

Komunikace na internetu

Počítačová síť je lokální propojení alespoň dvou zařízení, které spolu umí komunikovat. Počítačová síť může vlastně obsahovat další počítačové sítě a může vznikat jakýsi hierarchický systém. Tyto počítačové sítě komunikují nějakými protokoly a rozumí si. Probíhá tedy výměna informací a používají stejné technické prostředky.

V komunikaci rozdělujeme dva typy komunikací a to **peer to peer** a **client-server**. Peer to peer je způsob komunikace, kdy jsou si obě dvě strany rovný a zároveň si nepřikazují. Client-server je způsob, kdy je jedna strana autoritativní a je cílena různými klienty. Server zpracovává jejich příkazy a může je i odmítnout.

Internet je celosvětový systém navzájem propojených počítačových sítí. Propojení sítí je realizováno různými způsoby (např. drátově a bezdrátově), které tu nebudeme hloubkově probírat.

World wide web a domény

Hypertextový dokument je dokument, který může být vázán na jiné dokumenty. Propojují se tzv. **hyperlinky**. Tento dokument často není lineární, ve slova smyslu, jako jsou obyčejné dokumenty, které můžete číst ze shora dolů. Tento dokument může být definován tak, že se něco může zobrazit jinde, než bylo definováno.

World wide web je soustava propojených hypertextových dokumentů. Zjednodušeně se jedná o mapu propojených dokumentů, které lze navštěvovat pomocí **webového prohlížeče**. Používá se protokol HTTP (a HTTPS) ve formátu HTML. Pro adresaci ve WWW používá URL.

Doména je částí URL adresy a určuje jednoznačný identifikátor nějakého místa (přesněji počítačové sítě či server) na internetu. Vznikla z důvodu, že komunikace pomocí IP adres je složitá:

- pro lidi je složité si zapamatovat (v případě IPv4) 4 čísla pro každý web, který navštěvují
- v případě HTTP se otevírá pouze jeden port - tím pádem by na jedné IP adrese mohl být jeden web. To by šlo obejít různými dalšími porty či překonfigurace webů např. na 23.201.102.3/alza

Doména se rozděluje do několika částí, které jsou oddělené tečkami, všechny části musí být ASCII a ignoruje se velikost písmen. Základní je generická (doména 1. řádu, top level domain = **TLD**), která určuje úplný "zdroj" stránky, která je po poslední tečce. Většinou se jedná o země a nově i další věci. Jedná se například o `.cz`, `.uk`, `.fr`, `.io`, `.website`. Je možné si svojí vlastní generickou doménu nějak zařídit, musíte ale získat spoustu povolení a spřátelit si různé instituce. Stojí to velmi velké peníze, protože své TLD musíte spravovat, například pomocí WHOIS a zařizovat pro ně domény, administrace, ...

Nejčastější další doména je doména 2. řádu, která je nejčastější koupě. Určuje nějakou společnost, entitu či prostě danou stránku. Nejčastěji je doména kupována na rok, kdy první rok je levnější než ty následující. Jedná se např. o `seznam.cz`, `ssps.cz`, `google.com`.

Všechny následující domény lze označovat jako x. řádu a často se jim říká **subdoména** (sub - menší část z něčeho). Často je to nějaká podslužba dané stránky. Např. se jedná o `email.seznam.cz`, `skripta.ssps.cz`, `calendar.ssps.cz`. S domény větším jak 4. řádu se v podstatě v praxi nelze potkat.

Internationalized Domain Names (IDN) je standard, který dovoluje, aby se v doménách mohli vyskytovat libovolné znaky abecedy z celého světa. Doména je stále však tvořena ASCII, jen jsou znaky překódovány. Například znak `á` se přetvoří na `xn-1ca`. Pro českou doménu je vypnuto z bezpečnostních důvodů (přepis může přesměrovat na phishing stránku) a taky z konfúze, protože pro jednu doménu by bylo nutné koupit další pro zaručení bezpečnosti. Společnost CZ.NIC, která českou doménu spravuje pravidelně dělá dotazník, kde se na tuto technologii ptá. Více si lze přečíst [ZDE](#).

URL adresa

URL (uniform resource locator) je řetězec znaků s definovanou, který se používá pro adresaci na WWW. Když chceme webovou stránku navštívit, tak nechodíme pouze na doménu, ale můžeme určovat další věci, nejčastěji však například cestu, na které najdeme nějaký zdroj. Je rozdělen do částí a platí pro ně různé pravidla.

Části:

- **protokol** (nezáleží na velikosti písmen, povinná část, jen ji prohlížeč automaticky doplňuje) - např. `HTTPS`, `HTTP`, `FTP`, ...
- **doména** (nezáleží na velikosti písmen, povinná část)
- **port** (povinná část, jen ji prohlížeč automaticky doplňuje) - číslo od `0` do `65535`, pro `HTTP` se používá `80`, pro `HTTPS` `443`
- **umístění na serveru** - určuje místo souboru / zdroje na serveru, např.
`o-skole/pedagogicky-sbor.html`
 - lomeno většinou reprezentuje přestup do jiné složky (najdi soubor `pedagogicky-sbor.html` v složce `o-skole`), ale nemusí to tak být
- **parametry** - určuje parametry se kterými může poté stránky či server pracovat, např.
`?p=1` a `?name=Matěj Cajthaml&age=59`.
 - parametry vždy začínají `?`, aby se rozeznali od umístění na serveru - ten otazník tedy obsahovat nemůže.
- **kotva** určuje jedno místo na stránce, na které se při načtení přeneseme, více si o něm povíme v dalších lekcích, např. `#seznam`

Často se k URL zapisuje i `WWW`, které k ničemu neslouží a je to normální doména 3. řádu. Ukázky správných URL adres:

- `https://cs.wikipedia.org/wiki/Česko`
- `https://www.jakpsatweb.cz/html/url.html`
- `https://www.luxor.cz/checkout/delivery_and_payment`
- `https://ssps.cz/news/read/452`
- `https://www.preslovinny.cz/kategorie/technologie/`
- `https://www.postovnisporitelna.cz/portal/ucty/postovni-ucet`

URL se postupně vyvíjí, prošli jsme pouze základy a více informací naleznete na dokumentaci [ZDE](#).

Vyhledávání na webu

Webový vyhledávač je automatický systém vyhledávající informace na WWW. Automaticky tedy server prochází (crawluje) stránky a zjišťuje si z nich informace, které poté uživateli, který vyhledává, ukazuje. Z různých odkazů se tvoří mapa (pavučina) WWW.

Webový katalog je seznam odkazů, které vedou na různé weby. Je spravován ručně.

Webové vyhledávače lze omezit pomocí souboru robots.txt, ve kterém lze zakázat, aby tento web navštěvovali (nebo omezit nějaké URL). Etické crawleři toto Vaše rozhodnutí budou respektovat.

Na části WWW, kam se vyhledávače nemohou dostat, říkáme, že je to **hluboký web (deep web)**. Jsou to jednoduše veškeré stránky, které potřebují přihlášení. Například se jedná o internetové bankovníctví, Vaše zprávy na Instagramu, či cokoliv obdobného. **Jedná se o největší část WWW.**

Hluboký web se často zaměňuje s **temným webem (dark webem)**. To je web, na který se nedá dostat pomocí klasických technologií a je potřeba speciální software, speciální připojení, připojení v určitý čas a různé další ochrany. Dark web stojí na anonymitě pomocí různých technologií, které lze při určitých podmínkách obejít. Na dark webu se často konají ilegální prodeje a další věci.

Do obyčejného webu patří:

- stránky školy
- wikipedia
- Facebook profil (dle nastavení soukromí)

Do hlubokého webu patří:

- virtuální škola / moodle
- Bakaláři
- zprávy v Messengeru / Instagramu
- zprávy v e-mailovém klientovi

Do temného webu patří:

- tzv. onion web

Protokoly pro web

Pro webové stránky existuje nespočet protokolů, které jsou pro jeho funkčnost nutné. My si ukážeme ty nejzákladnější z nich. Více o nich jste se měli dozvědět v předmětu PSI v 1. a 2. ročníku. Protokol je vlastně takový definovač toho, jak dvě věci mezi sebou budou komunikovat, aby si rozuměli. Mají vlastní syntaxe, sémantiku a např. synchronizaci.

HTTP

Zkratka pro Hyper Text Transfer Protocol. Je určen pro přenos hypertextových dokumentů (definované v WWW). Běží na portu 80, ale lze jej zřídit na jakémkoliv. Moderně se nepoužívá

jenom pro přenos těchto dokumentů, ale i pro další věci a stal se takovým hlavním protokolem - je ve všech odvětvích IT.

HTTPS

Zkratka pro Hyper Text Transfer Protocol. Je to zabezpečená nástavba HTTP, používá asymetrické šifrování pomocí klíčů a certifikátů. Data jsou po cestě šifrována. Používá se port 443. Dříve se používal pouze na důvěrné informace, nyní se používá na vše (je nutný všude).

IP

Zkratka Internet Protocol, určený na směrování paketů na internetu. Funguje na síťové vrstvě (komunikace mezi více sítěmi). Existují dvě verze, jedna je IPv4, který je vyčerpaná. Proto vznikla nová IPv6, která má i mimo počet adres více výhod.

Formát IPv4: `192.168.0.1`

Formát IPv6: `2001:718::fec9:ca5`

DNS

Zkratka Domain Name System. Jedná se o převod doménového jména na nějaké IP. Jedná se o hierarchický systém, který určuje že např. `seznam.cz` se převede na `77.75.76.3`. Může se totiž stát, že nějaké subdomény mohou vést na jiné servery, aby se všechna doména nemusel nacházet na jednom serveru, což je nebezpečné. DNS neslouží jen na převod jmen, ale například i na určování e-mailových serverů a dalších záznamů pro počítače.

DNS se musí cachovat, tedy ukládat, aby se nemuseli počítače při připojení vždy ptát, kudy se mají vydat aby navštívili tuhle doménu. Nejbližší uložisko DNS je například v routeru, u ISP a dál. DNS server si ukládá svoje záznamy o daných doménách a komunikuje s ostatními, se kterými sdílí svoje záznamy.

FTP

Zkratka pro File Transfer Protocol. Slouží k přenášení souborů mezi počítači a servery. Podporuje nespočet operací, která každá dělá něco jiného. Využívá model klient-server a pro připojení je potřeba FTP klient, který je často i v prohlížeči.

Známí FTP klienti jsou např.: WinSCP, FreeCommander, FileZilla, ...

Digitální stopa

Digitální stopa jsou záznamy na internetu, které vytváříme při procházení WWW a internet. Tyto záznamy se nacházejí jak v **prohlížeči** (klientovi) tak i na **serverech**. Patří do toho očekávané věci - příspěvky, komentáře, objednávky a další věci, které logicky musí server uchovávat.

Digitální stopa patří do **cenných informací**, což jsou informace, které mohou společnosti využívat k lepšímu marketingu a obecně k cílení. Dělíme je na **vědomé** (výše popsané, např. příspěvky, ...) a **nevědomé** (historie procházení, samotná návštěva stránky, kde na webu

klikáte, ...). Tyto informace nemusí být určité použité k špatné věci (tj. zanechané nepříteli), ale **mohou být využity k zlepšení služby**.

Pro minimalizaci zanechávání stopy je dobré si nastavit prohlížeč tak, aby **posílal webům co nejméně informací**. Poté je dobré při první návštěvě webu nastavit cookies tak (podle cookies law od EU), že **nechcete využívat soubory ke sledování**. Proti některé stopě se nelze bránit - jak se říká, to co nahrajete na internet tak bude velmi pravděpodobně navždy.

K tématu patří taky **metadata**. Jsou to data, které popisují data (tj. **data o datech**). Jako ukázka můžou být například fotografie, kde se často ukládá, kdo je vyfotil, kdy byly vyfoceny, kdy byl soubor naposledy upraven a podobně. Tyto informace existují skoro ke každým datům - souborech, příspěvek, e-mailech, ...

K realizaci toho, jak moc jen samotný prohlížeč posílá na internet informací (metadat), doporučuji navštívit stránku [AmlUnique](#), na které si můžete ověřit jak unikátní jste. Webové stránky mohou například skenovat nainstalovaná písma a podle toho je velmi jednoduché kohokoliv poznat.

Prohlížeče

Webový prohlížeč je software, který je určený **k procházení WWW**, zjednodušeně webů. Jeho cílem je komunikovat se servery v různých protokolech, nejčastěji pomocí HTTP/S. Prohlížeč však má různé věci, které s procházením nemusí přímo souviset. Často řeší i jiné úlohy (záložky, kalendáře, e-maily, historii) a jsou nastavitelné. Hlavní částí prohlížeče je **vykreslovací (renderovací) jádro**, které slouží pro vykreslování stránek v jazyce HTML, se kterým se tento rok budeme učit.

Nejčastější jádra jsou **Gecko** (Mozilla FireFox), **Webkit** (Safari), **Blink** (Google Chrome) a např. Trident (Internet Explorer). Jádra se hlavně liší v tom, jak se webové stránky vykreslují.

Chromium je vlastně takový "prohlížeč" bez ničeho na víc. V podstatě obsahuje jen vykreslovací jádro **Blink**, které obsahuje Google Chrome. Je to velmi oblíbený nástroj, který se používá pro vykreslování webových stránek bez zbytečných věcí na více. **Velmi mnoho prohlížečů na něm staví svůj základ** a nepoužívají tím své jádra - jedná se např. o Vivaldi, Operu, MS Edge, Brave a spoustu další. Chromium je **opensource** a tedy každý do něj může nahlížet a upravovat jej. Chromium lze používat samostatně jako náhrada Google Chrome, ale neobsahuje například ani synchronizaci mezi zařízeními. **Hlavní nevýhodou Chromia** je to, že jej využívat již tolik prohlížečů, že se stal Google gigantem, který může velmi jednoduše ovlivňovat vývoj webu - **velmi se snížila kompetitivnost na tomto trhu**. Jediné použitelné možnosti je Gecko a Blink, kdy Webkit od Safari s weby velmi stagnuje.

Moderní prohlížeče by se měli co nejvíce přizpůsobit potřebám uživatele, jedná se např. o:

- úpravy barev
- úpravy velikosti písmen a fontů
- umí ochránit uživatele před reklamou či sledovači
- má anonymní režim
- je rychlý a nebere zbytečné zdroje počítače
- podporuje rozšíření
- má synchronizaci

Hostování

Hosting je služba, která Vám dovoluje nějak **pronajmout hardware**, který je schopen komunikovat s internetem a nebo např. něco vypočítávat. Existuje nespočet typů hostingů, my si ukážeme jen základní.

Serverhosting je pronájem celého serveru (počítače), na kterém si lze spustit jakýkoliv software. Zákazník má přístup k celému serveru a může složit úplně na cokoliv - např. hostování stránek, her či různé vnitřní systémy.

Webhosting je pronájem místa na cizím serveru, který je určen na hostování stránek. Je to omezené místo a jsou omezeny i další parametry. Server a web spravuje daná organizace, většinou nejsou práva pro nějakou konfiguraci.

Freehosting je nějaký typ hostingů, který je zdarma. Často chybí technologie, databáze a další služby. Vkládá se například reklama nebo je jinak omezena. Používá se taky jako ukázka služeb daného hostingů. Často je k dispozici i doména 3. řádu zdarma.

Mezi další služby, které hostingy nabízí, patří:

- doména
- certifikáty
- databáze
- e-mail
- výpočetní služby

Pro výběr hostingů existuje nespočet kritérií. Je nutné odhadnout, kolik webová stránka zabere místa a jaké technologie budou potřeba. Mezi známe české hostingy patří:

- Roští (nabízí Node.js hosting přímo)
- Heroku (dovoluje spouštět Node.js aplikace)
- Wedos
- Endora
- Forpsi
- Stable
- Active24

Certifikáty

Certifikát je určen pro jednoznačnou identifikaci entity (stránky, služby, uživatele) na internetu. Stojí na **asymetrickém šifrování** a podpisech. **Podepsání** je vlastně akt, kdy je certifikát podepsán (jiným) certifikátem a tím je schválena jeho existence - pravost.

Certifikát obsahuje **veřejný (encrypt) klíč**, který slouží k šifrování dat. Poté obsahuje **soukromý (decrypt) klíč**, který slouží k rozšifrování dat. K tomu obsahuje ještě **expiraci** daného certifikátu - jedná se o datum, po kterém by se certifikátu nemělo důvěřovat. Obsahuje další informace o tom, kdo jej **podepsal (vydal)**, **pro koho je určen**, pro jaké je domény a podobně.

Certifikační autorita je společnost, která podepisuje cizí certifikáty za určitý obnos. Poté je certifikát platný a kdokoliv si jeho pravost díky klíče certifikační autority ověřit. Tyto služby jsou často zpoplatněné (poskytují více věcí a služeb) a nebo i zdarma, jako např. [Let's Encrypt](#). **Certifikát může autorita zneplatnit.**

Důvěryhodnost certifikátu je založen právě na tom, že si kdokoliv může **zkontrolovat kdo jej podepsal**. Klíče pro asymetrické šifrování jsou natolik bezpečné na normální počítačích (na kvantových tomu tak nemusí být), **že je nemožné je v rozumném čase dešifrovat i se všemi PC na světě**. Přesto se používá životnost certifikátů, aby se certifikáty často měnili. Když si sám vytvoříte certifikát, tak si jej sám můžete podepsat svým certifikátem. Jedná se tzv. o **self-signed** certifikáty, které se hodí na osobní použití a prohlížeče a další služby jej berou jako nebezpečný.

Certifikáty se používají všude - např. v HTTPS, VPN, SFTP, SSH a spoustě dalších službách.

Pro lepší představení certifikátů doporučuji tento materiál k dispozici [ZDE](#).

Autorský zákon

Webové stránky jsou chráněny autorským zákonem. Jedná se tedy i o backendové projekty, které často uživatel nevidí. Často se na stránkách uvádí formulace

`2022 © Cajthaml, All right reserved`, která je vlastně jen pro upozornění. **Bez uvedení licence je stránka stále pod autorským zákonem.** Vašeho práva jako autora díla není možné se vzdát. Tento termín je však takovým standardem, že je velmi dobré jej tam psát.

Před používáním jakékoliv věci je dobré zkontrolovat **zdroj, licenci a podmínky**, pod kterými můžete danou věc používat. V žádném případě licenci neignorujte - můžete z toho mít velké problémy.

Pro ulehčení licencování věci existují předpřipravené licence, které definují jak se věci mohou používat již předem. Jedná se například o [Creative Commons](#), které definují různé kombinace licencí, které povolují a zakazují různé věci. Jedná se například o **zákaz používat věc pro komerční využití**, nebo že musí po úpravě díla být použita stejná licence. Druhá věc je **fair use**, kde použijete věc v zájmu sebe, ale však uvedete správně zdroj - jedná se např. pro školní účely, nemusíte ale daného vlastníka o použití informovat.

Správa obsahu

Content Managment System (CMS) je systém pro správu obsahu. Jedná se o nějaký **redakční** nebo **publikační** systém, ve které pomocí uživatelského prostředí lze editovat obsah. Existuje pro různé jazyky a platformy.

Jedná se např. o: WordPress, Strapi, Joomla!, Drupal a další.

HTML

Hyper Text Markup Language, ve zkratce HTML je značkovací jazyk. Nejedná se tedy o klasický programovací (jako C#) nebo skriptovací (JavaScript/Lua) jazyk, ale o jazyk, který jenom definuje určité data a to v XML s vlastními definicemi tagů. Technologií XML se v předmětu WBA zabývat ale nebudeme – není nutné si to ani pamatovat.

HTML bylo představeno v roce 1990. Od té doby vznikli různé verze, ta nejdůležitější je HTML5 z roku 2012, kterou se ve WBA budeme zabývat. Tato verze se hlavně zaměřila na to,

aby šel multimediální obsah spouštět na webu. Cílilo se i na to, aby webové stránky byly přehledné i pro počítače a daly se jednodušeji crawlerovat, resp. indexovat.

HTML se používají zejména pro webové stránky, protože definují různé tagy, které vytváří na stránce různé prvky, které může uživatel vidět. Tyto soubory mají obvykle příponu `.html`.

Komentáře jsou velmi potřebné všude. V HTML se zapisují jedním způsobem a to pomocí sekvence `<!-- komentář -->`, kde komentář může být cokoliv. Tyto komentáře jsou víceřádkové.

Soubory na webových stránkách

Na webových stránkách máme různé soubory a některé mají speciální význam. My si ukážeme jen hrst nejpodstatnějších z nich. Z předchozí stránky známe soubory HTML.

Jeden z nejdůležitějších souborů je tzv. index soubor. Jedná se o soubor, který se jmenuje index a má jakoukoliv příponu. Tento soubor se (v základním nastavení většiny webových hostingu) načte právě tehdy, když přistupujeme v URL na složku. Při přístupu server proskenuje složku a dle priority (obvykle např. `php > html > txt > ...`) zobrazí právě indexový soubor.

Na webových stránkách chceme například i obrázky. Nejzákladnější typy jsou rastrové obrázky, které uvnitř ukládají jednotlivé pixely a v případě komprimace nějaké zmenšené data. Druhé jsou vektorové obrázky, které místo jednotlivých pixelů ukládají rovnice, podle kterých se obrázek na počítači vykreslí. Rovnice jsou různého typu a způsobují, že se obrázky mohou nekonečně zvětšovat bez ztráty kvality. Na druhou stranu bývají (když jich máte desítky na webových stránkách), pomalé, protože se musí CPU zapojit v počítání. Pro rastrové obrázky známe přípony (resp. typy) `.jpg`, `.png`, `.gif`, `.webp` a pro vektorové `.svg`.

Soubory na webových stránkách vždy pojmenováváme bez diakritiky a speciálních znaků, bez mezer a nejlépe s malými písmeny. Často mají být názvy krátké a výstižné. Důvod je dobrý – protože se soubory zobrazují v URL, často je nutné aby se v nich uživatel vyznal a případně je byl schopen i napsat. Diakritika je špatná zejména na uživatele z jiných zemí (asi byste nechtěli psát čínštinou názvy url). Špatné názvy mohou generovat i řadu problému v prohlížeči.

Tagy a atributy

Tag

Tag je značka v HTML. Jména jsou rezervovaná a určují obsah a často i chování prvku na stránce. Definujeme tím tedy to, co se v prohlížeči vykreslí. Tyto tagy se vždy zapisují v špičatých závorkách – `<tag>`.

Výše představený tag je nepárový a není moc používán. Existuje ale párový tag, který určuje že něco někde začíná a někde končí (vymezuje kde a co). Skládá se ze dvou tagů, kde jeden je počáteční a druhý koncový. Koncový poznáme tak, že po úvodní špičaté závorce se nachází lomeno. Takový zápis tedy vypadá následovně `<tag></tag>`. Mimo tento zápis daný tag, pokud nebyl dříve definován, neplatí.

Jednotlivé tagy lze spojovat dohromady — vkládat je do sebe, např.

`<tag1>uz <tag>nevím</tag> co</tag1>`. Tento zápis znamená, že vše bude označené `tag1` a slovo `nevím` uvnitř bude označeno tagem `tag`. Nějakým nedopatřením se však může stát, že se tagy pokazí a nastane křížení tagů. To znamená, že nějaký párový tag začal uvnitř jiného párového tagu, ale ukončí se až někdy mimo něj. Taková ukázka je např.

`<tag1>uz <tag>nevím</tag1> co</tag>`. V tomto případě má prohlížeč několik možností. První z nich je ta, že se tagy pokusí opravit. Tento zapsaný tag by se například opravil následovně: `<tag1>uz </tag1><tag1><tag>nevím</tag></tag1><tag> co</tag>`. Při těchto změnách se Vám ale velmi může rozbít web (a případně i styly, kterými se budeme zabývat později). Proto se křížení tagů musí vyřešit již při vytváření stránky.

Další důležitý pojem je `element`. Element je kompletní celek, tedy párový tag a jakýkoliv obsah uvnitř něj. Obsah může být cokoliv — další tagy, text či cokoliv jiného.

Tagy jsou definované přesně a my je nebudeme využívat k jiným účelům, než byly vytvořeny.

Atributy

Jakýkoliv tag má atributy. Atributy definují nějaké další data, které popisují přesnější chování prvku. Některé atributy definují chování i pro děti (tj. pod tagy) daného tagu. Tyto tagy jsou většinou speciální pro daný tag a nikde jinde nefungují. Existují však i globální atributy, které lze nastavit v podstatě na cokoliv. Atributy se vždy určují u počátečního tagu. Atribut má vždy své jméno a hodnotu.

Atributy zapisujeme následovně:

```
<tag atribut="hodnota">text</tag>
```

Pro lepší pochopení ukážeme atributy na tagu `a`, který dle HTML5 definuje na stránce odkaz, na který lze kliknout. Pro určení odkazu používáme atribut `href`. Pro vytvoření odkazu na Google by kód vypadal následovně:

```
<a href="https://www.google.com">odkaz na google</a>
```

Mezi zmíněné nejpoužívanější atributy se řadí `id` a `class`. Oba dva slouží na stylování a skriptování stránek. Používají se na vybírání prvků na stránce pro další práci. Identifikátor, neboli atribut `id`, určuje jeden prvek na stránce s daným jménem. Na stránce může být pouze jedenkrát. Třídy, neboli atribut `class`, je seznam tříd, které mohou být definovány více prvkům.

Základní struktura HTML

Základní struktura HTML slouží k definici stránky. Tato struktura se musí používat na všech stránkách (výjimky v podstatě neexistují). Jedná se o základních šest tagů, do kterých se vkládají další. My si jednotlivé tagy ukážeme a popíšeme.

DOCTYPE

Tento tag slouží k určení, že pracujeme s HTML a s verzí HTML5. Tento tag je docela nový a pro ostatní verze (kterými se zabývat nebudeme) je jiný. Tento tag musí být vždy jako první tag stránky z důvodu, že když se stránka načítá, aby na ně prohlížeč mohl ihned reagovat. Pro HTML5 používáme zápis `<!DOCTYPE html>`.

HTML

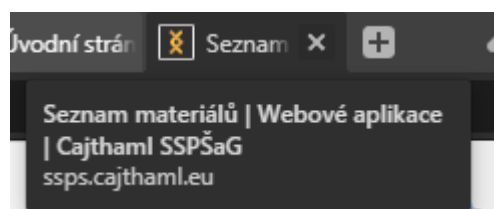
Definuje nám, odkud kam platí to, že píšeme HTML. Pomocí HTTP se posílají různé soubory a proto musíme definovat kořenový element. Uvnitř se nacházejí veškeré další tagy. Velmi důležitý je jeho atribut lang, který určuje jaký jazyk se na stránce primárně nachází. Z tohoto tagu prohlížeče čtou to, zda mají zobrazovat možnost překladu. Pro anglický jazyk používáme en, pro český jazyk cs. Tento atribut je velmi klíčový a proto jej musíte správně definovat v každé stránce. Seznam všech zkratk lze nalézt [ZDE](#).

HEAD

Obsahuje definice stránky, které přímo nezobrazují prvky na stránce. Určují ale její vzhled, chování a další metadata. Nic zde definované by se v prohlížeči nemělo přímo vykreslovat.

TITLE

Určuje, jaký nadpisek aktuální stránka má. Vkládá se do hlavičky (tagu head). Tento nadpisek mohou používat crawleři a roboti, ale zobrazuje se i v "tabu" stránky. Tag může obsahovat pouze text.



META

Tyto tagy určují specifické vlastnosti stránky a vkládají se do hlavičky. Jedná se například o kódování textu, šířku pro různá zařízení, SEO či responzivitu. Tento tag má spoustu atributů, ale většina se definuje pomocí atributu name, který určuje to, co se nastavuje a content, což je jeho hodnota.

Pro české stránky většinou definujeme kódování textu na hodnotu UTF-8 pomocí atributu charset. Pro základní fungování stránek na všech zařízeních definujeme metatag s atributem name s hodnotou viewport a content na "šířku zařízení". Ukázka těchto tagů je níže.

BODY

Nejjednodušší z uvedených tagů. Jeho existence je prostá — vkládáme do něj vše, co se na stránce vykreslí, tedy jednotlivé prvky.

Základní struktura tedy vypadá takto:

```
<!DOCTYPE html>

<html>

  <head>

    <title>Smíchovská střední průmyslová školy a gymnázium</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

  </head>

  <body>

    <p>Dobrý den!</p>

    <p>Vyzkoušejte <a href="https://google.com">tuto úžasnou stránku</a>!</p>

  </body>

</html>
```

Když byste si tuto stránku uložili do souborového systému Vašeho počítače a otevřeli jej v prohlížeči, uvidíte dva řádky textů. Na jednom pozdrav a na druhém text s odkazem. Taký stránka obsahuje titulek, který můžete vidět v prohlížečovém tabu. Stránka by taktéž fungovala na všech zařízeních.

Základní tagy

Paragraf — **p**

Určuje oblast, kde se nachází text. Tento tag by se měl používat pro každý text, který najedete na stránce. Pro lepší formátování existují další tagy, které označují část paragrafu. Nemělo by se nikdy stát, že text bude existovat samostatně např. v tagu `b`. Paragraf má obvykle od prohlížeče již nějaké formátování a to takové, že pod ním je místo, aby se daný paragraf rozdělil od ostatních. Zkratka je odvozena od slova paragraph.

```
<p>Kočka není pes, ale je to zvíře!</p>
```

Zalomení řádku — **br**

Používá se tehdy, když je potřeba text v paragrafu zalomit na další řádek. Jedná se o nepárový tag (tj. nemá obsah). Nikdy by se neměl používat na odsazení textu či pozicování. Zkratka je odvozena od slova break.

```
<p>Vítejte Honzo<br>ve svém světě.</p>
```

Tlusté písmo — **b** / **strong**

Zvýrazní označený text. Obecně se používá na něco důležitého, co by uživatel na stránce neměl přehlédnout. Zkratka je odvozena od slova bold.

```
<p><b>Zítra odpadá výuka</b> z důvodu společenské akce všech studentů.</p>
```

Kurzíva — **i**

Nastaví na daný text kurzívu, tedy takové nahnutí textu. Často se používá jako odchylka od daného textu, ale může označovat nějakým způsobem část textu. Zkratka je odvozena od slova italics.

```
<p>Vytvoříme dyson sphere, <i>což je sféra či dutá koule, která je postavena okolo hvězdy</i></p>
```

Podtrhnutí — **u**

Podtrhne daný text, zkratka pochází ze slova underline. Používá se jako zvýraznění důležité části textu.

```
<p>Na výlet <u>si nezapomeňte vzít bundu</u>.</p>
```

Horní index — **sup**

Posune text na vyšší index (tj. nahoru), často se používá pro matematické (či jiné) výrazy či tzv. footnotes, neboli poznámky pod čarou. Daný text se posune a zmenší. Zkratka je odvozena od slova superscript.

```
<p>Definujme a:  $x^{8} - 2x^{3} - x^{2}$ </p>
```

Dolní index — **sub**

Velmi podobné hornímu indexu, jen je dole. Používá se stejně a pro stejné účely. Zkratka je odvozena od slova subscript.

```
<p>Přidáme do koncentrátu:  $C_8H_{10}N_4O_2$ </p>
```

Přeškrtnutí — **s** / **del**

Označuje text, který již není platný, nebo je nesprávný. Často se používá pro to, když už něco není platné, ale je možné, že tuto informaci ještě někdo potřebovat. Zkratka je odvozena od slova strike, případně delete.

```
<p>Akce se uskuteční <del>23. 3. 3032</del> 25. 3. 3032.</p>
```

Vodorovná čára — `hr`

Nepárový tag, který vytvoří na stránce horizontální čáru, která slouží k rozdělení stránky na více část. Používá se jako oddělovač delšího textu a obdobně. Zkratka pochází ze slov horizontal rule.

```
<p>Začátek...</p>
```

```
<hr>
```

```
<p>Konec...</p>
```

Obrázek — `img`

Vloží na stránku obrázek z cesty uvedené v atributu src. Do atributu alt se vkládá alternativní text, který se zobrazí když se obrázek nenačte nebo např. v čtečce. Zkratka pochází z názvu image.

```

```

Link — `link`

Vloží do stránky nějaký další soubor, nejčastěji se používá pro CSS. Má dva atributy a to rel, který určuje co vkládáme a href, který popisuje danou cestu (resp. URL). Vkládáme do tagu head.

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

Span

Nemá žádný speciální účel a žádné základní stylování (kromě zobrazování). Používá se na označení celků v textu — k vlastnímu stylování.

```
<p><span>Testující</span> – člověk, co testuje</p>
```

Div

Podobné jako span, místo řádkového zobrazení používá blokové zobrazení. Později se naučíme další blokové prvky, které se hodí na rozložení stránek (určují, co uvnitř je).

```
<div>
```

```
<h2>Článek o věcech</h2>
```

```
<p>Pravdivé, řekl.</p>  
</div>
```

Pre

Podobný paragrafu (určuje textový blok), naopak od něj však nemaže prázdné místo (tabulátory, mezery, nové řádky, ...) a neupravuje text. Text se tedy na stránce vykreslí přesně tak, jak byl definován v kódu. Na více bude zobrazen v monospaced písmu, tedy písmu, kde každý znak zabírá stejně místa.

```
<p>  
  span {  
    color: #7e9dc0;  
  }  
</p>  
<pre>  
  span {  
    color: #7e9dc0;  
  }  
</pre>
```

Výše uvedený příklad vykreslí dva textové bloky, kde ten první bude mít všechny text na jedné řádce, podobně jako: `span { color: #7e9dc0; }`. Druhý bude na více řádkách, přesně tak, jak je v kódu a tedy i s mezery před jednotlivými řádky.

CSS

CSS je zkratka pro **Cascading Style Sheets**, neboli kaskádové styly. Jedná se o technologii, která se používá pro stylování a designování webových stránek. Je přímo závislá na technologii HTML. CSS nemá na webových stránkách v podstatě žádnou alternativu a musí se tedy používat. Používají se různými definicemi, které HTML zobrazení upravují.

Jako stylování si můžeme představit velikost písma, barvy pozadí, textu či obrázky a rámečky. S CSS lze vytvořit spoustu věcí, ale má i své výhody a nevýhody.

Výhody:

- umíme zvolit konkrétní prvky, a to již díky třídám tak i identifikátorům
- jednou definované styly můžeme díky identifikátorům a třídám opakovaně používat
- je to moderní technologie, která se díky komunitě stále upravuje a modernizuje
- vše se upravuje na jednom místě a jednoduše, není potřeba používat další technologie

Nevýhody:

- CSS a jeho vlastnosti (více později) jsou závislé na vykreslovacím jádře. Každá stránka tedy může na všech prohlížečích vypadat jinak.
- když se načte soubor s CSS, daná pravidla, které se nevyužívají, se musí načíst také.

- CSS není zpětně kompatibilní, to znamená, že nová verze (nová pravidla) ve starších verzích (resp. prohlížečích) nebudou fungovat.

Zcela jistě by se našlo ještě více výhod a nevýhod, tyto jsou však nejzákladnější a nejdůležitější.

CSS definice se nazývají pravidla. Pravidla se skládá ze:

- selektoru
- alespoň jedné vlastnosti, která se skládá ze:
 - jména vlastnosti
 - hodnoty vlastnosti

Vlastnosti jsou vlastně nastavení věci, které můžeme použít pro stylování. Za každou vlastnost se musí psát středník `;`. Jako první vlastnost, abychom mohli ukazovat další zápisy, kterou si ukážeme, je vlastnost `color`. Její hodnotou (prozatím) mohou být anglické názvy barev, jako např.: `blue, red, white, black, green, ...`, takový zápis by například tedy vypadal `color: green;`

Zápisy CSS

CSS lze zapisovat několika způsoby. Všechny způsoby jsou založeny na vlastnostech, které jsme si ukázali.

Inline

Nejjednodušší zápis CSS, kde se CSS zapisuje do atributu `style` na jednotlivých tag. Do atributu se zapisují jednotlivé vlastnosti oddělené středníkem. Vypadá to např.:

```
<p style="color: green"></p>
```

Tento způsob nepoužíváme kvůli tomu, že nevyužíváme zcela výhody CSS a ničíme kaskádu, kterou si vysvětlíme později. Tento způsob tedy využíváme jen zřídka a pro velmi (jinak nevyřešitelné) specifické problémy.

External

Nejpoužívanější způsob zápisu CSS. Tento způsob využívá nový soubor, ve kterém píšeme pouze CSS. Používáme celé pravidla. Má příponu `.css` a většinou je ve vlastní složce `css`, či obdobě. Tento soubor poté nalinkujeme do HTML pomocí tagu `link`. Daný odkaz musí splňovat cestu ve složkách. Máme možnost odkazovat na aktuální složku pomocí zápisu `./` a zpětně se ve složkách pohybuje pomocí `../`. Máme možnost i odkazovat na kořenovou URL domény a to pomocí `/`. Tento styl používáme na jakýkoliv typ webu, v našem předmětu zejména na větší projekty, práce či závěrečnou práci.

```
<!DOCTYPE html>
<html lang="cs">
<head>
  <meta charset="UTF-8">
```

```
<title>External</title>
```

```
<link rel="stylesheet" href="style.css"> <!-- aktuální složka, soubor style.css vedle d
```

```
<link rel="stylesheet" href="./css/style.css"> <!-- ve složce css v aktuální složce (ve
```

```
<link rel="stylesheet" href="../style.css"> <!-- v podřazené složce aktuální složky (ve
```

```
<link rel="stylesheet" href="/style.css"> <!-- jsme-li např. na example.com/test/test/t
```

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

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

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

```
</head>
```

```
<body>
```

Embed

V tomto stylu zapisujeme CSS přímo do tagu v HTML. Na této stránce se tedy CSS na HTML ihned aplikuje. Nevýhoda je, že nelze daný styl použít na jiné stránce jinak, než že

CSS soubor byl zbytečný.

```
<html lang="cs">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>External</title>
```

```
<style>
```

```
    selektor {
```

```
        color: green
```

```
    }
```

```
</style>
```

```
</head>
```

```
<body>
```

Import

Tento systém je používán často na externí CSS soubory. Import je typ příkazu v CSS, který přímo v CSS nalinkuje jiný stylovací soubor dovnitř a začne jej používat. Tento systém není potřeba znát, my jej používat budeme zřídka. Lze používat v Embed nebo External zápisech. Tyto zápisy mají mnohé věci pro responzivitu, které jsou pro nás aktuálně nepodstatné.

```
<html lang="cs">
```

```
<head>
```



```
<meta charset="UTF-8">

<title>External</title>


<style>

    @import url("test.css")

</style>

</head>

<body>
```

Selektory

Selektory jsou součástí již zmíněných pravidel. Jak název může vysvětlovat, slouží k tomu, aby se nějaké věci (prvky, tagy, elementy) na stránce vybrali a mohli se stylovat pomocí vlastností.

Selektování tagu

Vybere veškeré tagy na stránce, které jsou daného typu (daného názvu). Na ně poté aplikuje dané vlastnosti. Níže uvedená ukázka vybere všechny tagy `b` na stránce a nastaví jim tmavě oranžovou barvu textu.

```
b {

    color: darkorange;

}
```

Selektování všeho

Vybere všechny tagy na stránce. Vybírá i poletující text a cokoliv jiného.

```
* {

    color: blue;

}
```

Selektování pomocí tříd

Najde všechny tagy na stránce, které obsahují v atributu `class` uvedenou třídu. Daný element může obsahovat i další třídy. Níže uvedená ukázka vybere například prvek `<i class="kocka yoko bambi test"></i>`, protože obsahuje třídu `kocka`. Oproti tagu je před názvem třídy tečka. Třídy tedy nesmí obsahovat tečku, mezery a další speciální znaky.

```
.kocka {

    color: yellow;

}
```

Selektování pomocí identifikátoru

Vybere jeden tag na stránce, který má atribut ID stejného jména. Níže uvedená ukázka vybere prvek `<div id="test"></div>`. Identifikátor se pozná pomocí hashtagu v názvu. Identifikátory tento znak (a ani žádné další speciální) nesmí obsahovat.

```
#test {  
    color: pink;  
}
```

Vnořovací

Slouží k vybrání prvků dle posloupnosti v hierarchickém systému HTML. Poznáte jej díky mezeře mezi tagy, identifikátory, třídami či dalšími selektory. Zápis `div .information` znamená, že se najdou na stránce takové tagy, které splňují že jsou tagu typu `div` a uvnitř v nich (a klidně v dětech dětí, tedy KDEKOLIV uvnitř) mají jakýkoliv tag, který má třídu `.information`. Těchto částí může být nekonečně mnoho.

```
p span {  
    color: green;  
}
```

Seskupovací

Seskupí více selektorů tak, že se na stránce najdou takové prvky, které dané selektory splňují a je na ně použité stejné pravidlo (resp. vlastnosti). Může znovu obsahovat nekonečno těchto částí. Části (selektory) se rozlišují čárkou mezi nimi.

```
span, b, p .test span {  
    color: green;  
}
```

Výše uvedená ukázka obsahuje jeden seskupovací selektor, který se skládá ze selektorů `span`, `b` a `p .test span`.

Násobný

Na stránce se velmi často stává, že potřebujeme na stránce vybrat takové prvky, které mají více tříd najednou, mají daný tag a nebo třeba třídu a identifikátor. K vybrání takového prvku používáme již známé věci — tag dáme na začátek selektoru a potom určujeme jednotlivé třídy tečkami. Skládáme je za sebe bez mezer. Identifikátor můžeme dát před třídy nebo za ně.

```
span#main.green.important {  
    color: green;  
}
```

```
}
```

Výše uvedená ukázka vybere tag `span`, který má identifikátor `main` a má dvě třídy a to `green` a `important`. Ukázkový selektor je ZCELA ROZDÍLNÝ oproti např. selektoru `span #main.green.important` — je důležité si na mezery dávat pozor.

Přímý dědic

Velmi podobný vnořovacímu selektoru, rozdíl je že místo mezer používáme `>` a rozdílné chování stojí na tom, že místo toho aby KDEKOLIV uvnitř našel následující selektor, tak ho bude hledat přímo jako dítě daného tagu a dovnitř se vnořovat již nebude.

```
span > b {  
    color: green;  
}  
  
span#main > .green.important > b {  
    font-weight: bold;  
}
```

Komentáře a jednotky

Komentáře se v CSS zapisují jedním způsobem a to pomocí `/* komentář */`.

V CSS se používá mnoho jednotek, které definují velikost. Ukážeme si znovu pouze základní:

- `px` — pixel na obrazovce
- `cm` — definice centimetru, existuje ještě `mm`
- `%` — udává procentuální velikost z dané dimenze rodiče

Jednotky obecně zařazujeme do kategorií absolutní či relativní. Absolutní jsou ty, které se velikosti zařízení či na ničem jiném nezáleží. Relativní jsou takové, které na něčem jsou závislé. `px` a `cm` jsou absolutní, `%` je relativní.

Základní vlastnosti

Barva textu

Barva textu se nastavuje pomocí vlastnosti `color`. Její hodnotou je barva, v CSS můžeme barvy definovat způsoby:

- **NÁZEV:** dle názvů, které naleznete [ZDE](#).
- **RGB:** `rgb(x, y, z)`, kde jednotlivé znaky jsou čísla od 0 do 255, kde hodnoty `0, 0, 0` je černá barva, `255, 255, 255` je bílá.
- **RGBA:** `rgba(x, y, z, a)`, kde `a` je číslo od 0 do 1, kde 0 znamená zcela průhledná barva a 1 neprůhledná.
- **HEX:** `#abcxyz`, kde jsou po dvou vždy značí hexadecimální číslem pro červenou, zelenou a modrou barvu, stejně jako `a` u `rgba`, určuje se hodnota 0 až 255 ale v hexadecimální zápisu.

- **HEXA:** `#abcxyzde`, kde de značí průhlednost, stejně jakou o rgba.

```
.special {  
    color: red;  
}
```

Velikost textu

Velikost textu se nastavuje pomocí vlastnosti `font-size`. Její jednotkou může být cokoliv zmíněného.

```
.special {  
    color: purple;  
    font-size: 47px;  
}
```

Zarovnání textu

Určuje, jak se daný označený text zarovná vzhledem ke stránce. Vlastnost nazývá `text-align` a její hodnotou mohou být:

- `right` – zarovná text doprava
- `left` – zarovná text doleva
- `center` – zarovná text doprostředka
- `justify` – vyplní text blokově a případně vytvoří mezi slovy mezery

```
.right { text-align: right; }  
.center { text-align: center; }  
.justify { text-align: justify; }
```

Font textu

Nastaví daný font na textu. Vlastnost se jmenuje `font-family`. Daný font musí být nainstalovaný na PC nebo jej musíme nalinkovat z jiného zdroje (ukážeme si později). Jeho hodnota je jméno fontu. Je možné nastavit více jmen a to pomocí čárek. Když uvedeme více fontů, budou se načítat postupně. Pokud jeden z nich nebude nainstalovaný, přeskočí se a zkusí se další. Jako poslední možnost se častou uvádí text fontu, který najde daný typ fontu a pokusí se jej nastavit. Možnosti jsou:

`serif, sans-serif, monospace, cursive, fantasy`.

```
b {  
    font-family: 'Comic Sans', 'Comic Sans MS', sans-serif;  
}
```

Styl fontu

Nastaví jaký typ fontu bude nastaven danému fontu. Vlastnost se jmenuje `font-style`. Existuje styl normální a italic, který vytvoří kurzívu.

```
b {  
  font-family: 'Comic Sans', 'Comic Sans MS', sans-serif;  
  font-style: italic; /* italic / normal */  
}
```

Tloušťka textu

Nastavuje jak tlustý, neboli obteklý daný text bude. Vlastnost se jmenuje `font-weight`. Danou velikost musí font podporovat. Hodnoty jsou po stovkách, začíná od 100 a končí na 900. Kde 100 je nejtenčí typ.

```
.extra-bold {  
  font-weight: 900;  
}
```

Dekorace textu

Jedná se o složenou vlastnost, ve které definujeme typ dekorace, barvy, její styl a šířku. Vlastnost se jmenuje `text-decoration`. Barva může být jakákoliv, velikost jakákoliv s jakoukoliv jednotkou, typ dekorace:

- `line-through` – čára skrz text
- `overline` – čára nad textem
- `underline` – čára pod
- a nebo kombinace uvedeného

Styl může být:

- `solid` – plná čára
- `double` – dvě čáry
- `dotted` – tečkovaná čára
- `dashed` – čárkovaná čára
- `wavy` – vlnitá čára

```
.special {  
  text-decoration: overline green dashed;  
}
```

Vytvoří na všechny prvky s třídou special nad čáru zelené barvy, která je čárkovaná.

Barva pozadí

Nastaví danému prvku barvu pozadí na nastavenou hodnotu. Vlastnost se jmenuje `background-color`.

```
.sale {  
  background-color: red;  
}
```

Velikosti prvků

Jednotlivým blokovým prvkům můžeme určovat výšku a šířku, kterou budou zabírat na stránce. A to pomocí vlastností `width` a `height`, kde se používá číslo a jednotka.

```
.block {  
  width: 150px;  
  height: 150px;  
  background-color: #4fff0a;  
}
```

Výše uvedené pravidlo způsobí, že dané prvky s třídou `block` se vykreslí s výškou a šířkou `150px` a barevným pozadím.

Vnitřní osazení

Určuje jak se prvky uvnitř prvku budou odsazovat od jeho velikosti. Určuje tedy, jaké mezery budou od jednotlivých prvků až k okraji. Neovlivňuje ostatní prvky. Jedná se o složenou vlastnost `padding`.

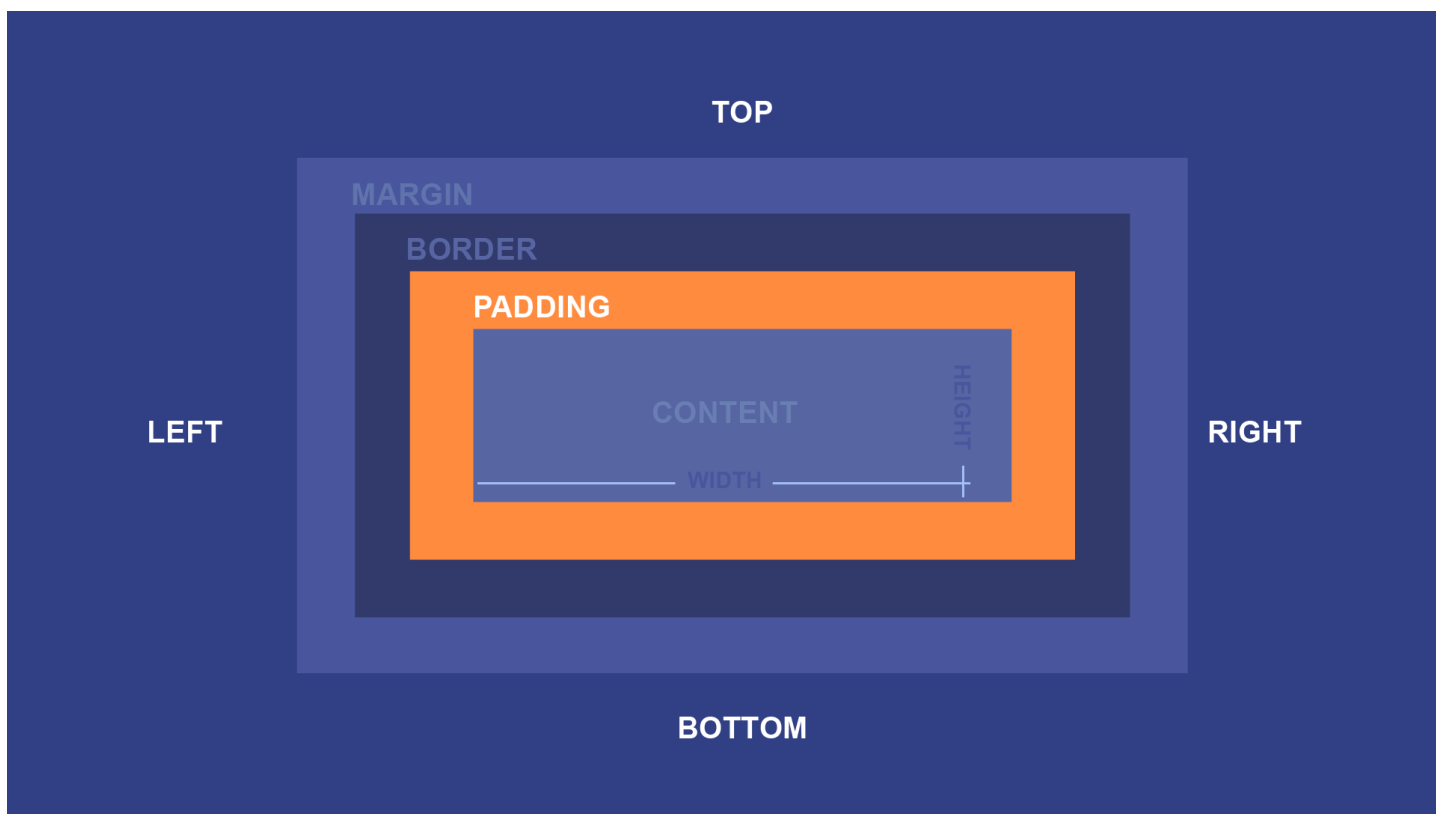
```
.block {  
  padding: 40px;  
  /*  
  padding-top: 40px;  
  padding-bottom: 40px;  
  padding-left: 40px;  
  padding-right: 40px;  
  */  
}
```

Vnější odsazení

Určuje, jak se prvek (klidně i řádkový), bude od ostatních prvků umístěn. Jedná se o složenou vlastnost `margin`. Nepočítá se do velikost samotného prvku a ovlivňuje jen okolní

prvky. Okolní prvky mohou mít nastavený vlastní odsazení, které budou sami respektovat – dané odsazení se však nikdy nespojí (nepočítá se dohromady).

```
.block {  
  margin: 40px;  
  /*  
  margin-top: 40px;  
  margin-bottom: 40px;  
  margin-left: 40px;  
  margin-right: 40px;  
  */  
}
```



A další věci

Tato lekce obsahuje jen nutné minimum, na které budeme celý rok referovat. Pokud se na to cítíte a myslíte si, že je potřeba si dále o HTML a CSS zjišťovat informace, doporučuji se podívat na následující témata:

- **formuláře** (jak v HTML, tak i jejich stylování)
- další tagy (listy, tabulky, moderní HTML tagy, nadpisy, ...)
- další vlastnosti (background-image, border, flex, grid, animace)
- responzivita