Úvod do informačních technologií

webové technologie

Martin Trnečka

Katedra informatiky
Univerzita Palackého v Olomouci

Služba WWW

- World Wide Web
- nejznámější („nejpoužívanější“) služba poskytovaná v sítí Internet
- umožňuje prohlížení, ukládání a odkazování dokumentů
- Sir Timothy „Tim“ John Berners-Lee (*8. června 1955 Londýn)
  - konec 80. let informačního systému ENQUIRE pro CERN
  - vznik jazyka HTML, protokolu HTTP, URL a prvního prohlížeče
  - počátek informační revoluce
  - 1991 → služba WWW
  - „otec webu“

URL

- identifikace souborů pomocí URL (Uniform Resource Locator)
- zjednodušená podoba URL:

  `protokol://doménové-jméno:port/umístění-na-serveru`
- například: `http://www.inf.upol.cz/adresa/index.html`
- speciální význam `index.html` a další
- absolutní vs. relativní URL
- WWW = HTTP(S)
- HTTP neřeší bezpečnost

Odbočka: URL, URI, URN

- často zaměňováno
- URI = identifikátor zdroje, například ISBN
- URL = umístění (lokace) zdroje, například adresa
- URL a URI lze u služby WWW zaměnit
- URN popis jména zdroje (striktní formát)

Protokol HTTP

- verze HTTP/1.1, HTTP/2, HTTP/3 (HTTP over QUIC)
- bezstavový protokol = neuchovává informace → cookies
- metody GET, POST

```
GET /pro-zajemce-o-studium HTTP/1.1
Host: www.inf.upol.cz
Connection: keep-alive
User-Agent: Mozilla/5.0
Accept-Language: cs
```

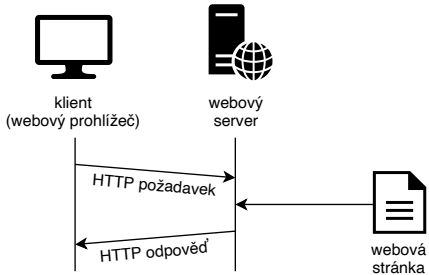
```
HTTP/1.1 200 OK
Date: Mon, 09 Sep 2019 09:14:40 GMT
Server: Apache/2.4.10 (Debian)
Content-Type: text/html; charset=UTF-8

data
```

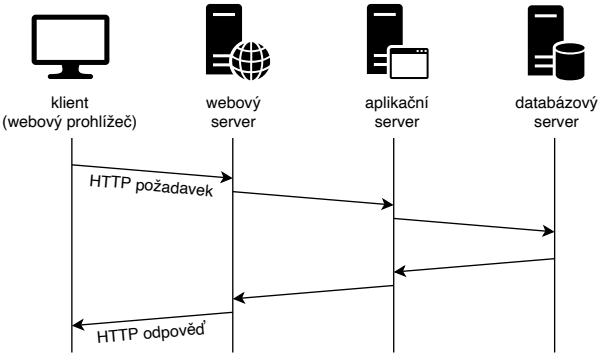
Webová stránka

- www stránka, HTML stránka, web stránka, web, stránka
- dříve dokument v síti Internet, dnes nelze přesně vymezit
- tvořena řadou *webových technologií*
- umístěna na *webovém serveru*, ale není nutné
  - „běžný“ počítač v sítí
  - specializovaný software → webový server
  - komunikace: klient-server architektura
- základní rozdělení:
  - statická
  - dynamická

Statická stránka



Dynamická stránka



Webové technologie

- obecně jakékoliv technologie spojené se službou WWW
- dnes $\rightarrow$ prostředky pro tvorbu webových stránek
- zahrnuje enormní množství technologií
- celá řada klasifikací:
  - základní
  - pokročilé
  - klientské
  - serverové
  - ...
- etapy vývoje webu: Web 1.0, Web 2.0, Web 3.0 (web3)

Webové technologie

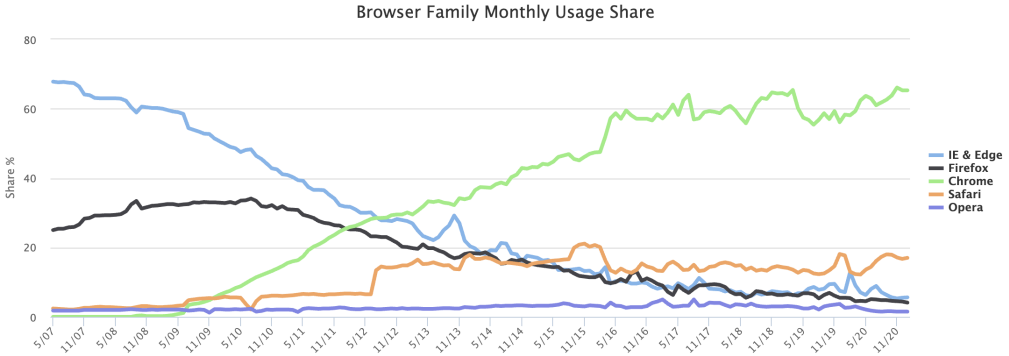
- základní klientské (front-end)
  - HTML, AMP, XML → popis sémantiky
  - CSS → vizualizace stránky
  - JavaScript, WebAssembly → skripty na straně klienta
- základní serverové (back-end)
  - webový server (Apache, IIS, nginx, Node.js, ...)
  - skriptovací jazyk (PHP, .NET, Python, JavaScipt, ...) → skripty na straně serveru
  - databázový server (MySQL, MariaDB, MongoDB, ...)
- front-end, back-end, full-stack vývojář/programátor

Webový prohlížeč

- uživatelské hledisko → nezajímavé
- „renderuje“ webové stránky
- renderování webové stránky = složitý proces
- rozdíl mezi prohlížeči → renderovací jádro

prohlížeč	jádro
Google Chrome	Blink (založeno na WebKit)
Safari	WebKit
Microsoft Edge	Blink (od verze 79)
Firefox	Gecko
Opera	Blink
IE	Trident

Webový prohlížeč: Statistiky používání



Obrázek: Statistika z `https://www.w3counter.com/trends`. Nejpopulárnější Google Chrome, ale ...

Odbočka: Webové vyhledávače

- „přívětivá“ cesta k webové stránce
- indexace = zanesení stránky do databáze vyhledávače
- proces indexace
  - stránku navštíví robot (web crawler)
  - stáhne její obsah
  - stažený obsah je analyzován (ohodnocení stránky, klíčová slova, různé metriky)
  - výsledek uložen do databáze
- optimalizace pro vyhledávače (SEO)

Odbočka: Google

- základem je PageRank algoritmus → důležitost stránky (0–10)
- core algoritmus = PageRank + neznámý algoritmus (zvažující řadu kriterií)
- na výsledek jsou aplikovány filtry
  - Panda (2011), filtrace stránek s nízkou kvalitou
  - Penguin (2012), filtrace stránek s mnoha odkazy
  - Hummingbird (2013), filtrace stránek obsahující spam (analýza textu), částečně nahrazen RankBrain (2015) a BERT (2019)
  - E-A-T (2014), autoritativní domény
  - 2016–2017, orientace na mobilní vyhledávání
  - 2020 – pandemic
  - Page Experience, aka UX (2021)
- core algoritmus a každý filtr je průběžně aktualizován
- jiné vyhledávače mají podobnou architekturu

Odbočka: Přístupnost

- uživatelé (nejen) s handicapem
- přístup = zpřístupnění webové stránky těmto uživatelům
- dáno různými standardy
- státní instituce musí (s výjimkami) splňovat

Webové standardy

- konsorcia
  - W3C (World Wide Web Consorcium), `http://www.w3.org/`
  - konsorcium WHATWG (Web Hypertext Application Technology Working Group), `https://whatwg.org/`
- specifikace (standardy)
- složitý životní cyklus specifikace (`https://www.w3.org/2020/Process-20200915/`)
- důležité:
  - rozpracovaný standard → status *working draft*
  - hotový standard → status *recommendation* (doporučení)
- webové prohlížeče implementují tyto standardy
- realita: ...
- ukázka: `https://caniuse.com/#feat=webp`
- novější standardy nahrazují předešlé

Jazyk HTML (HyperText Markup Language)

- nový způsob standartizace
- 28. ledna 2021 WHATWG reviewer draft → status recommendation W3C
- textový soubor s příponou `.html` (není nutné)
- webová stránka obsahuje:
  - obsah
  - odkazy
  - značky jazyka HTML

Význam HTML značek

- značky přiřazují význam obsahu (udávají sémantiku)
- Q: Je sémantika nutná? A: Ano!
- sémantika má vliv na celou řadu klíčových atributů
  - SEO
  - přístupnost
  - strojové zpracování
- značky HTML neříkají nic o vzhledu (vizualizaci) obsahu (poněkud nepřesné)

Význam HTML značek

- kategorie značek (`https://www.w3.org/TR/html52/dom.html#kinds-of-content`):
  - Metadata
  - Flow
  - Sectioning
  - Heading
  - Phrasing
  - Embedded
  - Interactive

- zastaralé dělení značek:
  - reflektuje výchozí zobrazení značky
  - blokové – jejich použití způsobí vytvoření bloku
  - řádkové (inline) – nepřekročí rozsah řádku

Syntaxe HTML: Element

- (normální) element → *element*, dříve párový element
- syntaxe:

```
<znacka [atribut_1="hodnota_1" ... atribut_n="hodnota_n"]>
obsah elementu
</znacka>
```

- příklad:

```
<a href="https://cs.wikipedia.org/wiki/Douglas_Adams" rel="external" title="více o Douglas
      Adamsovi">
Douglas Adams
</a>
```

Syntaxe HTML: Prázdný element

- *prázdný element*, dříve nepárový element
- syntaxe:

```
<znacka [atribut_1="hodnota_1" ... atribut_n="hodnota_n"]>
<znacka [atribut_1="hodnota_1" ... atribut_n="hodnota_n"] />
```

- v HTML 5 již není znak / povinný (ani zakázaný)
- příklad:

```
<img src="pruvodce.jpg" alt="Obálka průvodce" width="200">
```

Syntaxe HTML

- při renderování jsou ignorovány:
  - komentáře `<!-- 42 -->`
  - vícenásobné mezery
  - zalomení řádků
  - tabulátory
  - neznámé značky
- terminologie:
  - element HTML = součást jazyka HTML → značky, například element `p`, tedy `<p></p>`
  - element (webové stránky) = element HTML a jeho obsah
  - HTML značka = jedna konkrétní značka, například `</p>`

Syntaxe HTML: Atributy

- povinné, doporučené, volitelné, globální
- několik způsobů zápisu:

```
<input type="text" disabled>
<input type="text" value=yes>
<input type="text" value='no'>
<input type="text" value="yes or no">
```

Zanořování elementů

- vytváří vztah potomek-rodič
- značky musí být uzavírány v pořadí LIFO

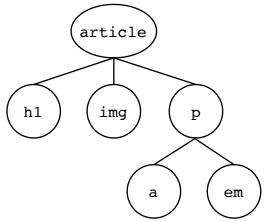
```
<!-- správné zanoření -->
<p> ... <em>trilogie v pěti dílech</em> ... </p>

<!-- chybné zanoření -->
<p> ... <em>trilogie v pěti dílech ... </p></em>
```

Vztah potomek rodič

```
<article>
  <h1>Kniha Stopařův průvodce Galaxií</h1>
  <img src="pruvdce.jpg" alt="Obálka průvodce">
  <p>... <a href="https://cs.wikipedia.org/wiki/Douglas_Adams" rel="external"
  title="více o Douglas Adamsovi">Douglas Adams</a>, ... <em>trilogie v pěti dílech</em>.</p>
</article>
```

- elementy vytvářejí hierarchickou strukturu potomek-rodič



- `h1` je *potomek* `article`
- `p` je *rodičem* prvků `em` a `a`

Pojmenovávání elementů

- pojmenování jedinečným názvem = atribut `id`

```
<div id="content">
  <article>
  ...
  </article>
</div>
```

- pojmenování opakujícím se názvem = atribut `class`

```
<div class="navigation active">
  <article>
  ...
  </article>
</div>
```

- názvy by měly být smysluplné

Základní struktura webové stránky

```
<!DOCTYPE html>
<html lang="cs">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Nadpisy

- `h1`, ..., `h6`

```
<h1>nadpis</h1>
<h6>nadpis</h6>
```

- postupně klesající důležitost
- používáním nadpisů vytváříme *outline* (osnova) webu
- výskyt nadpisů musí být postupný

Strukturální elementy

- Sectioning elements
- elementy určující (základní) strukturu stránky
- strukturální elementy:
  - `header`
  - `nav`
  - `footer`
  - `article`
  - `section`
  - `main`
- restartují outline webu

Odkaz

- element a
- atributy:
  - `href`, URL adresa destinace (relativní, absolutní)
  - `title`, popisek, (důležitý význam)
- kotva (odkaz na element)

```
<a href="#stoparuv-pruvodce">Stopařův průvodce Galaxií</a>
...

<article id="stoparuv-pruvodce">
...
```

Obrázek

- prázdný elment `img`
- různé formáty (WebP, JPG, PNG, GIF, . . .)
- atributy:
  - `src`, URL adresa obrázku (relativní, absolutní)
  - `alt`, popisek obrázku (má vliv na SEO a v případě nedostupnosti obrázku, popřípadě pro čtečky stránek)
- obrázky by měly být vždy optimalizovány pro web

```
<img src="smiley.gif" alt="Smiley face" height="42" width="42">
```

Text v HTML

- `p`, odstavec, žádný (skutečný) text by neměl být mimo odstavec
- `strong`, zvýrazněný text, sémantická důležitost
- `em`, zdůrazněný text
- `small`, krátký dodatečný komentář (např. autorská práva)
- `i`, `b`, `u`, jiný typ nebo jiná výslovnost (např. jiný jazyk, technický pojem), tučný (bez ovlivnění sémantiky), stylistika (píše se jinak, text obsahující záměrnou chybu)
- `br`, prázdný element, řádkový zlom
- `hr`, tématická změna, (nikoliv vodorovná čára!)

Seznamy

- `ul`, seznam s odrážkami
- `ol`, číslovaný seznam
- `li`, položka seznamu
- seznam definic `dl`, položky `dt` (term), `dd` (obecně data ve tvaru název-hodnota, např. otázky a odpovědi)

Další elementy

- audio
- video
- canvas, 2D kreslící plátno

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Formuláře

- elementy pro tvorbu formulářů a jejich prvků
- vytváří obsah → nepřiřazují sémantiku (až na několik výjimek)
- lze programovat vlastní prvky (HTML + CSS + JS)
- `input`, `textarea`, `button`, `select`, ...

```
<form method="post" action="zpracuj-formular.php">
 <fieldset>
  <legend>Osobní informace</legend>
   <!-- pro oddělení je použito p, lze i jinak -->
   <p>
    <label for="jmeno">Jméno:</label>
    <input type="text" name="jmeno" id="jmeno">
   </p>
 </fieldset>
</form>
```

Tabulky

- zobrazení tabulkových dat či dat tabulkové povahy
- webový pravěk: využití tabulek pro layout
- syntaxe:
  - `table`
  - `thead`
  - `tbody`
  - `tfoot`
  - `tr`
  - `th`
  - `td`
  - `caption`
- atributy `rowspan` a `colspan` → roztažení buňky přes řádky a sloupce

Elementy `div` a `span`

- generické kontejnery (bez sémantického významu)
- historie značky `div`

HTML entity

- stránky by měly být v UTF-8 kódování (existují výjimky)
- možnost zapisovat speciální znaky přímo v HTML
- syntaxe:

```
&nazev_entity;
```

- příklad:

```
&copy;
&raquo;
&amp;
  <!-- nejdůležitější -->
```

Validní HTML

- validní = syntakticky (a částečně sémanticky) korektní HTML kód vzhledem k aktuálnímu `DOCTYPE`.
- nevalidní HTML může být renderováno nečekaným způsobem
- velký vliv na SEO
- automatická validace: `http://validator.w3.org/`
- HTML kód by měl být vždy validní!
- sémantická správnost $\rightarrow$ nelze ověřit

Nesprávné používání HTML

- jazyk HTML dává „volnost“
  - sémanticky nesprávný kód se zobrazí
  - syntakticky nesprávný kód se zobrazí
- špatné HTML
  - horší přístupnost a SEO
  - složitá HTML struktura $\rightarrow$ pomalé renderování stránky
  - horší vizualizace

Vzhled webové stránky

- HTML určuje sémantiku
- sémantika částečně určuje vzhled (např. nadpis)
- dodatečná specifikace (ignorována většinou prohlížečů) určuje výchozí vzhled každého elementu
- vizualizace stránky = Cascading Style Sheets (CSS)
- textový soubor s *CSS pravidly* určující vzhled elementů
- obecný princip: oddělení obsahu (HTML) od jeho prezentace (CSS)

Verze CSS

- formálně nic takového neexistuje
- specifikace CSS = sada modulů (specifikací)
- každý modul má svoji verzi označenou jako *level*
- rozšíření předchozí funkcionality = vyšší číslo (např. selektory level 3, level 4)
- zcela nová funkcionalita (FlexBox level 1, FlexBox level 2)
- → potřeba `caniuse.com`

Syntaxe CSS pravidel

```
selektor {
  vlastnost_1: hodnota_x;
  ...
  vlastnost_n: hodnota_z;
}
```

- terminologie:
  - CSS pravidlo
  - selektor
  - deklarační blok
  - deklarace
  - vlastnost
  - hodnota
- poslední středník není nutný (je lepší jej psát), bílé znaky neovlivňují syntaxi

Syntaxe CSS pravidel

- komentáře: `/*` a `*/`
- CSS je třeba udržovat přehledné a organizované!
- formátování CSS pravidel, strukturování pravidel

Začlenění CSS do HTML

1. externí soubor (přípona `.css`)

```
<head>
  <link rel="stylesheet" href="file.css">
</head>
```

2. přímo v HTML stránce (embedded), element `style`

```
<style>
  img {border: 1px solid black;}
</style>
```

poznámka: lze `@import url(file.css)`, pomalé, lepší nepoužívat

3. inline definice, globální atribut `style`

```
<img src="dark.jpg" alt="Dark Vader" style="border: 1px solid black">
```

- Q: Jaký je mezi nimi rozdíl? A: Zásadní.

Konstrukce selektorů: Jméno elementu, zřetězení

- jméno

```
h1 {color: gold;}
```

- zřetězení, zápis pomocí znaku ,

```
h1, h2 {color: gold;}
```

Konstrukce selektorů: id a class

```
/* elementy které mají class="nadpis" */
.nadpis {color: gold;}

/* element který ma id="clanek" */
#clanek {color: gold;}

/* elementy em které mají class="nadpis" */
em.nadpis {color: gold;}

/* element který ma id="clanek", zbytečné, může se hodit pro přehlednost */
article#clanek {color: gold;}
```

Konstrukce selektorů: Kontext elementu

- kontext elementu = místo potomek-rodič hierarchii
- zápis pomocí mezery

```
<!-- HTML kód -->
<h1>Stopařův <span>průvodce</span> po <span class="podnadpis">galaxii</span></h1>

<p>je prvním dílem pentalogie označované jako <span>trilogie v pěti dílech</span></p>
```

```
/* CSS kód */
h1 span {color: red;}

h1 span.podnadpis {color: blue;}
```

Konstrukce selektorů: Přímý potomek (rodič)

- zápis znaku >

```
<!-- HTML kód -->
<article>
  <h1>Nadpis</h1>
  <p>A</p>
  <p>B</p>
  <p>C</p>

  <section>
    <p>D</p>
    ...
```

```
/* CSS kód */
article > p {color: gold;}
```

- vybere jen prvky `p`, které jsou přímými potomky `article`

Pseudo-element a pseudo-třída

- pseudo-element → část stránky, která není určena žádným elementem (první písmeno, první řádek), ale chováme se k ní jako k elementu, zápis s ::
- pseudo-třída → elementy identifikované na základě jejich pozice v HTML nebo vlastnosti, zápis : (jakoby určené třídou)

```
/* pseudo-element, výběr prvního řádku elementu p */
p::first-line {
 color: gold;
}

/* pseudo-třída, výběr prvního potomka elementu p */
p:first-child {
 color: gold;
}

/* pseudo-třída, výběr elementů a na kterých je kurzor */
a:hover {
  color: gold;
}
```

Dědičnost

- klíčový koncept v CSS

```
<!-- HTML kód -->
<p>Stopařův průvodce Galaxií, jehož autorem je Angličan Douglas
Adams, je prvním dílem stejnojmenné pentalogie označované jako
<em>trilogie v pěti dílech</em>.</p>
```

```
/* CSS kód */
p {color: gold;}
```

- potomci přejímají vlastnosti svých rodičů
- mění se `p` i `em`
- ne vše se dědí
- vynucení dědičnosti: `inherit`
- dědičnost usnadňuje práci

Kaskáda

- více zdrojů CSS
- elementy mohou být vybrány různými selektory

```
<!-- HTML kód -->
<article id="prvni-clanek"> <!-- id by se nemělo používat -->
  <h2 class="nadpis"></h2>
</article>
```

```
/* CSS kód */
h2 {...}
article h2 {...}
.nadpis {...}
#prvni-clanek h2 {...}
```

- kolize CSS pravidel
- řeší kaskáda → uvedené způsoby nejsou ekvivalentní

Kaskáda

- deklarace různých vlastností → spojeny
- deklarace stejných vlastností → *pravidlo (princip) kaskády*:
  1. *specifičnost*, více specifický selektor má přednost před méně specifickým selektorem (dáno standardem, začátečníci o něm mnohdy ani neví)
  2. pořadí, později uvedené pravidlo má přednost
- lze obejít pomocí důležitosti (`!important`, ideálně nepoužívat!)

Výpočet specifičnosti

- specifičnost = číslo ve tvaru `a b c` (lexikální uspořádání)
  - `a` = počet `id` v selektoru
  - `b` = počet `class`, výběru atributů a pseudo tříd v selektoru
  - `c` = počet elementů a pseudo-elementů v selektoru
- inline styl vetší specifičnost než jakýkoliv jiný autorský styl (pro jednoduchost `1 0 0 0`)
- `!important` obejití specifičnosti, největší specifičnost (pro jednoduchost `1 0 a b c`)
- univerzální selektor: `0 0 0 0 0`

Kaskáda: Příklad

```
<!-- HTML kód -->
<div class="druhy prvni"></div>
```

```
.prvni {color: blue;}

.druhy {color: red;}
```

- aplikuje se druhé pravidlo (rozhoduje pořadí)

Kaskáda: Příklad

```html
<!-- HTML kód -->
<article id="prvni-clanek"> <!-- id by se nemělo používat -->
  <h2 class="nadpis"></h2>
</article>
```

```css
/* specifičnost: 0 0 1 0 1 */
#prvni-clanek h2 {color: blue;}

/* specifičnost: 0 0 0 1 0 */
.nadpis {color: red;}
```

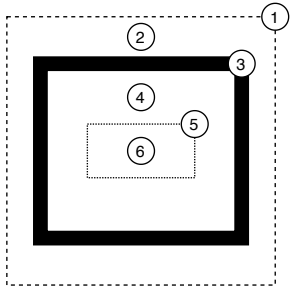
- aplikuje se první pravidlo (rozhoduje specifičnost)
- častý zdroj chyb

Zápis hodnot

- hodnota z výčtu
  - výčet přípustných hodnot pro každou vlastnost
  - např. `bold`, `block`, ...
- textový řetězec
  - apostrofy, uvozovky → výběr dle výskytu druhé symbolu
  - lze provádět escape pomocí \
- číslo (celé, desetinné)
  - faktor, bez jednotky, udává změnu, například `1.1`
  - délka, s jednotkou, například `12em`, případě záporné hodnoty (pokud to má smysl)
  - absolutní vs. relativní jednotky
  - 0 se píše vždy bez jednotky
- barva

CSS box model

- určuje velikost elementu



Obrázek: Box model elementu. ① venkovní hrana elementu, ② venkovní okraj (margin) elementu, ③ rámeček (border) elementu, ④ vnitřní okraj (padding) elementu, ⑤ vnitřní hrana elementu, ⑥ obsah elementu

Základní vlastnosti

- box model
- layout
- pozicování
- text

Responzivní design

- 2007 začátek revoluce
- různá zobrazovací zařízení
- dnes nutnost
- adaptivní vs. responzivní design
- fluidní layout, beakpointy, kombinace

CSS preprocesory

- pomalý vývoj CSS → CSS preprocesory
- skriptovací jazyk (preprocesor) → transpilace → CSS
- přináší
  - funkce
  - proměnné
  - řízení běhu programu
  - pohodlnější syntaxi
- příklad: SASS, LESS, Stylus
- dnes spíše konzervativní technologie
- nový trend postprocesing

JavaScript

- původní název LiveScript
- plnohodnotný skriptovací jazyk
- primárně určen pro skriptování na straně klienta (pouze část jazyka)
- velmi populární (např. Node.js, Adobe Acrobat, vývoj aplikací)
- interpretován webovým prohlížečem
- ECMAScript standard jazyka, JS implementace ECMA Script
- každoroční update, číslovány dle roku
- prohlížeče implementují standardy (analogická situace jako u HTML a CSS)

JavaScript

- základní funkcionalita
  - manipulace s HTML a CSS
  - události
  - AJAX
- nemá žádné grafické schopnosti
  - umí generovat HTML
  - spojení s elementem `canvas`
- omezení na straně klienta
  - částečně potlačeno HTML 5 API

Začlenění JS do HTML

1. externí soubor (přípona `.js`)

```
<head>
  <script src="file.js"></script>
</head>
```

2. přímo v HTML stránce (embedded), element `script`

```
<script>
  alert("Hello world");
</script>
```

Použití JS na webové stránce

- webová stránka reprezentovaná jako DOM (Document Object Model)
  - stromová struktura
- JS má stránku přístupnou v objektu `document`
- JS disponuje metodami pro výběr prvků na stránce, jejich přidávání a odebírání

WebAssembly

- WASM
- transpilace z programovacího jazyka (C, C#, Rust, ...) do instrukční sady virtuálního stroje prohlížeče
- „binární soubor“ pro webové prohlížeče
- velká rychlost
- není náhrada za JS

Webové aplikace

- obecně stránky s masivním nasazením skriptů (na straně klienta nebo serveru)
  - nelze jednoduše vymezit
- myšlenka: aplikace je webová stránka
  - mobilní aplikace (např. Apache Cordova, React native)
  - desktopové aplikace (např. Electron)
  - progresivní webové aplikace (PWA)
- platformová nezávislost $\rightarrow$ horší UI a UX

Strana serveru

- webový server
- skriptovací jazyky
- webové databáze
- MVC architektura
- REST API
  - GET/PUT/POST/DELETE

AJAX

- modifikace části stránky bez nutnosti jejího celého načtení
- asynchroní komunikace se serverem + JS

Odbočka: Knihovny a frameworky

- poskytují pokročilejší funkcionalitu
- vystavěné na základních webových technologiích
- přirozený vývoj
- například:
  - Bootstrap, miniCSS, materialize, ...
  - jQuery, Prototype, ...
  - React, Angular, Vue, Lit (Google Polymer), ...
  - React native, Meteor, Tauri, ...
  - Nette, Symphony, Laravel, Falcon, Django, Flask, ...

CMS systémy

- content management systems
- systémy pro správu obsahu
- blogy, e-shopy
- WordPress, Drupal, Joomla!

Sémantický web

- původně označován jako Web 3.0
- Tim Berners-Lee (2001): web = obrovské množství stránek
- sémantický web = přidání významu obsahu pomocí metainformace
- zápis pomocí: mikroformáty, mikrodata, RDF, OWL, JSON-LD