

Attributes

Attributes are written straight into the tag.

```
<tag attribute="value">
<tag attribute="value">Paired tag with attribute.</tag>
```

It is prohibited to cross tags

Tags can be within other tags, however they cannot cross each other partly.

```
<b><i>bold italic</b></i>
```

Correct writing:

```
<b><i>bold italic</i></b>
```

Size of characters

In html the size of characters does not matter, it is therefore possible to write both **<TAG>** or **<tag>**, whereas in xhtml, which is the latest version of html, neither tags nor attributes can be written in capital characters.

In **url** it is necessary to respect the size of characters, that is, FILE.html is not the same as file.html.

2.5. HTML

HTML stands for Hypertext Markup Language. Hypertext markup language was developed from SGML and became the most widely-used language for creating web pages. In the past, the versions most often used were HTML 2.0, HTML 3.2, HTML 4.01 and HTML 5. From HTML was developed also [XHTML](#) (extended hypertext markup language) as an application of XML, which is not of higher significance, in my opinion. In 2010, HTML 5 started to be considered; most of its innovations have already been used.

Structure of a html file

The most frequently used "template" of a page:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="windows-1250">
    <title>Name</title>
  </head>
  <body>

page text

  </body>
</html>
```

2.6. CSS

History of CSS

CSS appeared around 1997. It is a collection of methods for graphical design of web pages. The abbreviation stands for Cascading Style Sheets. They are called cascading, since definitions of a style can be layered, but only the last one is valid. However, this is not important now.

There has also been designed CSS 2, improved and more sophisticated forms of styles; however, they do not work well in the most widely used Internet Explorer.

When to use CSS

In 2015 it could be said that the whole web is formatted using CSS. From the previously used HTML, only boldness and italics were still used. Therefore it is good to have some knowledge of CSS if you want to create web pages. First of all it is necessary to know how HTML works. Without any basic knowledge, it is not recommendable to start working with CSS. CSS shall be studied if:

- you want to have formatted websites - colours, justify the text, columns layout, etc.
- you often write texts for the Internet without losing time formatting,
- you deal with script, especially [JavaScript](#)
- administrate (or plan) web with many pages that are supposed to have a similar design,

2.7. Other applications of CSS

There are three possibilities of using CSS

The style can be declared in three ways (see the examples below). It is enough to know at least one of the following methods:

In the source text for a formatted element by means of the attribute `style="..."`. This is referred to as **direct style**. It is quite unsuitable.

By means of "**style sheet**" in the heading of the page. Style sheet is a list of styles, giving information about formatting (e.g. green headlines). In a page, a style sheet is written between tags `<style>` and `</style>`.

By means of external style sheet -- it is a **file *.css**, that the page is linked to with a tag `<link>`. The file contains a style sheet. The main advantage is that many pages can be linked to one file and all the pages have similar design.

Examples

Make a paragraph in red font using CSS. Three possible ways can be used:

Direct style

In the source, this paragraph declaration is written:

```
<p style="color: red">This paragraph will be red.</p>
```

Explanation: `<p>` is a mark designating a paragraph. Attribute "style" is a general attribute that could be used for each element.

Style sheet

In document heading, a style sheet is written between tags `<style></style>`:

```
<style>
  p {color: red}
</style>
```

Paragraphs are written in the body of the page:

```
<p>This paragraph will be red. </p>
<p>This paragraph will also be red, because all paragraphs will
be red.</p>
```

If we want only some paragraphs to be red, use ["classes" and "identifiers"](#).

By external CSS file

There will be a file created, named e.g. `styly.css`. It will contain only the following text:

```
p {color: red}
```

In the heading of a html document we want to change by style, the following link must be written:

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

In the body of the document, all paragraphs will be red.

2.8. Syntaxe

As you may have noticed, CSS are not a part of HTML; therefore they are written in a different way. If the table below appears to be too theoretical, please pay attention only to the examples in the lower part.

| | |
|-----------------------------|--|
| Direct style: | <code><tag style="characteristics">styled element</tag></code> |
| Style sheet: | <code><style> tag {characteristics 2nd tag {characteristics} </style></code> |
| Characteristics simplified: | <code>characteristics: value; 2nd characteristics: 2nd value</code> |
| characteristics in general: | <code>characteristics: value [, value2] [; another characteristics]</code> |

Examples:

| | |
|------------------------------------|---|
| Direct style | <code><p style="color: red;">text of red paragraph</p></code> |
| Style sheet | <code><style> p {color: red} body {background-color: yellow;} </style></code> |
| Simple writing of characteristics | <code>color: red</code> |
| complex writing of characteristics | <code>font-family: Arial, Arial CE, sans-serif; color: red;</code> |

It shall be note where quotation marks, colons, curly brackets, semicolons, and commas are used. Example of the a correct writing:

```
h2 {color: green; background-color: yellow}
```

spaces and line endings do not play a significant, they can be added and left out. The characters size does not play a role. There is a [list of characteristics](#) and their values available.

The explorer ignores the values it does not recognise.

In style sheets, comments are made similarly as in Java between `/*` a `*/`. Two slashes do not work.

Example with a headline

It is quite easy in style sheet or external css file.

```
<style>
  h1 {color: green;}
  h2 {color: blue;}
</style>
```

This way, the entire document will contain green first level headlines and blue second level headlines, however, only supposed that the tags <h1> and <h2> were used for writing a text. In other words, style sheets can be used only in well-structured texts.

2.9. CSS styles

CSS styles are cascading, they are used for creating a style of a web page (colour, font, font size). Using CSS, one file can influence the design of the whole web.

Outdated methods

Before CSS styles, for a web page style, the element , was used, that is no longer used. Compared to CSS, it has the following disadvantages:

- If you often changed the style of a text, this tag appeared in the source code very often, which slowed down the page.
- It enables to change font, colour, and size only

2.10. DHTML

Maybe you have already come across the abbreviation DHTML, or Dynamic HTML. This language is created nearly exclusively from JavaScript, VBScript (language with characteristics similar to JavaScript) and CSS styles. This language uses the advantages of HTML, JavaScript, and CSS, thus creating a perfect design and good-looking pages.

2.II. CSS styles and classes, identifiers and style

CSS - Cascading Style Sheets – cascading styles were first implemented by Microsoft in 1996 into Internet Explorer 3.0. CSS style completely removes and introduces <style>. By means of CSS styles, it is possible to define colour, font, size, and many others (box, underlining, boldness, waviness, display, bullets, margins..)

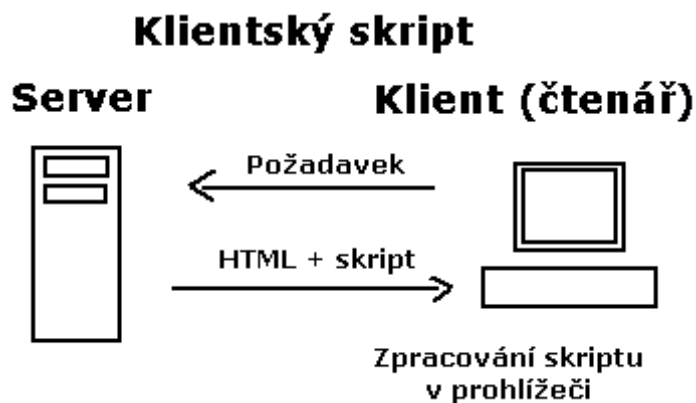
CSS styles are applied mainly by means of classes and identifiers. These enable to create a CSS style by means of one attribute only and the user do not have to repeat one code ten times. In addition, it is also possible to define the style of the elements (h l, table, etc.) by means of selectors. For example, each element <input> will always have a red text – this is possible to make with one CSS rows.

3. CLIENT-SIDE LOGIC - JAVASCRIPT

3.1. What is JavaScript

JavaScript is a programming language used on websites. It is written directly into HTML code, which is a great advantage due to its simplicity.

JavaScript is a client script. This means that the programme is sent with a page to the client (into the explorer) and it is performed there (unlike server scripts, which are performed on a server and the client gets only the results).



Legend: klientský skript – client script, klient (čtenář) – client (reader), požadavek – requirement, zpracování skriptu v prohlížeči – processing script in explorer

There are also other languages of client scripts, e.g. VBScript. However, there are so rarely used, that when you mention “scripts” it mostly means “JavaScripts”.

JavaScript is not Java

JavaScript is often confused with Java. Java is a separate programming language. It only has a syntax similar to JavaScript.

Necessary skills

- [HTML](#), basics of HTML at least
- Basics of programming

3.1.1. The language characteristics

JavaScript is a language

- interpreted – it does not have to be compiled
- objected – it uses the object of an explorer and built-in objects

- dependent on explorer – but works in most explorers
- case-sensitive – the size of characters in the ZÁPIS matters
- its syntax is similar to C, Java, and the like

Limitations of the language

- JavaScript works only within an explorer.
- Users can block JavaScript
- There are various different versions of the language and explorers, which causes frequent errors.
- it cannot access files (except for cookies) or any other system objects.
- It cannot save data (except cookies).

This makes it only a secondary language, applicable only for HTML pages.

How to deal with JavaScript

After mastering the basics, it is recommended to notice scripts on other websites. Most scripts are written directly in the source code of the webpages, so it is possible to copy it (some codes are in external files, but even those can be downloaded).

3.1.2. Explanation of script

Script is written into HTML between the tags `<script>` and `</script>`. Everything written between the tags is a programme written in the language Javascript.

The example uses the command **document.write()**. This enables normal writing into the document stream. The written text is immediately displayed in the explorer.

Normal text must be written between quotation marks (unlike [variable](#)). A line must not be wrapped between the quotation marks.

Each JavaScript command is ended with a semicolon or wrapping a line.

How to create the first script

Everything created within JavaScript is called script. It can be placed in the page or it is possible to create a link to it. In such a case the page uploads in the JavaScript page. Separate files written in JavaScript have the extensions `.js` or `.jse`. The extension `.js` is more common. The only thing you need for creating a script is a source text editor ([PSPad](#), text editor or any HTML editor). For browsing you need an explorer (at least Internet Explorer and Mozilla Firefox so that you can check scripts in these most commonly used explorers).

Inserting a script into a page

Script is written between the tags `<script>` and `</script>`. They can be inserted between the section "body" or "head" (it depends on the purpose of a script).

```
<html>
  <head>
    ...
    <script type="text/javascript">
      .. javascript script body ..
    </script>
    ...
  </head>
  <body>

document body..
  <script type="text/javascript">
    .. javascript script body..
  </script>

document body..
  </body>
</html>
Tag <script>
```

Syntax of the tag `<script>` is the following:

```
<script type="text/javascript" src="url of external file">
  <!--
      javascript script content
  //-->
</script>
```

Attribute `type` designates a type of a script (in the case of JavaScript, "text/javascript"). Since there are explorers that do not understand JavaScript, it is recommended to write `<!--` at the beginning of the script and `-->`, at the end of the script, otherwise the explorer would write the script as a normal text (now, it will consider the text a comment and will not display it).

3.1.3. Script writing

In JavaScript it is important to distinguish between capital and normal letters (it is case-sensitive), therefore `document.write` is not the same as `DOCUMENT.write`. This rule must be observed, otherwise the script will not work.

JavaScript is a multi-platform, object-oriented scripting language, whose author is Brendan Eich from the company Netscape.

Now, JavaScript is usually used as interpreted programming language for WWW pages, often inserted directly into a HTML page code. It usually controls various interactive elements GUI or creates animations and image effects.

JavaScript was originally a trade name of an implementation of Netscape which was originally developed under the name Mocha, later LiveScript. It was announced with the company Sun Microsystems in December 1995 as a complement to the languages HTML and Java. For the version of the company Microsoft, the name JScript is used. This is supported by the platform .NET.

A programme in JavaScript is usually started after downloading a WWW page from the Internet (on client side), unlike other interpreted programming languages (e.g. PHP and ASP), which are started on the side of server even before their download from the Internet. This implies certain safety limitations. For example, JavaScript cannot work with files, otherwise it could threaten the privacy of the user.

JavaScript can be also used on the side of a server. The first server-side implementation of JavaScript was LiveWire of the company Netscape launched in 1996. Nowadays there are more possibilities, including open-source implementation Rhinola based on Rhino, gcj, Node.js and Apache.

Besides DHTML, JavaScript is used for writing of extension for many applications, e.g. Adobe Acrobat.

JavaScript can be also used in the operating systems Windows using the programme Windows Script Host and thus replace batch files MS-DOS.

4. WEB ARCHITECTURE

Web architecture design

We never start to create web with graphical design. Firstly, the objectives of the individual pages are defined and their location on the web is specified.



Why is it important?

Thanks to the web architecture design, it is possible to imagine what the resulting web will look like and what his functions will be.

This way it is possible to identify and correct potential defects, thus saving money it would require to repair programmed applications later.

Web structure design

In the structure design we have to define all web pages, their mutual links and usual user scenarios. It is necessary to make sure that the information is structured appropriately, easy to use and leads to objectives achievements.

Web navigation design

A suitable navigation is an important part of a web. It enables a visitor to find quickly what they are looking for. This shall be paid great attention to when designing a web.

Wireframe modelling

Once we know the content of the web, wireframe modelling can be started, which is the basis for the graphic design of the web.



4.I. Web applications programming

Web application usually means a server-side [script](#) (a code ensuring the programme function). It is often connected with a [database](#), a system storing web applications data (in a simplified way, a database can be imagined as an MS Excel file). The script output is a web page, which is passed to the explorer for display.

Schéma webové aplikace



Legend: schéma webové aplikace - web application graph, skript - script, www stránka - www page, databáze - database

The task of web applications is mostly to increase the interaction of the Internet presentation with its visitors or to facilitate the web administration, i.e. to spare repetitive work in creation of www pages.

Depending on the functionality requirements, a web application can consist of just a few lines of code (e.g. when sending contact forms), but there are also web applications consisting of thousands of lines.

More complicated web applications are often linked to other software within a company, e.g. to ordering systems, accounting programmes, etc. This enables to save costs of human work. It is also possible to connect the application to online payment systems.

Examples of simple web applications

- Contact form
- Guestbook
- Discussions or chat
- Catalogues and price lists
- Dictionaries
- Banner systems

For larger presentations it is better to tackle the whole presentation in a dynamic way, e.g. using templates or deploying the [content management system](#). A specific form of such an application is [internet shop](#). Due to the many advantages they offer, [weblogs](#) are becoming increasingly popular, as well as various [intranets](#) and [extranets](#). All of them belong to special types of web applications.

When programming all web applications, the following aspects shall be taken into account:



- **Safety** – is a priority for any web application, as there is always a risk of losing or destroying of data. There is also a risk that the data or information will be stolen by an unauthorised person or that the web server will be hacked by means of an application (a risk of losing company image).
- **Using available resources** – in programming each web application we try to use finished pieces of codes from other resources; for this purpose we own an extensive scripts archive. This way it is possible to spare some work necessary for the development of the application and thus also money and the total time required for the realization of the order.

- **Scalability** – when the web application is proved to work well in practice, usually it starts to be modified, improved and extended. If these modifications are considered in the application design, their incorporation is much easier and cheaper. For this reason, most complex web applications are tackled in a modular way.
- **Speed** – slow web applications is not very [applicable](#), it is also a problem for [explorers](#). Therefore, we try to optimize all scripts to be fast, and therefore, it is also recommended to deploy a finished application to [our servers](#). Since there are also the latest development tools installed, which are missing in many servers intended for commercial webhosting, the development of the web application is also faster and less expensive.
- **Maximum load** of web application is a term associated with high traffic. If a web is visited by a higher number of visitors (e.g. when you expect a product to be presented), the server is often incapable of serving them (we say that the server has "crashed"). The capability of a web application to withstand the high load requires choosing suitable tools, database optimisation, calculations and other special techniques, such as *pre-caching*. Based on the experience, we can say that the low load resistance is evidence of low capabilities of the programmes who created the web application.
- **Testing** – before starting a web application, all its functions shall be tested on a development server. This has the same configuration as the actual server, which enables a reduction of possible problems during the actual start-up.

4.2. Technology of our web applications

Currently, the world´s most widely used scripting language for web applications programming is [PHP](#) and database [MySQL](#). This combination with a web server (programme) [Apache](#) is called **triad**. It has been proved to be useful for its flexibility. Other advantages of this system include the accessibility of functions and fragments of codes, as well as the continuous development of these programmes.

If there are reasons for it in terms of web applications or simplification of their development, other programming languages are used, e.g. [Perl](#), [Python](#) or database [PostgreSQL](#). Each of them is suitable for the specific needs of a specific web application.

The greatest advantage of all mentioned technologies is their inclusion in [open source](#). It means that they are **free** (which is one of the significant factors of their popularity and widespread).

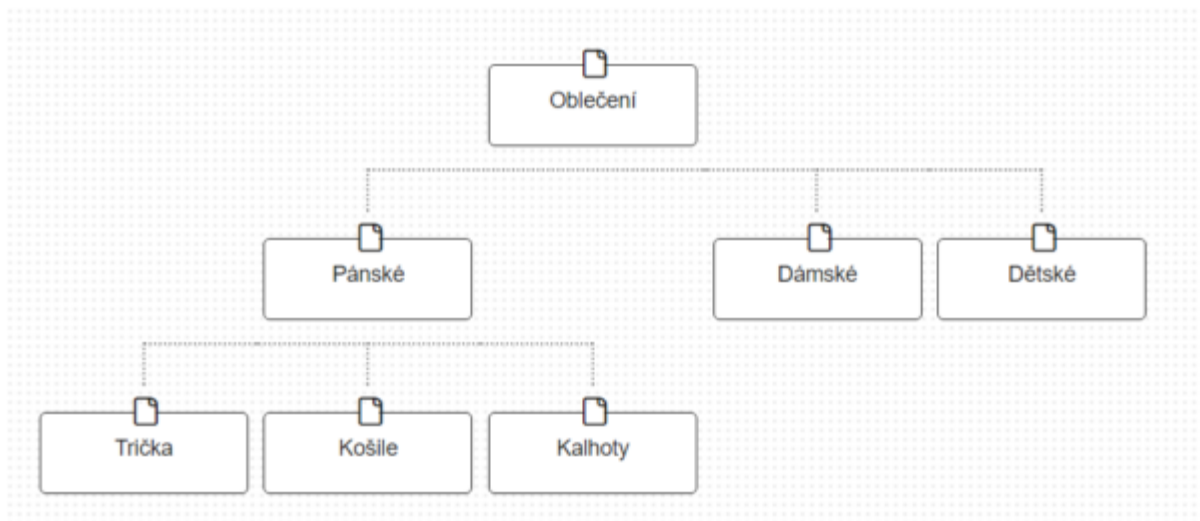
4.3. What is web architecture

Information (content) web architecture is a method of arranging the web information into a logical whole.

It is clear that the information will not be on the same page, but the web will contain more pages arranged in various levels, starting from the general to the details. Everything is thematically linked.

Example: a web selling clothes is arranged as follows: men's clothing > shirts > short sleeve. The way the information is arranged within a web is called web architecture.

For more information, see the figure below:



Legend: oblečení - clothing, pánské - men's, dámské - women's, dětské - children's, trička - T-shirts, košile - shirts, kalhoty - trousers

Why to deal with web architecture

Good web architecture is important for the following reasons:

- It enables a user to achieve its goal (the desired web content).
- It creates a logic of the whole web, perceived and used by both users and explorers and search engines
- It enables to arrange the web content so that it is easy to find in search engines.

When preparing a new web or e-shop, the content (information) architecture of a web shall be dealt within the basic web design (before graphic design and realization of the web - see [4 steps of a professional web design](#)).

Good web architecture enables to arrange properly all the important topics. This is reflected in easier orientation as well as in better traceability by specific key works in search engines. As it is already known, traffic of search engines is free and is highly relevant.

4.4. How to create a perfect architecture

Within [MD webdesign](#), web architecture is created based on the keywords analysis. This means to find out what keywords the users look for in Google and Seznam search engines, to analyse them and then to categorize them to get better insight.

If we want to be traceable by a specific keyword, it is recommended to have a web page on the topic of the keyword and the keyword itself.

Keywords analysis enables to find out what the search engines users are interested in. Including these topics (keywords) in web architecture significantly increases the chances to get better positions in searching.

When creating information architecture, the content of the individual pages is also described. Web content is designed on the basis of analysing customers / clients in the form of so-called person ([for more information about persons, see Wikipedia](#)). Thanks to this, it is possible to design a web that would meet the expectations of concrete customers. And we are able to provide information they need.

Creation of web pages: starting with information architecture

Information architecture is a plan for sorting information on the web. For good functioning of web pages, great attention has to be paid to creation of a clear information architecture.

As an architect working on a house design works with a space, light, and shapes, information architect works with information, structure, and priorities.

Generally, a well design information architecture shall be understandable even in the text form. Using colours or arrows does not help if the information web structure is not understandable.

The main reasons for confusing web include: the terminology used, arrangement requiring professional knowledge, illogical arrangement of information.

In order to avoid the problems mentioned above, information architecture design shall include the following steps:

- Identification of target group
- Gathering information
- Grouping of information

- Prioritization
- Creation of information architecture

The individual steps:

1. Identification of a target group

Target group shall be identified before the information architecture is designed, as it changes dramatically the view to the information published.

In the case of public administration sites, the target user is almost everyone; therefore it will be difficult to define the knowledge or habits that most users will share. However, we are able to suppose which knowledge the users will NOT have.

Practice shows that the users accessing the websites from towns and municipalities usually do not know the difference between the terms "town" and "authority". They do not know the terminology and for them it is difficult to understand the organisational structure of an authority. The situation is even worse due to the fact that the same agenda is usually managed by different departments in different towns.

It follows that the internal structure of an authority is not a good example for creation of informational architecture presented to the general public.

2. Gathering information

The second step in designing information structure is to specify which information the visitor will look for on the web. Public authorities are obliged to publish certain information but it is not all information the web shall contain.

Public administration webs usually contain a huge amount of information, ranging from minutes of meetings, decrees, budgets, decisions, regulations to current events in the municipality and list of cultural or sports events. .

A part of this step is also a decision on which information will NOT be published on the web. It is always necessary to consider which information the web operator will be able to update regularly. The problems with outdated information can be avoided when designing the information architecture.

3. Grouping of information

If we know which information the web shall present, thematic units shall be created. These units shall contain all the information gathered in the previous step, without overlaps if possible.

At this moment at least one representative of general public shall be involved in the process. As an office worker you are aware of the structure of the office and the managing

process. For public, this is different; it is not practical to expect a visitor to be aware of the inner structure of the office on a web.

The visitor goes through the web "from the top": they start from a general level and gradually specify their query with further clicks until they find the desired information. However, when designing information architecture, the process is "from the bottom" - information is sorted and categorized into groups and units. This discrepancy can be misleading for a visitor, as they do not know the web content and do not know whether the information IS on the web. Therefore, it is essential to test the understandability of the information structure "from the top".

4. Prioritization

So far, common sense and testing with users was enough. Prioritizing in the information architecture design is, however, connected with analysing. Therefore, it is necessary to have concrete numbers at disposal, e.g. the analysis of the most searched terms on the web. Priorities can also change over time, e.g. in the summer, there are frequent queries about swimming pools etc., while in the winter, the users are interested e.g. in snow conditions.

There are many methods of prioritization: e.g. placing the information at the top of the structure, placing it on the front page or main page of the thematic unit, highlighting by means of contrast, colours, size, etc.

For example, pages focusing on municipal waste shall contain information about the current price and due date of the charge without forcing the visitor to read the pdf. files with a notice. As it is frequently sought information, it can also be highlighted with a larger font.

5. Creating information architecture

When creating information architecture, a suitable tool can be used, but a tool itself is not enough. Information architecture resembles a tree.

5.PHP /BASICS/

PHP is a programming language working on the side of server. PHP enables to store and change website data. The original meaning was Personal Home Page. It was established in 1996 and it has undergone many changes. Nowadays, the abbreviation stands for Hypertext Preprocessor.

PHP language

PHP is one of the most widely used programming languages used for creation of web applications. PHP is a server-side and servers for generating a [HTML/XHTML](#) page code, which is later sent to explorer (unlike client-side [JavaScript](#), which works only after its displaying in the explorer).

The main advantage of PHP is its independence on the platform (Windows, Linux, Unix...). Other advantages include a wide range of applications. PHP can work with files and many different [databases](#), it can generate and edit graphics, send and receive mails, create PDF, support all important internet protocols...

Since PHP has relatively free syntax (the way it is written), it is easy to learn, especially if you have experience with other programming languages. Together with web server [Apache](#) and database [MySQL](#), it creates so-called triad, three programmes most widely used for generating web pages. This brings another advantage of PHP - a huge number of fragments, user-defined functions and ready-made solutions to common problems on the Internet.

Possibilities of PHP

PHP is not difficult to understand, and knowing the basics is enough. It can save, change and delete data. Everything is done within a web server (where there are source codes of web pages). PHP script is first executed on the server and it sends only the result into the explorer (it means that it first calculates 300/30 and then it sends number 10 into the explorer). Therefore, in the source code, there is only "10" (unlike JavaScript, which calculates directly in the explorer). Unlike JavaScript and HTML, the PHP source code is not displayed in the explorer.

PHP can be used for creating a discussion forum, guestbook, counter, opinion poll, graph; using a simple code, it is possible to dispose of the entire content of the web. Moreover, it is possible to connect the pages with databases, e.g. [MySQL](#).

Purpose of PHP

There is at least one function that every web could use. On web pages, certain parts are often repeated: header with links, menu, footer. With PHP, it is easy to create a template for a web where the files with menu and footers can be inserted. This way it is possible to

write menu only once and copy it into other pages. And changing the menu is very easy (see [PHP menu](#)).

PHP files

Web page with PHP elements has mostly the extension .php, but there can be also other extensions, e.g. .phtml, php3, php4, php5. Some hostings determined which version of PHP script to start based on the extension (current version is 7). However, this is an exceptional case; most often, .php is enough.

Installation

PHP is a language that does not work only with a certain version of an explorer (unlike HTML or JavaScript). It must be installed in the computer. The basis is a web server and libraries. To support PHP, it is necessary to install it and configure a server (usually [Apache](#)). It is recommendable to use [PHP Tread for installation of PHP programme](#).

Webhosting with PHP

Not every webhosting includes PHP support. Support for webhosting is an above-standard service for extra charge. However, it is possible to get free webhosting with PHP support (e.g. Webzdarma.cz, PHP 5). When choosing webhosting for PHP pages it is necessary to read carefully what the offer contains.

PHP – basic information

Dynamic pages (i.e. pages first generated by a server and then sent to the client) are nowadays an essential part of any more complex website. The main scripting languages used for creation of such pages include ASP (Active Server Pages) and PHP. In the following part we will deal with introduction to PHP.

What is PHP?

PHP is a server-side scripting language inserted in a common HTML code. What does it mean? Each page containing PHP scripts is taken by the server and all commands listed on the page in PHP, then a clean HTML code is sent to the client (which is a result of the script). Theoretically, the server can look for PHP scripts in all files sent, but mostly it is configured to look for them only in the files with extensions .php, .php3 or .phtml. PHP commands are inserted directly into the HTML code and are separated by tags <? and ?> (or <?php and ?>).

What is PHP good for?

PHP is a very versatile language in which it is relatively easy to programme e.g. a [news server](#) or [virtual shop](#). Data can be stored in common text files or in a database (PHP is compatible with almost all commonly used databases, e.g. MySQL). Processing the data from the forms is very easy, various simple online tests including the visitors success rate

can be easily created; also a quality advertising system can be easily programmed. The strength of PHP is demonstrated by its use on the servers Email.cz, Centrum.cz or Bill-Board.cz

What is necessary to work in PHP?

PHP is inserted into HTML, so any common HTML editor can be used for creating PHP scripts. It is even enough to use Notepad. The most important thing is to be able to place and test the created scripts. For this, PHP support has to be installed in the server (PHP, now in version 4, can be free download from www.php.net). PHP is most efficient as a module of Linux system, but it can be used also with Windows (you only need this information if you are the administrator of your webpage). If you use any of free webhosting offers, PHP support is not very likely, but e.g. the server www.kgb.cz offers free webhosting with PHP support (with some limitations, e.g. it is not possible to use databases, etc.). In the case of paid webhosting, the provider shall give information about PHP - it is either free or for a certain charge. The provider shall also be able to say what extension it is necessary to use for the files containing PHP scripts (as it was said above, mostly these are .php, .php3 or .phtml).

First script in PHP

Here is the first PHP script that writes the current time:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=windows-1250">
    <title>PHP - example 1</title>
  </head>

  <body bgcolor="#FFFFFF" text="#000000">
    <center><font face="Arial CE, Arial" size="5">
      Current time: <?php echo Date („H:i:s”); ?>
    </font></center>
  </body>
</html>
```

Instructions: Save in the server e.g. as a file example1.php and watch it in an explorer. For a script to work, it has to be first interpreted by a server, i.e. it could not be explored off-line from a harddisc.

How does the script work?

It shall be noticed that it is basically a classic HTML page that also contains one PHP command - echo Date („H:i:s“), which is separated from the remaining HTML by tags <?php and ?>. The server first takes the required file example1.php, and seeing its extension is .php, it lets it pass through PHP interpreter and performs all the commands - these are sought between the tags <?php and ?>. It comes across the command echo, which servers for writing into the resulting file. The function Date returns the date and time, in the brackets, there are the parameters determining the format in which the time shall be displayed (hours, minutes, and seconds separated by colons - function Date will be dealt with later). The following semicolon serves to separate the individual commands. It is redundant in this case, as there is only one command, but it is recommended to get used to writing it. Instead of PHP command, the PHP interpreter writes the current time and the server then sends the page to the visitor. If you want to view the source of the script result, you will only see a pure HTML, not PHP. It results that unlike client-side languages (JavaScript), nobody is able to get to your code and thus it cannot be copied.

Summary:

1. PHP commands are inserted in a common HTML code, they are separated by the tags <?php and ?> (or only <? and ?>).
2. To look only for PHP commands, the file has to have the right extension - usually .php, .php3 or .phtml.
3. The visitor will only get a pure HTML code from the server (that is, the code after all PHP commands are performed).

6.PROCESSING HTTP REQUEST

6.1. Request methods

GET

It is the most widely used method. It servers for retrieving an object (html file, image,..)from the server. The answer is "cachable". GET request is therefore accompanied by a number of headers in which it is specified, how old the document is, if it has been changed, etc. GET request usually does not have a body.

POST

Using this method, it is possible to deliver the information from the user to the server within a body (POST is often used for sending bigger data from web forms, for uploading files, etc.).

HEAD

is similar to GET, but the body is not transmitted in the response. This request can be used e.g. to find out if the object really exists (for controlling the links within a page).

PUT/DELETE

creates/deletes the given object from the server. These methods are not used very much in practice.

OPTIONS

It is used for retrieving information about a given context (or "*" for the whole server). The client can find out which requests they can send to the given context.

OPTIONS * HTTP/1.1

Host: www.root.cz

Example of implicit setting of server

TRACE

It is used for tracing the entire request path. In the body of an answer, the client gets the requests of the individual systems through which the request went. This method is used by administrators and web programmes who want to find out why the server is returning an expired document, etc.

Headers

Protocol HTTP version 1.1 defines a large number of headers for requests and responses. There are some of them:

6.1.1. Request headers

Accept*

Headers of this type indicate what the client is able to handle. The server then chooses the most appropriate alternative. The headers include Accept (MIME types of documents, Accept-Charset (character set, very important in the Czech environment), Accept-Encoding (encoding of transmitted data, mostly it is used for choosing a compression), and Accept-Language (language of the document).

Connection

In the protocol HTTP 1.1, parameter "close" is defined, which requires an immediate closing of the connection after the first requested document is transmitted.

Referer

Using this header, the client announces the URL of the page from which the link was generated.

Host

HTTP 1.1 introduces the support of so-called name-based virtual servers. This method enables to operate more virtual servers from one IP address, but the client has to specify the name of the server they want to communicate with.

User-Agent

This header is used to identify the client programme either for statistical purposes or for the purpose of providing different content to different explorers etc.

6.1.2. Response headers

Content*

Headers describing the content (body) of the response. It can contain e.g. content-length, its MD5 digest (Content-MD5), language (Content-Language), type of the document (Content-Type), and other attributes. It shall be noted that those headers are used not only in the responses. If a body also contains the request (e.g. in the case of the POST method), they have to be used as well.

Server

This header is used to identify the server (usually there is its name, version, and sometimes also other information).

Expires

A server can use this information to indicate the expiration of a document. After this time, the client should download a new version.

There is a number of other headers that can be used e.g. for document downloading control ("download only if the document has been modified since...") or for provide the server with a user name and password for accessing the non-public parts of the server. Similarly, a server can describe more accurately its response and inform the client when the document was modified for the last time or if its caching in public or private caches is allowed.

6.2. Basic features of HTTP protocol

To fully understand the information, it is necessary to have a knowledge of basic features of HTTP protocol and its functioning. HTTP protocol is an application-level protocol for distributed hypermedia information systems. In practice this means that this protocol is generally used on the internet not only for transmitting data between a client and a server, but for many other purposes. HTTP protocol is stateless, i.e. it does not recognise the clients from whom requests are made. If one client sends a request and then sends another one, the server does not recognise it was the same client.

HTTP exists in 3 versions - 0.9, 1.0 and 1.1 . The first of them, referred to as HTTP/0.9 existed as a simple protocol that could transmit data on the internet in a limited way. Version HTTP/1.0 enabled to transmit data in the MIME format, so it could also contain meta-information about the data transmitted. The most important improvement to the version HTTP/1.1, which is also the latest version, was that all connections became permanent, which means that the connection is closed when either client or server sends a header for closing. HTTP used to close the connection after each server's response. This improvement considerably accelerated the transmission, because the server does not have to open a new connection for each image, frame and applet.

6.3. HTTP protocol request format

The request of the HTTP protocol has the following format:

```
METHOD URL OF HTTP DOCUMENT
HEADER
blank line
OTHER-DATA    only in the case of the method POST
```

The request method indicates how the server should process the request. The methods will be dealt with later. Headers are sent in the following format:

NAME OF THE HEADER: HEADER VALUE

Each header has to be in a separate line. All lines have to be finished by tags CRLF (`\r\n`). At the end of all headers must be a blank line, even if there are no other data.

Requests in PHP are sent through so-called *sockets*. A socket is basically a connection between a client and a server. To work with them, it is necessary to open the socket first. For opening, the function `fsockopen` is used:

```
fsockopen(server, port);
```

Example:

```
$sock = fsockopen("www.interval.cz", 80);
```

When the socket is open, it is possible to send the request by means of the function `fputs`:

```
fputs(socket, request);
```

Example:

```
fputs($sock, "GET /index.html HTTP/1.1\r\nHost:
www.interval.cz\r\n\r\n")
```

In the following part, we will deal with HTTP methods.

7.HTTP PROTOCOL REQUEST METHODS

In HTTP/1.1, there are seven basic methods of HTTP requests:

- GET
- POST
- HEAD
- OPTIONS
- PUT
- DELETE
- TRACE

After each request, individual headers can follow. In the version HTTP 1.1, in each request it is compulsory to use the header Host, which specifies the host. After headers, a blank line must follow, as it has already been mentioned.

7.1. GET method

The GET method is the simplest and one of the basic method. Every time a page from a server is loaded without sending a form with the POST method first, this method is used for retrieving the page from the server. The result is thus a page and its headers which we ask about using this method. The format of this method is as follows:

GET URL-OF THE PAGE OF THE PROTOCOL VERSION

HEADERS

blank line

Example:

GET /index.asp HTTP/1.1

Host: www.interval.cz

blank line

7.2. POST method

The POST method works as the GET method, but in the case of the POST method it is possible to send data to the script after the headers and blank line. This method is used for sending the data from a form with the POST method. The format of this request is as follows:

POST URL-OF THE PAGE OF THE PROTOCOL VERSION

HEADERS

blank line
DATA FROM THE FORM

Example:

POST /processdata.php HTTP/1.1
Host: www.formulare.cz
Content-Length: 29
Content-Type: application/x-www-form-urlencoded
blank line
array1=value1&array2=value2

In the case of this method, there are more headers that have not been mentioned yet. Content-Length indicates the length of the data from the form (in bytes), and Content-Type: application/x-www-form-urlencoded designates the MIME type of the data from the form.

HEAD, OPTIONS, PUT, DELETE, TRACE

If you do not programme an Internet explorer, you are not likely to encounter these methods. Therefore, the following table shows only the basic meaning of these methods.

| Method | Description |
|---------|--|
| HEAD | It works as GET, but it does not return the body of the page, only the header. This is used e.g. for finding out whether the page has changed from the last request. |
| OPTIONS | It is used for requests about server possibilities |
| PUT | It works as GET, but it retains the body of the request at a location given by the required URL. It is similar to sending files via FTP. |
| DELETE | It removes the document from the server. The document that is to be removed is given the URL of the request. |
| TRACE | It is used for tracing the request through all proxy servers and firewalls the request passes through. It is similar to the tool TraceRoute. |

7.3. Using data sources

Modern and efficient data processing applications do not store information directly in their own files, but they use some of the external data sources. There are a number of data sources, starting from a simple text file through file databases e.g. in the DBF format to databases stored in SQL servers.

Each such source usually uses its own format for storing data and methods for retrieving the data stored. To avoid the problem with the diversity of data sources and to avoid the necessity to develop applications for each of such sources, applications programmers created a standard tool that enables to access various data sources by means of one standardized platform – ODBC.

Such a standardized platform enables the users to use the data from various sources at various locations in a complex and simple way. For example, it is possible to use the data from Access in Word or Excel without having to export them and load them into the required application first

7.4. What is ODBC

ODBC stands for Open Database Connectivity. ODBC data sources are accessible to applications through a relevant driver which can be perceived as a mediator for the communication between a user's application and external data source. The application sends the request about the data to ODBC. The relevant driver translates the request so that the external data source understands it and sends it to the data source. The response from the data source is also sent through ODBC, which translates the result into the standard form and returns it to the application.

The principle of ODBC is standard; therefore any application capable of using the ODBC driver can access any external data source from any producer that provides the appropriate driver. ODBC thus enabled to standardize the access to the data for the applications without solving how the data source works.

On the one hand, the application only needs to know how to work with ODBC. On the other hand, the producers of various data sources provide the relevant ODBC drivers so that their sources are easy to access for the applications. This is advantageous for the applications programmers that have independent standardized access to data, as well as for the users who get the applications that can access various data in a cheaper way and faster.

Working with ODBC in Windows

ODBC setup controls are already an integral part of the operating system Microsoft Windows. **ODBC data source administrator** is in the control panel. Here it should be noted that the procedures and images are valid in the operating system Windows 10, but similarly they are valid in other versions of Windows. However, in some operating systems, the ODBC administrator may be accessed in a different way.

8. DATA SOURCES

To fill the mailing lists and templates with the right data at the right time, Mailkit uses **XML & RSS data sources**. Their use is limited depending on the type of user account. The version Mailkit Base is limited to using XML data sources intended only for the import of the recipients into lists, while Mailkit Syndicate and Agency support both XML and RSS data sources even for loading data in templates.

Using XML data sources for recipients list

To create a new list of recipients from a data source it is necessary to set up a new XML data source.

- Name - name of data source. If the data source will work for the recipients lists, the list will have the same name.
- Description - description of data source.
- Source - URL address of the XML or RSS which will work as a data source.
- Authorization - it is ticked if the access to the data source location is password protected.
- Type - you can choose from RSS or XML data source.
- Destination - the data source destination is selected from the drop-down list. It will be used either for the list of recipients or template.
- Update automatically - if this option is ticked, the source will automatically update before the campaign is sent.
- Expiration time - it sets up the time after which the data source shall be update in the case of automatic updating.
- Last update - it shows the date of the latest data source update
- Empty records:
 - Reset - the recipients data will be deleted according to the empty records in the import.
 - Retain current value - the recipient data will remain in the same form as before the import.

How to prepare data source

The data source has no defined structure and can be in the formats XML or JSON, which must be fully valid. The data source file must be posted to a URL available from the Mailkit servers and secured against third-party access, as it is sensitive information.

As each client uses different information system with different possibilities, the data source system is universal and does not set specifically the structure of the required data. However, there are some technical limitations of the data sources:

- Fully valid XML or JSON
- Attributes are not supported in XML format (e.g. first_name="Jana" gender="f" country="cz")
- UTF8 character encoding recommended
- Do not use ";" for separating more values - use "|"
- The unique record identifier and the only compulsory array is the e-mail address. If there are more records with the same e-mail address, they will be overwritten.

As mentioned above, the only data compulsory is **e-mail**; other data are not compulsory, but important. The general rule is "the more, the better", but also nothing must be exaggerated. The data source shall be important the maximum of the recipient data available which are possible to be used for existing and future e-mail campaigns.

Import of data from data source

To obtain content, the source branches must be assigned to the contact fields. To assign, click on the name of the data source and click on the **View Structure** button. After assigning all fields, click on **Save**. Subsequently, it is possible to continue with **Import** for the current data import. At this moment, a new **Recipients list** will be created (it will have the same name as the source that created it), and the data from the data source will be imported according to the previous assignment.

At the same time, Syndicate and Agency account users have a possibility of automatic update. If automatic update is chosen, the data source will be automatically imported and the list of recipients will be updated before each campaign delivery begins. This parameter is not recommended for data sources containing more than a thousand of records, since the update can take a few minutes, which delays the delivery of the campaign. For larger data sources, it is recommended to use the possibility of regular planned update scheduled for a specified time of day by the customer support.

Using XML & RSS data sources in templates

Setting up XML & RSS data sources for the use in templates is similar to the use for recipients lists, but without having to assign the meanings to each field. The values are determined by a string of names in the template, therefore it is easy to set up any XML or RSS source.

Above, there is an example of a template code for which the RSS data source called *EXAMPLE* is used. The command *FOREACH* creates a loop for parsing and finding all records. Each of standard RSS tags is easy to solve and insert into HTML code, allowing the data output in the template. For more information, see [Emails templates](#).

Product data sources

Data sources can also be used for transmitting the product offer to Mailkit and for subsequent use of product information in campaigns. This is where the power of data sources and programmable templates are shown, which enables to combine data from more sources and personalize the content for the individual recipients automatically.

For product information, it is possible to use any of the common product feed formats for Heureka, Zbozi, Google, etc., or generate the own feed with the necessary information. Since the product feeds are very extensive and the speed of working with the data contained is important, these data sources are transmitted directly into the SQL databases, and it is still possible to work with them. For setting the product data source, you shall contact the customer support that will help you with the implementation and use.

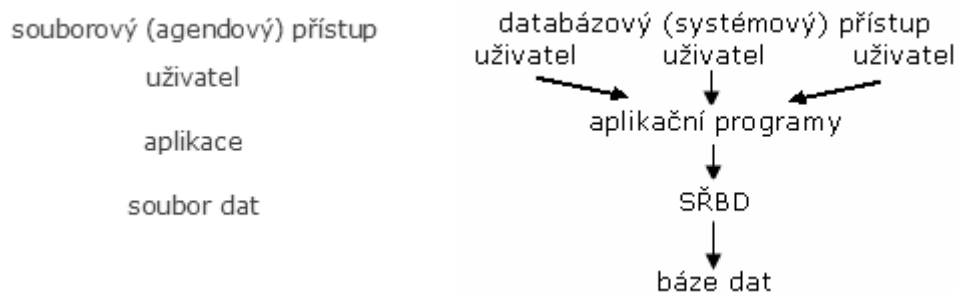
Delivery feeds

Delivery feeds are special data sources that are used for passing structured information for the realization of the campaign delivery. While a campaign usually uses the pre-set list of recipients, which is used for delivering the campaign according to the rules set, in the case of delivery feed, the campaign is sent only to the addresses listed in the feed. It is an alternative to API call `mailkit.sendmail_mass`, that is, the way to get highly structured data to be processed when sending the campaign to Mailkit e.g. from personalization systems or CRM. These feeds must have a strictly defined structure in the XML format.

9. DATABASE APPROACH

Requirements to database system:

- data consistency check - database shall be able to ensure compliance with certain rules of so-called integrity constraints and secure ensure data against possible accidents that may occur during transactions.
- transaction - sequence of manipulation with data, which must take place in order to ensure that the data are stored properly, e.g. transfer from 1 account to another account (this must be done correctly in both accounts)
- large volumes of data - relatively to the possibilities of storage media, the database must be able to hold adequate amount of data
- data management - development stages



Legend: souborový (agendový) přístup - file (agenda) access, uživatel - user, aplikace - application, soubor dat - data file, databázový (systémový) přístup - database (system) access, aplikační programy - application programmes, báze dat - data base

Database access

- large database system - companies Informix, Sybase; smaller (more affordable) database systems - MS Access, Paradox, FoxPro; and small database systems available for free - e.g. My SQL
- SQL language- standard that enables to use data sources managed by various database systems

Creation of a database for the IS organisations - a complex issue requiring people with different professional skills. When designing the conceptual scheme of the database, it is decided what the database will contain. During the IS operation, it is important for the users, if they are able to use the data base as an information source.

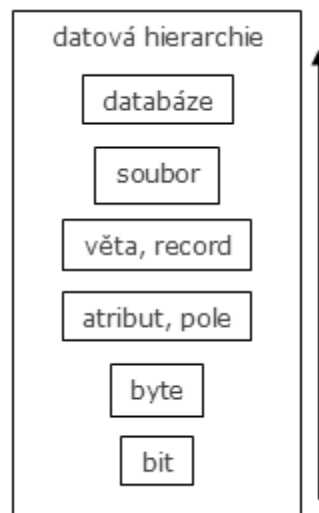
Requirements to computer IS:

- horizontal and vertical integration of information
- fast aggregation of information from the lower stages

- rational presentation of the information in time, form, and space
- time frequency
- Extent of the information stored

Data organization - files and databases

- transactions processing: batch processing or online processing
- current trend - object-oriented and hypermedia databases



Legend: datová hierarchie - data hierarchy, databáze - database, soubor - file, věta, record - statement, record, atribut, pole - attribute, array

Design of structured database

- the reality whose reflection shall be the designed database. It consists of several objects (entities)
- there can be different relations between the monitored entities (e.g. between the entities of the same type = recursive relationship)

Cardinality of the relationship: symbolic designation 1:1 (professor XY's wife is ZN); 1:N (professor XY teaches students); M:N (which students are taught by which professors)

Integrity constraints of the database - all rules defining the allowed values (a combination of values) of the attributes, display format

Relational data model

- It assumes the existence of single-value attributes
- representation in the form of a relational table, in which a tuple corresponds to a row, and an attribute corresponds to a column

Relational database

- all data is in the form of 1 or more tables named by columns
- each column contains data from 1 domain (i.e. 1 data type)
- the elements of the individual columns (with name and type) are usually called items or arrays, and the term row corresponds to record (statement)
- in terms of a relational model, sessions describe both entities and the relationships between them
- based on this dual use, it is possible to distinguish between so-called entity sessions (sets of arranged tuple attributes describing the entities) and relationship sessions (sets of arranged tuples).

Data warehouses

Warehouse principle - 2 main objectives

- unification of data view in the individual so-called production systems provides a clear access to data - differently called but the same objects are seen as one object
- differently measured but the same variables are measured by the same measuring tool

10. STRUCTURED AND UNSTRUCTURED DATA

10.1. Structured data

Basic types (classification to distinguish allowed and non-allowed manipulation and values)

- text (strings, expressing information by means a text code, certain set of elements that can be recorded, or that can define syntax)
- numerical - real rational numbers
- date, time - limited values it can achieve (February 30, 27:00)
- logical - meeting the conditions of existence or non-existence of an object characteristics - 2 values (0 and 1, A and N)
- category - value of characteristics selected from a scale (often dials, enables to record values only by means of a code)
- Structuring creates a data organisation that enables efficient storage, processing and retrieval of data as needed → structured data create search keys (sometimes referred to as identification keys) - keys that clearly identify data records are called private keys (identification keys) - a basic condition of data and database system
- Information about something (reflection of reality) - Name and surname, address, age, phone number, weight, price,... number of points, category, average grade,... number of pieces, number of pages
- Operations or what can be done - addition, rounding, multiplication, connection (name+surname), shortening, sorting,..., day in a week, negation, ...
- Data type (must be defined) - number, text, date and time, logical data (yes/no)
- Encoded data - various encodings - text, letters - various code tables (ascii, ebdc, ...) national alphabet; date and time (how we write a date)

10.2. Unstructured data

Data type: free text, audio, video, graphics, multimedia

- Provide more information that structured data
- Problem: it is very difficult to search using unstructured data. Solution: unstructured data are supplemented with structured data (mp3 title)

10.3. Data volumes - structured and unstructured data

- ascii text page (notepad) 1.8 kB
- word text page 50 kB
- vector graphic A4 30 kB
- bitmap image A4(jpg, rgb) 5 MB
- 1-minute audio record (wav) 10 MB
- 90-min video record 3 GB

Data Warehouse (DWH) is a special type of relational [database](#), which enables to solve tasks focused primarily on analytical querying over large data sets.

Subject orientation

In the case of a common relational database, it is common to try to minimize the redundancy of data stored, which is achieved by their normalization into [third normal form](#) and internal interconnection of the individual logical functional units. In the data warehouse, there is always an effort for a clear internal separation of the individual functional units. The result is a structure clearer for a user (manager, business analyst) at the expense of increased requirements for a storage space.

Integration

A common operational application (a programme) over a relational database solves a certain specific kind of tasks over "its" specific data. In a data warehouse, it is necessary to gather the information from many various sources and group them not by their origin but by their logical meaning (it is closely related with the subject orientation - all data concerning a certain functional area must be "on one stack" regardless where they are from).

Low variability

The data are usually uploaded into a warehouse in larger batches (e.g. at weekly or daily intervals) and are not modified afterwards.

Historicization

The data in a data warehouse are usually stored in their historical form, not only in its current state. This is due to the necessity to carry out analyses focused on the development over time. From the users' point of view, in a common relational database, only the current state of data objects is interesting.

10.4. Technological characteristics of data warehouses

The requirements for data warehouses imply their technological characteristics:

- A data warehouse must contain a tool for uploading data from various data sources. These sources can have different data formats and different physical location. They do not have to be only relational databases.
- A data warehouse stores data not with regard to the best editing conditions, but with regard to the best and fastest execution of complex requests. Therefore, for data storing, OLAP technology is often used.
- It cannot be said which requests and tasks will the users have in the future (at the time of creating a data warehouse, we only know the type of the tasks, not the individual tasks and requests).

10.5. Logical structure of data warehouse

From the logical (user's) point of view, the data in a data warehouse are divided into diagrams. Each diagram corresponds to one analysed functional area.

The core of each diagram consists of one or several fact tables. They store the analysed data - numerical and financial values which are used for analytic calculations - aggregation, sorting, etc. Most storage space in a data warehouse is occupied by fact tables containing detailed information from all sources, i.e. more than other tables.

Fact tables are linked to dimensions by means of [foreign keys](#). Dimensions are tables that contain lists of values used to categorize and sort data in fact tables.

Example

It is necessary to store information about all sales from cash registers of hypermarkets in a data warehouse. The data will be further analysed on the basis of the time of the sale, shop, type of goods, supplier, ongoing marketing events, and the payment method (with a card, in cash).

The Sales diagram will contain a fact table - Sales items, where information about the type of the goods sold, price, and number of pieces (or the amount sold) will be stored for each item sold.

Besides this fact table, the diagram will also contain dimensions for sorting the sales items: time dimensions Date and Hour (within a day), dimensions Shop, Type of goods, where there is one row for each item (e.g. Strawberry yoghurt 250 ml), Category of goods (Yogurt), Department (Dairy products), Supplier (Dairy XX), etc.

The fact table must be directly or indirectly linked to each of these dimensions using a [foreign key](#).

There are two options in terms of storing hierarchical dimensions:

- From the entire hierarchy, one dimension table will be created that will store the data for higher levels of the hierarchy redundantly. This way a diagram is created where each dimension table is linked directly to a fact table. According to the shape, this diagram is called a star schema.
- 3NF will be applied to hierarchical dimension, so only the dimension at the lowest level of the hierarchy will be linked directly to the fact table. Other dimensions will be linked to any of the lower dimensions in the hierarchical structure. This is called a Snowflake schema.

II. AJAX

AJAX stands for Asynchronous JavaScript and XML. It is a generic term for technologies of the development of [interactive web](#) applications that change the content of their pages without having to completely upload them. It is performed using [asynchronous](#) processing of [web pages](#) by means of a library in [JavaScript](#). Unlike classic web applications, they are more user-friendly, but require using modern [web browsers](#).

These applications are developed using the following technologies:

- HTML (or XHTML) and CSS for presenting information;
- DOM and JavaScript for displaying and dynamic changes of presented information;
- XMLHttpRequest for asynchronous data exchange with a web server (typically, XML format is used, but it is possible to use other formats, including HTML, text, JSON or EBML).
- Like DHTML, LAMP or SPA, Ajax is not a concrete independent technology. It is a term referring to several technologies together with a specific goal.

Advantages

Advantages include elimination of the necessity to reload and redraw the entire page in each operation (unlike the classic model of www pages). If a user clicks a vote button in an opinion poll, the whole page must be reloaded from the server, even if only the vote results are updated and the rest of the content remains the same. Using AJAX, the vote is sent in the background, the server sends only the parts of the pages that have changed, and those parts are updated and redrawn on the page. Moreover, there is no unpleasant effect when after the activity, its block elements, images, etc. are gradually adjusted and formatted in the continuously uploaded page. It can also be annoying that after the specific activity, in the middle of a longer page (scrolled down), the newly uploaded page is displayed scrolled up. Using AJAX, the user works more fluently and the speed (especially in the case of faster internet connection) is nearly the same as the speed of common desktop applications.

This also has the potential to reduce the load on the web servers and web in general. Since it is not necessary to create the whole HTML document for each request, but only the changes made have to be marked, the volume of the data exchanged is significantly smaller and theoretically, it can also positively influence the load of the database servers or other backend systems.

Disadvantages

On the other hand, AJAX can increase the *number* of exchanged HTTP requests, and even though they transmit a smaller volume of data, in the case of inappropriate implementation the load does not decrease.

The disadvantages include mainly the changes in the web usage paradigm: web pages behave as applications with a complex internal logic, not as a sequence of pages which can also be navigated using Back and Next buttons. Similarly, it is not possible to pass the [URL](#) of the page in which something has been "clicked" using the AJAX technology. Modern AJAX applications are able to recover (at least partly) history browsing using various techniques (e.g. using a part of the address after the # or using invisible [IFRAMES](#)). This, however, makes the pages design complicated and more time-consuming, as well as more work on implementing using AJAX.

Another problem of AJAX applications can be the network [latency](#): the need for Internet communication has a negative impact on the response speed and the user interface interactivity. If the user is not announced clearly that the application is processing their request (and communicates with the server in the background), all they register is a delayed response (the users may even try to start the operation again as they think that the system has ignored the request, thus generating higher load to the server and / or cause something they did not plan, e.g. to order ten tickets instead of two). As a suitable solution, it is recommended to show somehow that the user's request is being processed, e.g. using text or animated icon.

Another AJAX disadvantage is the necessity to use modern graphics browser that support the necessary technologies. However, all currently used common browsers support these technologies at least basically; the problem is only with the minority Lynx browsers or with hardware-weaker browsers, e.g. in some [mobile phones](#) and [PDA](#). Within web accessibility, it is required that the pages are accessible from browsers even without AJAX, which means more time and work for the developers and higher costs for the site buyers.

AJAX

As it has been mentioned above, AJAX stands for Asynchronous JavaScript and XML. AJAX is a modern technology frequently used in the current web applications. It is often mentioned, as it is a part of RIA, a new programming style leading to higher user's comfort and functionality of applications.

AJAX is not a new technology, it is only a combination of already known technologies - HTML (or XHTML), JavaScript, XML and XMLHttpRequest.

Why is AJAX so advantageous? The applications using AJAX can send and retrieve data from a server without having to reload the whole page (unlike classic links). AJAX can thus be used for various purposes, e.g. autocompletes (forms that fill in automatically depending on the key pressed), AJAX opinion polls, and other more complex applications that are able to facilitate the work of a user.

AJAX has also several disadvantages, especially when used improperly, it significantly reduces the usability of the pages. As with other technologies, it is therefore necessary to plan AJAX application carefully and test it on users.

As you already know, Javascript is so-called client-side language, which means that it is done on the client's web browser side and is interpreted by javascript interpreter. It has a wide range of characteristics which include also the capability of performing asynchronous operations / tasks.

What exactly does AJAX asynchrony consist in? It is the ability of Javascript to call a script or element API on the server and not wait for the response. Instead, code execution continues (for example, the user can see the page and can work with it). When the response comes, code context execution is stopped (sooner or later, depending on the priority of the task) and there is a callback (it is a function that is performed if the server provides a response). Such a code or function is called "non-blocking", because it does not block the course of script by waiting for the response or processing.

AJAX is a method of programming in Javascript. It is not a new language or framework or a third party library. It is a method of exchanging the data in the application with a database, server scripts (PHP, Java, ASP, etc.) without having to update / reload the whole page. Technically, we do not update (in terms of refreshing) any part of the application or a page - this is actually the AJAX essence.

Why XML is contained? It is one of the data format in which AJAX can retrieve the information from the server. The most widely known and used are JSON, XML, text, binary data. Each of them can be used in a different way.

AJAX enables to check the data entered by the user even before the user sends them by confirming the form. To some extent, this can also be done by Javascript, but on in the case of a registration form with a conditional password, user name or e-mail, which have to be unique within a table or database, and these are exactly the situations for which AJAX is the right technology. It enables to perform the same operations that could be performed by calling PHP scripts when redirecting after sending the form, but unlike them, it allows you to make this call in the background.

AJAX basics

The main idea behind AJAX is in the fact that parts of a web page are uploaded asynchronously, thus changing the content of the page. It means that changing the content will not cause uploading of the whole page, only the modified content. For this purpose, the object XMLHttpRequest is called from JavaScript. This request asks the server about the content that is typically rolled back using XML and JavaScript. The script then analyses this XML. The data from XML script then translates and changes the web page using DOM (Document Object Model) correspondingly.

The basic principles of working with AJAX will be described on a practical example - a discussion forum, in which the discussions in HTML are displayed as a tree. If the users want to read a contribution, they have to click on its headline. In the case of a classic model, the request would be sent to the server and subsequently a whole page would be created, that would contain the text of the contribution and the whole "tree". In the case of AJAX model, only the text of the contribution will be sent as a reaction to the request. Adding the text to the original web page will be ensured by a relevant JavaScript programme.

An ActiveX object was available for older Internet Explorer versions (starting with the version 5.0) for performing this task. Like other explorers, IEZ makes the corresponding object available natively. The current API XMLHttpRequest of the Internet Explorer is available on MSDN; Mozilla project API is also available.

AJAX's own trick is XMLHttpRequest - it can be processed asynchronously, which in this context means asynchronously to the arrangement of the rest of the page as well as to the running scripts. This means that the data can be downloaded by means of HTTP and that the data can be downloaded without affecting the interaction of the user and the page. The requested page is also created in the background.

What is AJAX? Simply said, AJAX is a technology that uses scripts to facilitate the communication of a page and a web server. This way it is possible to change the page content without having to upload it again.

Theoretically, it is possible to create dynamic pages without server scripts, that is, to have one page for the whole web, which only changes its content. However, this is not recommendable - as it has already been said, it is a script technology, so it depends on the client and it cannot be secured it will work. Nevertheless, it can be used as an add-on for web pages - e.g. opinion poll or a search assistant as in the case of Seznam.cz.

If you are sure that AJAX will work for the visitors of your web pages, it can be used also for a more sophisticated applications. It can be used e.g. for a chat that downloads only new contribution; if the visitor is alone there, checking for updates slows down, which saves a lot of data transmitted.

12. FRAMEWORKS

Framework (application framework) is a [software](#) structure, which serves as a support for [programming](#) and development and organization of other software projects. It can contain supporting [programmes](#), [API](#) libraries, support for [design patterns](#) or recommended procedures in the development.

- Purpose
- Architecture
- Examples
- Related articles
- External links

Purpose

The objective of a framework is to take over typical problems of a given area, thereby facilitating the development so that the designers and developers can focus on their tasks. For example, the team that uses [Apache Struts](#) for developing [web pages](#) for a bank can focus on bank operations and not also on ensuring a perfect navigation between the individual pages.

It is argued that using frameworks will make the code slow or inefficient and that the time saved by using a foreign code has to be given to studying the framework. However, its re-deployment or its use for a large project saves a significant amount of time. After uninstalling the framework, some applications will not be able to work any more.

Architecture

Frameworks consist of so-called frozen spots and hot spots. Frozen spots define the overall architecture of the software structure, its basic components and the relations between them. These parts do not change in any framework use. On the contrary, hot spots are components that create a completely specific functionality together with the programmer's code; therefore they are each time different.

In an [object-oriented environment](#), framework consists of abstract and classic (non-abstract) classes. Frozen spots can be represented by abstract classes and the code (hot spots) itself is added by implementing abstract methods.

Framework is a software structure that serves as a support in programming and development and organization of other software projects. It can include supporting programmes, API library, design patterns or recommended procedures in the development.

Another very important factor in choosing is what the framework is needed for and its purpose. Generally, frameworks can be divided into two groups:

- Sets of scripts – libraries – covering all kinds of needs
- Scripts creating one concrete web application

Advantages

- Faster development
- Less code
- Universal code – changes or new functions are added in a much easier way
- Nice URL

The development speed is higher. Framework makes working easier, avoiding programming routines such as connecting to a database, checking the right option with a sect, or validating a form. Cool-Url, XSS, separate application and presentation logic, changing DB server from MySQL to PostgreSQL will be really easy.

Thanks to framework, it is possible to focus only on the development, but it is redeemed by the time you need for learning to work with frameworks. When you learn to do something, soon you will find out it can be done much easier and faster. Learning to work with framework can be a matter of a month, but also a matter of several months or a year. Finally, you will extend the framework by libraries and you will be able to significantly shorten the programming time. Framework means saving time and money

Choosing a framework is not easy. It is recommended to have at least two - one for easy and second one for more complex tasks. Before each project, it has to be decided which one is more suitable. Of course, the best thing to do is to learn those with a good large community, those that you are sure their development will not be finished in a week.

- CakePHP: quite easy to learn; quality community; long development of a new version
- Zend Framework: very efficient; it covers all needs in creating any web applications; large community; sophisticated; long titles;
- CodeIgniter: small, simple; developed by a company, not by a community; better than Cake in some issues x "stupid" in others (it does not have layouts)

Monolithic frameworks gradually decouple into separate components, which brings a lot of advantages. While using only one framework component used to be difficult and sometimes even impossible, nowadays it is possible to install its component easily. The development cycle of individual components is different. They can have their own repositories, issue trackers, development teams.

Components can be updated no new versions continuously, without waiting for another version of the whole framework. It is also possible not to update some component, e.g. because of a BC break.

The meaning of the term “framework” is thus shifting, and “versions” now almost do not have their original meaning. Instead of a framework XYZ, version 2.3.1., a set of component in different versions working together is used.

12.1. What is a framework?

When creating modern web applications, frameworks, or development frames are often used. It is a set of libraries and source code that could be repeatedly used in order to facilitate work and whose functionality can cover a part of the application created. Frameworks can solve problems that can be generalized in some way; otherwise creating a framework would not make a sense. This allows the application creator to focus on the functions that are unique and exclusive for the application, and does not have to solve routine problems.

Using Javascript frameworks

Frameworks can be created for most programming languages. Javascript framework is thus a framework that is used to facilitate work and programming in Javascript. Frameworks use is very wide and every day, the portfolio of problems Javascript frameworks are able to solve is bigger. Generally, it can be said that they can be used for effective writing of source code; to a large extent they also solve incompatibility between web browsers and a written code, thus saving the programmer’s time and enabling to use elements that would be difficult to programme. Their strength and use is mainly connected with using the AJAX technology, as they improve the interaction of the user and application by means of asynchronous communication between the client and the server. Furthermore, they enable dynamic and simple access and change of the individual elements of a page (DOM model), GUI elements, assign events and animate them. Last but not least, some frameworks contain pre-built GUI components that either cannot be implemented using any other technology or it is too difficult and inefficient. These elements are easy to work with and implement within a framework using just several code lines.

13. 10 BEST PHP FRAMEWORKS FOR DEVELOPERS

PHP, known all over the world as the most popular server-side scripting language, has developed a lot since in static HTML files, first inline code fragments started to appear.

At that time, developers had to create complex webs and web applications; at a certain complexity, it was too **time-consuming and it took too much effort** always start from scratch. Hence the need for a more structured and natural way of development appeared. PHP frameworks provide an adequate solution to this problem for developers.

Why to use PHP framework

Firstly, we will deal with the strongest reasons why many developers prefer using PHP frameworks, and how these frameworks improve the development process. PHP frameworks advantages are as follows:

- Enabling rapid development
- Providing well-organized, reusable and maintainable code
- Enabling growth over time, since framework-based web applications are scalable
- No worries about low-level web security
- Following MVC (Model-View-Controller) pattern, which secures the separation of presentation and logic
- Encouraging modern web development practices including object-oriented programming tools

1. Laravel

Although Laravel is a relatively new PHP framework (it was released in 2011), according to the latest online survey on Sitepoint, it is the most popular framework among developers. Laravel has a huge ecosystem with a platform ready to be hosted and deployed immediately, and its official web offers many tutorials in the form of screencasts called Laracasts.

2. Symfony

The components of the framework Symfony 2 are used by many projects, such as the system for managing the content of Drupal or the software phpBB for running forums. It is also used by Laravel. Symfony has an extensive developers' community and many supporters.

3. CodeIgniter

CodeIgniter is almost 10-year-old lightweight PHP framework (originally released in 2006). CodeIgniter has a very straightforward installation process that requires only minimal configuration. It is an ideal option to **avoid conflicts with PHP versions**, since **it works**

smoothly on almost all shared and dedicated hosting platforms (currently, it requires only PHP 5.2.4).

4. Yii 2

Choosing Yii frameworks gives a page a boost for performance, since it is **faster than other PHP frameworks**. It extensively uses the “**lazy loading**” technology. Yii 2 is purely **object-oriented** and it is based on the **DRY** (Don't Repeat Yourself) coding concept, so it provides a **clear and logical code base**.

5. Phalcon

[Phalcon](#) framework was released in 2012 and soon became popular among the PHP developers. It was said to be as fast as a falcon, as it **was written in C and C++ languages in order to achieve the highest possible performance optimization**. For using Phalcon, it is not necessary to learn language C, because the **functionality is exposed as PHP classes ready to be used in any application**

6. CakePHP

CakePHP has been used for a decade (first version was released in 2005), but it is still one of the most popular PHP frameworks, since it has always strived for being up-to-date. The latest version, CakePHP 3.0, engaged session management, improved modularity by separating several components, and increased the ability to create **more self-sufficient libraries**.

7. Zend Framework

[Zend](#) is a robust and stable PHP framework packed with a number of configuration options; therefore it is usually **not recommended for smaller projects**. It is, however, great for **the more complex ones**. Zend partners include e.g. IBM, Microsoft, Google, and Adobe. The coming new version, Zend Framework 3, will [optimized for PHP 7](#), but it will still support PHP 5.5.

8. Slim

[Slim](#) is a PHP micro framework that provides only what is needed. Micro frameworks' design is minimalist, they are great **for smaller applications**, for which a fully equipped framework would be too much. The Slim creator was inspired by a micro framework Ruby called [Sinatra](#).

9. FuelPHP

[FuelPHP](#) is a flexible full-featured PHP framework that supports not only MVC, but also its developed version, [HMVC](#) (Hierarchical Model-View-Controller). FuelPHP adds an **optional class** called [Presenter](#) (previously called ViewModel) between the View and Controller layers **to include logic necessary for generating views**.

10. PHPixie

[PHPixie](#) is a brand new framework started in 2012 in order to create a high-performance framework for read-only webs. Like FuelPHP, PHPixie **implements design pattern HMC** and is created by means of independent [components](#), that **could be used also without the framework**. PHPixies components are 100% tested through [unit tests](#) and require only a minimum of dependencies.