**Introduction to Web Development**

# Linked in => na portfolio

Pokud si budes verit, mel bych se zeptat po pomoci (zkušenost a praxe jobs.cz) =>

## Front-end develompent

* the part where you create the actuall website
* HTML, CSS, JavaScript, JQuery, JSon
* HTML and CSS are the building blocks (even for back-end), that can create a functional basic website by itself. For the more complex function you will need JavaScript and JQuery. Later, if you want to connect with back-end development you want to learn JSON
* Others: Angular (javascript framework = make one paged websites),

### HTML (Hypertext Markup Language)

* puts the content for the webpage

### CSS (Cascating style sheets)

* the style and composition for the webpage

### JavaScript

* makes the webpage do cool things (animation, moving things) => like instagram and pinterest
* the fun part that pays more than the others

### Jquery

* javascript library
* simplifies a lot of things in JS

#### JSON

* javascript object notation
* connects with back-end development

## Back-end development

* all the thing that goes in the background
* makes the website do stuff
* login system, search field, email messaging through the web, etc…
* HTML, CSS, PHP, SQL, AJAX, OOP PHP

### PHP (Hypertext Preprocessor)

* one of the programming languages you can use
* helps the user interact with the website
* handles the information

### SQL (Structured Query Language)

* whe we want to interact with our databases
* handles pulling out or inserting the information from the database

### OOP PHP (Object Oriented Programming PHP)

* is another way of programming but in a more structured way
* for better readibility (for other programmers in a company)
* if you want to get hired by a company

## SEO (Search Engine Optimization)

* there are also people who make a lot of money of SEO (maintaining a good SEO for big companies)

### On-page

* we optimize the code so Google likes our website
* search words, titles, description, content quality, semantic markup, etc…

### Off-page

* we build up some value by having people visit our page
* link quality, text, reputation, country, shares, etc…

# Static v Dynamic Website Design

There are basically two main types of website - static and dynamic.  
A static site is one that is usually written in plain HTML and what is in the code of the page is what is displayed to the user.

A dynamic site is one that is written using a server-side scripting language such as PHP, ASP, JSP, or Coldfusion. In such a site the content is called in by the scripting language from other files or from a database depending on actions taken by the user.

### Static sites - advantages

Flexibility is the main advantage of a static site - every page can be different if desired, to match the layout to different content, and the designer is free to put in any special effects that a client may ask for in a unique way on different pages. This allows theming - for instance an author may want a different theme for a different book and associated pages or perhaps for a series of books, in order to match the cover designs or the context of the stories.

Cost is generally lower up-front than a dynamic site.

### Static sites - disadvantages

The main problem with any static site appears when you wish to update the content. Unless you are conversant with HTML and the design methods used in the site then you have to go back to the designer to have any content changes made. This may be perfectly ok when a new page is required which needs design input, but if all you want to do is change some text then it can be a nuisance for both client and designer.

The second main problem is scalability. If you wish to sell products on your site and you have a lot of them then you may have to construct individual pages for each one, which can take considerable time, effort and cost.

Costs - there are ongoing costs for updating the content.

### Dynamic sites - advantages

The main advantages of dynamic sites are that by connecting them to databases you can easily pull in information in an organised and structured way to create product pages or categories of related products sorted in a variety of different ways depending on how the user wants to view them.

This ability to connect to a database means that you can also create a [content management system](http://www.spiderwriting.co.uk/content-managed.php) - an interface which allows the client to input and manage data via a web-based series of administration pages. That content can be text for their pages and images to go along with the text, or items in their product range with categories, specifications, short and long descriptions, images, etc. In both these cases it can be as simple or as complex as the client requires.

There are little or no ongoing costs unless there is a change in the basic design or an extra capability added.

### Dynamic sites - disadvantages

The design of a dynamic site is more fixed than a static one because many of the pages are essentially a template into which data and content is poured to create multiple pages of a similar type. So for instance all your product pages will be essentially the same page layout with different data being displayed. While some customisation cabability can be built in it is usually quite limited, such a selecting from a set of pre-defined options. Individual layout changes to particular pages are not usually possible.

Costs are higher initially than for a static site, and additional functionality may also cost more, particularly if it‘s something that wasn‘t envisaged originally and requires re-writing of the core code or database.

## Frameworks

* makes certain programming languages a lot easier (creating and reading)
* Bootstrap and W3.CSS (for HTML and CSS),

## CMS (Content Managment System)

* tool that makes websites without coding
* if we have a client, for whom we programmed a website but in the near future he wants to change the content by himself, then we need to include the CMS system
* WordPress is probably the best known

### HTML

* adds content and structure to the webpage

#### Tags <>

* have a certain function

**Html**

* has the bigges priority as a tag (with the exception of the pseudo class **:root**)
* to create content and structure, everything is contained inside

<html>

</html>

* you can also specify the language **<html lang=“en“></html>**

## Head

* one of the main tags in your html document

Contains: **meta tags =** has an internal function which isnt shown in the website itself (with the exception of **title =** the webpage a title name)

**link tags =** connects the html document to others documents like **CSS stylesheet**, **favicon** = the icon next to your title

## Meta Tags

* tags which help the website in some way but arent visible on the webpage
* some meta tags help wit SEO (search engine optimization = helps search engines like google to see if your page is revelant to show)

### Some important meta tags

**<meta charset=“UTF-8“> (defines what kind of characters you use inside your website)**

**<meta name=“viewport“ content=“width=device-width, initial-scale=1.0“> (helps with responsivnes)**

**<meta name=“description“ content=“This is a website that teaches HTML and CSS“> (description of your website, use 120-140 characters)**

* use google search console for keywords to use in description and title

**<title>Julianswebpage</title>**

<meta name=“keywords“ content=“coding, tutorials, html, css“> (used to help with SEO but dont use it its not used anymore)

<meta name=“author“ content=“Julian Xu“> (helps with SEO but dont use it, its not used anymore)

## Favicon

* the small icon on top of url link next to your title
* should be a really small size img (not 500x500 px)

### In HMTL (<head>)

<link rel=“shortcut icon“ type=“image/png“ href=“img/favicon.png“>

## Body

* one of the main tags inside your html document
* contains most of the content

<body>

(Context inside the page)

</body>

## Line break

* next line (used for line breaking)

# <br>

* before the tag => pushes the whole thing down a line
* after the tag => makes a new line under the tag
* you should use <p> paragraph (breaks line from both lines)

## Paragraph

* used for a paragraph of text, titles and headers

<p>

Hello world

</p>

Attributes= Title=”the showup text” (description of the paragraph)

style=”color:red;” (can be styled without CSS)

## Division

* can be used as a container, wrapper, etc..
* it boxes things together
* used for easier styling in CSS
* don’t overuse it

<div></div>

Attributes: Class

### ID

## Id and class

* use **<div class=”category”> </div>** to categorize most of tags, also is used for css styling.
* use **<div id=”category”> </div>** only rarely, can be good for anchoring a inner link, making a footer that stays at the bottom
* you can add more of the same classes but no more of the same id

## Section

<section> use for dividing themes (article about rolercoasters)

</section>

# Rozdíl mezi section a div

Muzu pouzit section místo div (např když dělám layouts/rozlozeni) => div není doporuceny?

## Article

* used as for a text paragraph
* often used in section

<article><p>TEXT</p></article>

## Comment

* is a invisible comment which helps to organize

<!--TEXT-->

## Span

* used for styling (go down for more)

## Nav

* for menu navigation

<nav>

</nav>

## Anchor

* adds a anchor which connects to some source

<a>Google.com</a> **WON’T WORK** (needs a refrence)

<a hrf=”Google.com”>Google.com</a> For linking to an external page

**use <a hrf=”Google.com”>Google.com target=”\_blank”</a> to open in new window**

**<a href=”index.html”>Brown Bear</a> For linking to a internal page**

**<ul>**

**<li><a href=”#introduction”>Introduction</a></li>**

**</ul>**

**<div id=”introduction”> For linking to same page**

## Images

Needs a source

<img src=””>

For manipulating the photo use CSS <class>

If its in a folder

<img src=”folder/image.jpg”>

Attributes: alt=”Description of photo” (for search engine, **very** **important**)

### Pair with link

<a href=“<https://en.wikipedia.org/wiki/Opuntia>“ target=“\_blank“><img src=“#“ alt=“A red prickly pear fruit“/></a>

### Can manipulate the resolution (better used with CSS)

<img src=”” width=320 height=260 controls>

### Video

* the best format is .mp4

**<video src=””> needs a closing tag unlike <img>**

</video>

### If a browser doesn’t support the format of the video

<video controls>

<source src=”video.mp4” type=”video/mp4” > (you can add more sources)

Your browser doesn’t support mp4 video, change browser

</source>

Attributes: autoplay (plays automatically)

poster=”#.jpg” (the thumbnail)

controls (having the control over the video)

## Unordered and ordered list

* a list of items
* can be used to bullet point or number things, making a navigation anchor (like in Wikipedia), menu navigation, etc..

### Bulletpointed

<ul>

<li>item1<li>

</ul>

### Numbered

<ol>

<li>item1<li>

</ol>

## Writting styles and colors

### Bold and italic

<strong>Hello</strong> = **Hello** (Bold)

*<am>Hello</am> = Hello (italic)*

Don’t use <b>, <i> = we used to use it in the past but mobiles can‘t render it

### Header

* how big should the text be
* used for headers, main titles, etc..
* from h1 – h6

<h1>Hello</h1> Hello

(Only use one H1)

<h2>Hello</h2> Hello

<h3>Hello</h3> Hello

### Span

# For colors

<span style=“color:red“> **TEXT** </span>