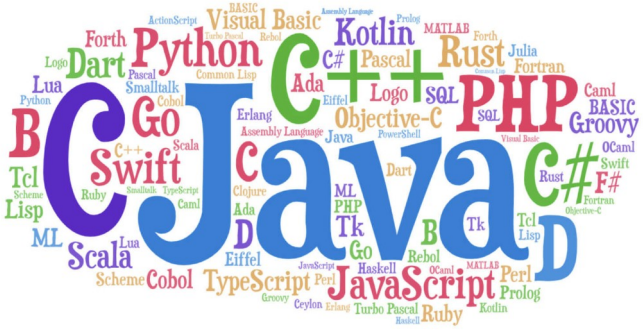
Malá kniha programovania

Autor	Vydavateľ	Licencia	Vydanie	Autor obálky
Stanislav Hoferek	Greenie knižnica	CC-BY-NC-ND	Prvé (2021)	Stanislav Hoferek

O knihe

Poďme variť špagety. Áno, nie je to typická veta, ktorú by ste očakávali pri knihe o programovaní. Táto krátka kniha za vás nič nenaprogramuje a nevyrieši všetky problémy sveta. Dokonca to nie je ani kuchárka. Naučí vás však myslieť ako programátor či ako niekto, kto pracuje s akýmkoľvek kódom. Bez ohľadu na množstvo zátvoriek a úvodzoviek. Poslanie je tak mimoriadne jednoduché – vysvetliť základ, informovať o tom, ako sa dá narábať s informáciami a pritom použiť veľa prirovnaní. Varovanie: Ak neviete urobiť špagety, toto nie je kniha pre vás!



MALÁ KNIHA PROGRAMOVANIA

STANISLAV HOFEREK

Na čo je dobré mať myslenie ako programátor? Na veľa vecí. Okrem iného na to, aby ste mali lepší plat, aby ste sa menej narobili a aby ste mohli zarábať aj vtedy, keď oddychujete. Táto kniha hovorí o základoch myslenia a je na každom, koľko z toho si zoberiete. Štýl písania knihy je viac veselý, ako vážny. Malá kniha programovania je tiež vhodná príručka pre školy a všetkých samoukov, ktorí uvítajú takúto pomôcku v slovenčine.

Obsah

Kapitola I: Špagety. Čo sú to špagety?

Televízna súťaž milionár. Posledná otázka. Ide sa domov s naozaj veľkým balíkom, alebo sa spadne na garantovanú hranicu a môže sa prerobiť kúpelňa? Každý povie, že vie, čo sú to špagety. Ale schválne, kto z vás niekedy niekomu vysvetľoval, čo sú to špagety? Máte niekedy diskusiu o špagetách? A kto môže vedieť o špagetách viac, ako vy?

Nasleduje logická otázka – prečo práve špagety? Prečo nie niečo iné? V ktorom programovacom jazyku sa spomínajú špagety? A prečo práve toto jedlo v oblasti, ktorá s jedlom veľmi nesúvisí?

Starý vtip hovorí o tom, že programátor je stroj, ktorý premieňa pizzu na strojový kód. Vie to pobaviť, no hlavne je to veľmi presné. Niekto, kto dokáže vyrábať kód, vyrába kód. Je to podobné, ako keď niekto, kto vie učiť nemčinu, učí nemčinu. Jasné, že je to pre mozog náročné a treba si dopĺňať energiu. Výsledkom je však práca, ktorá má zmysel vtedy, keď sa robí dobre.

V spomínanom Milionárovi by mohli byť rôzne odpovede na otázku, čo sú to špagety. Klasické ABCD. Ktoré bude správne? Podľa pravidiel súťaže môže byť správna len jedna odpoveď, zatiaľ čo v reálnom živote to tak nemusí byť. A inak popíše špagety niekto z talianskeho mesta Bologna, než niekto z Londýna, Bratislavy či z Wellingtonu na Novom Zélande. Aj preto, že klasické špagety môžu všade vyzerať inak. Tak, ako môže inak vyzerať pivo, palacinka či napríklad bežný pracovný deň.

Keď poviem priateľke, že chcem špagety, tak viem, že mi ich zariadi. Svojou ženskou logikou by prišla k záveru, že chcem varené jedlo s omáčkou a so syrom. Ja, ako muž, by som mohol pochopiť, že chce, aby som v obchode okrem iných vecí zobral aj dve alebo tri balenia špagiet. Len tak, do zásoby. A keď sa ma spýta, prečo som doniesol tri balenia, tak automaticky a bez premýšľania poviem, že sa to nepokazí.

Programovanie a myslenie programátora je ako špagety. Môže byť rôzne, môže sa odlišovať. Základ však býva rovnaký a cieľ tiež. Urobiť niečo, čo má zmysel. A ak sa všetko podarí, tak bude dobre. Programátor, podobne ako skúsený kuchár, chce dobrý kód alebo recept. To dôležité, v dobrom poradí a bez zbytočného balastu.

Kapitola II: Vodu máme? A soľ?

Ľudia, ktorí nikdy nestavali dom, často povedia, že dom sa stavia od základov. Je to pravda, je to jednoduché… a keby niekto začal reálne stavať od strechy, bola by to automaticky katastrofa. Alebo nie?

Janka pri varení špagiet najskôr otvorí fľašu s omáčkou a vyleje túto zmes na panvicu. Až potom dá hriať vodu, do ktorej v správnom okamihu vloží obľúbenú podlhovastú cestovinu. Robí to tak ona, jej mama, babka, teta Zuzka, ten fúzatý pán z Hurbanova, farárova vnučka a aj ten podivín, čo vie vymenovať všetkých apoštolov v desiatich jazykoch. Len Lukáš, ktorý dostal štvorku zo slovenského jazyka a zhodou okolností bol najlepší na Slovensku v matematickej súťaži, dá najskôr hriať vodu. Je to rebel! A zároveň niekto, kto vie, že na prípravu vody a uvarenie cestovín potrebuje viac času, než na prípravu omáčky. Nemá túto múdrosť od rodičov, ale prišiel na to sám, jednoduchým pozorovaním. Možno vie, že voda má vysokú tepelnú kapacitu, alebo si na svojom telefóne zapol stopky… alebo niečo celkom iné – ale vie to. A čuduje sa, že to tak nerobia ostatní.

Programovanie, stavba domu, varenie špagiet – na všetko je nejaký postup. Niekedy sa to dá skrátiť, inokedy nie. A je na nás, či sa chceme zaujímať o to, ako to niekto robí lepšie. My, na Slovensku, rátame s tým, že Taliani vedia urobiť lepšie špagety, Nemci vedia po nemecky lepšie ako my a tiež to, že priemerný Brazílčan vie lepšie tancovať a hrať futbal ako priemerný Slovák. Je to vždy pravda? Nemusí byť. Ale logicky by to tak malo byť. Podobne, ako by malo byť logické, že dom sa stavia od základov.

Potom príde John z Ameriky a ukáže, že ak sa dá vyrobiť zároveň každá stena, všetky potrebné diely na strechu a pár ďalších zlepšovákov, tak môže byť jednoduchý dom postavený za pár hodín. Toto sem, toto tam a hotovo. Možno to nebude mať takú trvácnosť ako stredoveký hrad, ale pokojne mohol byť prvý hotový diel strecha. Zdvihne sa, položí sa kde sa má a pevne sa to spojí. Alebo sa rovno donesie hotový dom na obrovskom aute aj s kľúčom a jediné, čo je treba urobiť, je pripojiť na vodu a ďalšie inžinierske siete.

Programovanie je ako recept na špagety. Ak niekto chce urobiť napríklad veľmi pekný eshop, tak vie, že potrebuje pripraviť grafiku, texty, musí to doladiť na rôzne zariadenia, vychytať bugy a musí tomu dať jasnú štruktúru. Tak, ako dobrý kuchár potrebuje na naozaj dobré špagety vodu, soľ, spomínanú cestovinu, syr… a nejakú veľmi dobrú omáčku, kde pôjde to, čo tam má ísť. Tak budú všetci spokojní. A ak chce niekto viac syra, alebo napríklad inak urobené menu v eshope, tak sa to spraví. Však prečo nie?

Kapitola III: Priblíženie

Mama učí svoju dcéru, ako správne urobiť špagety. Varuje ju, čo sa môže stať, keď sa budú variť príliš dlho alebo príliš krátko. Hovorí o omáčke, o syre… a dcéra sa tvári, že počúva. Oveľa radšej by išla von. Všetky jej kamošky išli na ten super film, len ona musí ostať doma a učiť sa robiť hlúpe špagety. Stále špagety. Nemá ich rada a po tom, čo sa stalo, ich má ešte menej rada.

Na ďalší deň ide s otcom na miestny hrad, kde sú ukážky ľudových remesiel a rôzne zaujímavosti zo života v stredoveku. Jej otec šermuje s veľkým, obojručným mečom a núti svoju dcérku, aby to skúšala aj ona. Čo má robiť? Skúsi to, nech má otec radosť, ale oveľa radšej by hladkala toho krásneho koníka, ktorého hladkajú ostatné deti.

Prvá vec, ktorá je dôležitá, je priblíženie. Je o tom vlastne aj táto kniha. Kto chce ísť do kina na ten úžasný nový film, tak asi nechce tlačiť do hlavy recept na špagety. Koho zaujme niečo, k čomu sa nemôže dostať, ten sa nebude sústrediť na niečo iné. Človek, ktorý sa chce naučiť napríklad programovať, potrebuje spojiť učenie s niečím, čo je mu blízke, čo ho baví. Dvaja chalani, jeden šachista a jeden futbalista, sa majú možnosť naučiť niečo o programovaní. Zhodou okolností ich otec má dokončiť dva eshopy. S kým bude pracovať na eshope s futbalovými dresmi a loptami a s kým na obchode, kde sú šachovnice a knihy o šachových veľmajstroch?

Pri troche snahy sa dá všetko so všetkým spojiť. Urobiť dobrú webovú stránku môže byť ako uvarenie dobrých špagiet, alebo ako víťazstvo nad silnejším súperom v akomkoľvek športe. Treba na to prípravu, zaistiť všetko potrebné a hlavne odhodlanie a trpezlivosť.

Kapitola IV: Nevedeli sme, ako chutí prst na nohe

Keď sa narodil Herkules, tak bol okamžite silnejší ako dospelý muž. Táto a mnohé ďalšie legendy je spojená s niekým naozaj výnimočným. Realita je však taká, že legendy ostávajú legendami a na začiatku nášho života nevieme nič. Nevieme, koľko je 1+1. Nevieme, že existuje nejaký deň a nejaká noc. Nevieme poslednú číslicu čísla pí, ale to je normálne – lebo nevieme, čo je to číslica čo je to matematika. Na základe pozorovania a od iných ľudí sa dozvedáme, že niekedy prší, inokedy sneží a že v telke je neskutočne veľa reklamy. Nevieme, ako chutí prst na našej nohe a tak to musíme preskúmať.

Inštruktor lyžovania kedysi vôbec nevedel lyžovať. Majster sveta v hode kladivom niekedy nevedel zdvihnúť lyžicu. Človek, ktorý kráčal po Mesiaci, niekedy nevedel kráčať po vlastných tu na Zemi. Ale… zvládli sme to. Postupne. Nie všetko naraz. Od základov. Od jednoduchých vecí, kde sa nedá veľmi čo pokaziť, po niečo, čo je náročné, citlivé na najrôznejšie omyly a niekedy aj nebezpečné, ale výsledok stojí za to.

Keď sa niekto učí programovať, veľmi často začína s **Hello World**. Jednoduchým kódom, ktorý ukáže, či daná technológia funguje správne. Ako tvrdí expert na rôzne programovacie jazyky a zároveň jablká, svet nezaujíma, že ho zdravíš. Základ je však dobrý. Mnohé programovacie jazyky a vlastne čokoľvek, čo má svoju postupnosť, sa učia od toho najjednoduchšieho. Pamätáte si, keď ste prvýkrát otvorili Word alebo niečo na písanie a dali ste do toho prvé písmená? Pravdepodobne nie. Videli ste papier, hore ikonky a na ten papier sa dalo písať klávesnicou. Uloží sa, vytlačí sa, pošle sa mailom a hotovo. Netreba na to žiadne školenie. Ak by ste ale chceli, aby napríklad v Exceli každá bunka svietila na zeleno, žlto či červeno podľa toho, aké tam je číslo, tak by ste sa to museli naučiť. A možno by ste to používali často, pretože je to tak jednoduché a hlavne prehľadné. V tomto prípade ide napríklad o podmienené formátovanie, ktoré sa dá výborne použiť.

Programovanie je zaujímavé už v tom, že jednoduché veci môžu mať ohromný zmysel. Dobrým príkladom sú chyby. Stačí si zobrať napríklad recept. Píše sa v ňom, že do cesta treba 2 vajíčka, ale každý sa už naučil, že na tento recept je oveľa lepšie dať 4 vajíčka. Nebude potrebný človek, ktorý by tomu ďalšiemu hovoril, že okrem všetkých inštrukcii treba ešte zmeniť počet vajec. Jednoducho sa pri prepisovaní receptu z papiera do elektronickej podoby nenapíše 2 vajíčka, ale 4 vajíčka. Jednoduché, rýchle, prehľadné. A bez hlúpych výnimiek.

Keď som začal pracovať na úrade, stretol som sa s tým, že každý človek dostal 3 papiere, ktoré boli zošívačkou spojené. Papiere boli asi 100x okopírované a už bolo len ťažké vidieť všetky čiary a písmená. Kto niečo potreboval, musel vypísať časť papieru 1, väčšinu papieru 2 a zase časť papieru 3. Následne sa podpísať tam, kde nebolo miesto na podpis. Takto to fungovalo a nebolo to potrebné meniť, však by to fungovalo ešte roky. Alebo je zmena dobrý nápad?

Zmenil som text tak, aby bolo miesto na potrebné veci. Všetko nepotrebné sa vyhádzalo, opravili sa chyby v texte a bol z toho dokument na jednu stranu, kde nemusel nikto nič vysvetľovať. Nebola potrebná zošívačka a všetko sa urýchlilo, pričom sa starší ľudia prestali sťažovať na drobné a veľmi svetlé písmená na bielych papieroch. Na zmenu k lepšiemu prešli všetky kancelárie za jednu minútu, pričom príprava textu trvala 10 minút.

Kapitola V: Kde sa to mám všetko naučiť?

Keď sa s niekým rozprávam o programovaní, tak často počujem od iného človeka, že to nie je pre neho. Nemá na to školu. Prvé, čo mi napadne, že keď Mozart vo veľmi mladom veku hral na klavíri, tak na to asi tiež nemal štyri vysoké školy. Ale hral. Šachisti nechodia na šachovú školu, ale aj tak hrajú šach. Automechanik, ktorý nemá školu a všetko ho naučil otec a starší brat, dokáže opravovať autá. Škola nás nenaučila jazdiť na bicykli ani jesť pizzu.

Škola nás naučila, ako počítať kompóty u babičky. Má 7 políc a v každej je 7 kompótov. Koľko je ich spolu? Škola však nenaučí, že ak sa dajú všetky kompóty na tú najvyššiu policu, tak je väčšia šanca, že to celé spadne. Škola na informatike ukáže, čo robí počítačový procesor, ale neporadí, ako si kúpiť nový počítač, na ktorý nebude treba nadávať už od prvého dňa. Najlepším programátorom na škole môže byť pokojne dievča, ktoré nemá v láske počítače, ale vie si najlepšie zapamätať konkrétny príklad z knihy a zapísať ho na tabuľu bez jedinej chybičky. Bude to však správny človek, ktorý bude v budúcnosti využívať presne tento program? A bude vedieť nájsť chybu v zložitom programe, ktorý po malých zmenách prestal fungovať?

S programovaním je to ako s písaním a čítaním. Jasné, niečo sa vieme naučiť sami. Na niečo iné sa zas hodí, keď s tým niekto pomôže. Na internetových diskusiách je časté, že niekto pridá svoj kód a posťažuje sa, že to nerobí presne to, čo chce. Pozrie sa na to niekto skúsenejší a rýchlo zistí, že program je napísaný dobre, ale odkazuje na súbor, ktorý nefunguje správne. Takáto pomoc je pre niekoho spásou a víťazstvom v ťažkom boji, zatiaľ čo profík uvidí chybu pomaly z kilometra a navrhne najjednoduchšie z viacerých riešení.

Tak, ako pri varení špagiet alebo čomkoľvek inom je vhodné vzdelávať sa. Teoreticky i prakticky, sám i s inými. Nie je žiadne tajomstvo, že na Profesii a podobných portáloch je veľký záujem o ľudí, ktorí ovládajú HTML + CSS + PHP + JavaScript + Python + SQL + Adobe PS + 300 ďalších vecí. Logicky by tiež jazyková škola chcela zamestnať schopného človeka, ktorý ovláda perfektne angličtinu, nemčinu, francúzštinu a ruštinu a ešte bude svojim šarmom priťahovať žiakov od konkurencie. Kde sa to však všetko naučiť?

V roku 1993 mal internet málokto, a tak bolo aj veľmi málo expertov a tiež rôznych manuálov. Bol problém dostať sa k internetu, nie to ešte naučiť sa všetko, čo bude mať nejaký reálny zmysel až v budúcnosti. V roku 2003 sa už internet bežne používal u ľudí, ktorí mali na to veľmi dobrý dôvod. V roku 2013 mal internet ktokoľvek a už to dávno nie je miesto len pre počítačových odborníkov. Napriek tomu, že je oveľa viac návodov, lektorov, kurzov a všetkého možného je dlhodobý nedostatok programátorov, správcov siete a tak podobne. Každý rok je ich potrebné viac a viac. Čím skôr sa tak začne, tým lepšie.

HTML nie je programovanie v pravom zmysle slova. Dá sa však považovať za úplný základ myslenia. Vytvorenie jednoduchej stránky je jednoduché, dá sa ľahko naplniť informáciami, ktoré dokáže zobraziť každý človek na každom zariadení. Pre mnohých je to dobrý začiatok. Hodí sa taktiež pre ľudí, ktorí chcú zjednodušiť nejakú obyčajnú činnosť. Napríklad vďaka rámcom, čo je dnes už veľmi málo používaná technológia, sa dá stránka zložiť z viac stránok. Nemusíte tak sledovať štyri stránky s výsledkami, úplne postačí jediná.

Kapitola VI: Potrebujem odborníka?

V súčasnosti je množstvo možností, ako sa niečo naučiť. Ak chcete dobre strieľať z luku, vytvárať si vlastné keramické hrnčeky či recitovať Vergília v latinčine, tak môžete. Asi to bude chcieť otvoriť peňaženku, zaobstarať si všetko potrebné a potom to už pôjde.

Ako je to s programovaním? Naučiť sa programovať je možné a často aj jednoduché. V tomto prípade často stačí zobrať niečo z toho, čo je voľne na internete práve pre ľudí ako ste vy. Najrôznejšie návody, hotové kúsky textov či vysvetlené jednotlivé možnosti rôznych jazykov. Dá sa povedať, že pre človeka, ktorý vie všetko, je na internete všetko. Až na jednu maličkosť.

Je úplne bežné, že nájdete kód, ktorý napríklad skrýva nepotrebnú časť textu a ukáže ju len vtedy, keď to bude potrebné. Zároveň nájdete kód, vďaka ktorému sú niektoré najdôležitejšie časti ešte viac zvýraznené. Čo ak sa tieto dva kódy dostanú do nejakého konfliktu a výsledok bude vyzerať veľmi zle? Napríklad tak, že vôbec nič nebude fungovať? Žiadne návody nemajú všetko, čo by človeku mohlo napadnúť. Je to zvlášť pri vytváraní úplne nového obsahu, ktorý musí fungovať čo najlepšie aj pre tých, ktorí používajú netypický či mimoriadne zastaralí softvér. Tu sa veľmi hodí práca odborníka.

Skutoční odborníci, ktorí pomôžu s kódom, nie sú úplne lacní. Za jedinú hodinu si vedia vypýtať celkom dosť. Na druhej strane, ak sa počas tej jednej hodiny naučíte oveľa viac, ako zo štyroch hodín štúdovania návodov a vyrieši problém, ktorý vás trápi už tretí deň, tak sa takáto pomoc skutočne oplatí. Experti na rôzne jazyky sa dokážu s vami dorozumievať napríklad cez softvér na zdieľanie plochy aj s audiom a vy priamo môžete písať kód presne tam, kde to navrhne lektor. Výhodou takýchto školení je okrem iného i to, že okrem spôsobu písania sa naučíte aj niečo dôležitejšie – spôsob myslenia. Podarí sa tak nielen urobiť dobrý kód, ale zároveň sa dá ľahko naučiť dobrý zlepšovák alebo sa dá vyhnúť zbytočným chybám v budúcnosti na oveľa dôležitejších projektoch. Mne napríklad dobre pomohol Petr „Pepa“ Pavel, ktorý školí tak firmy, ako i úplných začiatočníkov s programovaním. Okrem iného robí mentora v zásielkovni.

Existujú dva obrovské rozdiely medzi profesionálmi a laikmi. Jeden z nich súvisí s odbornými, presnými názvami. Postup, ako postaviť dom, urobiť akúkoľvek prácu či správne organizovať čas, je algoritmus. Logická postupnosť. Taktiež sa rozlišuje frontend a backend. Frontend je to, čo vidí akýkoľvek používateľ, ktorý si spustí program alebo otvorí webovú stránku. Backend je to, čo je vo vnútri a spôsobuje, že to celé funguje.

Príklad backendu: Vypíš mi štáty južnej Ameriky. Využi túto databázu. Štáty zoraď podľa abecedy od A po Z. Oddeľ štáty čiarkami.

Príklad frontendu: Argentína, Bolívia, Brazília, Chile, Ekvádor, Falklandy, Francúzska Guayana, Guyana, Kolumbia, Paraguaj, Peru, Surinam, Uruguaj, Venezuela.

Druhý rozdiel je v tom, že skutočný profesionál sa neustále vzdeláva. Učí sa nové technológie, skúša nové možnosti a berie do úvahy aj bezpečnostné riziká či optimalizáciu na rôzne zariadenia.

Kapitola VII: Idem na to. Čo potrebujem?

Pripravení na písanie kódu? Výborne. Táto veľmi stručná kapitola skúsi povedať, čo na to treba, prípadne mimoriadne pomôže.

- **Chuť**. Chuť pustiť sa do niečoho nového, nepoznaného. Dostať sa ďalej, ako ste sa dostali doteraz. Nebude to výprava na štýl Star Trek, ale stále je to do neznáma, ktoré je pod tým, čo všetko poznáme. Ak vás zaujíma, ako v skutočnosti fungujú veci a čo sa dá urobiť, aby niečo fungovalo dobre, máme prvú ingredienciu. Išlo by to aj bez chuti, ale bola by to otrava. A tú nikto nemá rád.
- **Angličtina**. Angličtina a technika ide dohromady. Množstvo termínov je v angličtine a používajú ich aj Slováci a Česi. Namiesto komplikovaného slovenského termínu sa často použije anglický, ktorý je univerzálny a často veľmi výstižný. Iste, dá sa naučiť pracovať ako programátor bez znalosti angličtiny, ale základné veci, ako napríklad kladenie rozumných otázok, sa veľmi hodia. Na internet píšu často aj o technických veciach ľudia, ktorí používajú angličtinu ako druhý jazyk. Nepotrebujete tak žiadne štátnice z angličtiny na úžasnej úrovni, ale ak dokážete prejsť 2-3 hry v angličtine, dokážete tiež vytvoriť jednoduchý kód.
- **Pochopenie vstupu a výstupu**. Ak viete, ako funguje počítač z hodín informatiky, ste na dobrej ceste. Pri programovaní sa často používa metóda „if – then – else“, takže „ak – potom – inak“. Ak pôjdem do mesta, urobím nákup. Inak budem ležať na gauči a útočiť na slané tyčinky.
- **Znalosť základných klávesových skratiek**. CTRL+C a CTRL+V vie používať aj Andrej Danko. Základné skratky na kopírovanie a vkladanie, rýchle ukladanie (CTRL+S), prepínanie okien (ALT+TAB), obnovenie stránky (F5) a niekoľko ďalších je požiadavkou. Človek, ktorý sa nenaučí niečo, čo použije každý deň veľmi často, bude mať obrovský problém pri niečom zložitejšom. Väčšina skratiek je rovnaká v rôznych programoch a tak sa mnohí nemusia vôbec nič učiť.
- **Textový editor/IDE**. Aj jednoduchý poznámkový blok sa dá použiť na napísanie dobrého webu alebo programu. Postupne však každý prechádza zo základného poznámkového bloku na pokročilejšie textové editory alebo tzv. IDE. Je jedno, v čom sa píše kód, ale dobrý nástroj vám môže pomôcť napríklad s doplňovaním potrebných znakov či farebným odlišovaním. Inak povedané, s dobrým programom, s ktorým sa naučíte pracovať, ste pri písaní produktívnejší. Veľmi populárny je napríklad Notepad++ alebo Atom. Medzi známe IDE (kompletný balík potrebných nástrojov) patrí NetBeans, Eclipse a podobne. Pri učení môžu veľmi dobre pomôcť aj webové nástroje, napríklad codepen.io/pen.
- **Webový prehliadač**. Žiadna novinka. Ak sa píše napríklad webová stránka, potrebujete moderný prehliadač, cez ktorý sa dajú vychytávať chyby. Taktiež potrebujete niečo, čo používa množstvo ľudí, aby ste vedeli, ako to presne vyzerá. Pri písaní webovej stránke je časté ukladanie aktuálnych zmien v textovom editore a obnovovanie stránky vo webovom prehliadači. Ak máte väčší monitor, môžete dať tieto dve okná vedľa seba, to veľmi zrýchli prácu. Najčastejšie riešenia sú Google Chrome a Mozilla Firefox.
- **Ultra mega výkonný superpočítač**. Potrebujete ho, ak chcete popri písaní kódu hrať najnovšie hry, strihať video a sledovať do toho film v Ultra HD rozlíšení. V prípade, že chcete hlavne pracovať s kódom, úplne postačí akákoľvek šunka. Ideálne však niečo, kde sa nebude 2 minúty spúšťať internetový prehliadač. Vytváranie kódu nie je vôbec náročné na hardvér a poslúžiť môže čokoľvek.

Kapitola VIII: Kde som to mal zabočiť?

Ako už teraz určite viete, existuje viac rôznych programovacích jazykov. Každý má svoje výhody a niektoré sa dajú výborne kombinovať. Jednotlivé programovacie jazyky sú dobre popísané na rôznych miestach, napríklad na STUBA:

http://www2.fiit.stuba.sk/~bielik/courses/flp-slov/jazyky/programovacie_jazyky.html

Ak by vás špeciálne zaujímala história, tak tu je klasický nápis Hello World hneď v 50 rôznych jazykoch:

https://medium.com/javarevisited/70-years-of-hello-world-with-50-programming-languages-2400de893a97

- C++, Ruby, Python, Perl, JAVA a mnohé ďalšie jazyky sú veľmi zjednodušene vhodné na klasické aplikácie. Ak chcete vytvoriť napríklad novú hru, môžete vyskúšať napríklad niektorý z nich. Podobných jazykov je neuveriteľné množstvo, vrátane rôznych veľmi netypických, urobených z čistej recesie. Často je dobré začať s takým jazykom, ktorý používajú ostatní vo vašom okolí a vedia dobre poradiť. S programovacím jazykom sa väčšinou pracuje tak, že vytvárate kód, ktorý je dobre pochopiteľný z pohľadu človeka. Následne tento kód špeciálny softvér skompiluje – upraví tak, aby ho dokázal správne spúšťať počítač.
- Webové jazyky. Ak chcete urobiť dobrú webovú stránku, alebo meniť akýkoľvek obsah na internete, veľmi sa vám hodia znalosti HTML. HTML sa dá výborne kombinovať s CSS, PHP či JavaScriptom, čo znamená, že najrôznejšie funkcie sa dajú vytvoriť niekoľkými rôznymi možnosťami. Výhodou HTML, CSS a JavaScriptu je tiež to, že dokáže vytvoriť peknú prezentáciu alebo iný druh obsahu bez akéhokoľvek špeciálneho softvérového vybavenia. Uložíte, pustíte napríklad v Google Chrome a hotovo. Webové jazyky sa dajú výborne učiť napríklad cez web w3schools.com, kde je tiež množstvo dobrých príkladov.
    - HTML použijeme, keď chceme pridať tabuľku, text, obrázok alebo iný obsah.
    - CSS použijeme, ak chceme, aby daný obsah vyzeral pekne a uhladene.
    - JavaScript použijeme, keď chceme pridať niečo, čo sa generuje. Napríklad skript, ktorý náhodne vyberie jeden z obrázkov alebo urobí jednoduchý výpočet či celú kalkulačku.
    - PHP je vhodné na prácu s pridávaním obsahu z databázy, vytvorenie prihlásenia a tak podobne. Spolu s JavaScriptom generujú obsah, zatiaľ čo HTML a CSS ho hlavne zobrazuje konečnému používateľovi.
- Špeciálne programovacie jazyky, určené napríklad na netypické zariadenia. Sem sa dá zaradiť napríklad Assembler, vďaka ktorému funguje všetko neuveriteľne rýchlo, ale jeho používanie rozhodne nie je nič pre začiatočníkov.

Povedzme, že chceme vedieť robiť základné webové stránky. Stačí prejsť návodom, napríklad na vyššie uvedenom w3schools, kde je popísané, ako sa pridávajú obrázky, odkazy na iné stránky a tak podobne. Zároveň otvárať v ich vstavanom editore konkrétne kódy a meniť ich. Len tak, vidieť čo to urobí. A zároveň vytvárať svoju stránku, či už od základov, alebo z ktorejkoľvek voľnej webovej šablóny. Google ich pozná celkom dosť. Ak sa vyhľadá napríklad free responsive html website, tak sa trafíte do čierneho. Stačí prepísať texty, urobiť pár zmien a postupne vyhádzať všetko, čo nepotrebujete. Následne nahrať na web.

Kapitola IX: Mám dvoch drakov a žiadnu princeznú

V roku 1997 počítačový softvér porazil Garryho Kasparova v šachu. Mimoriadne výkonný počítač, dobre napísaný softvér a bohatá databáza šachových partii na jednej strane. Na druhej strane bol vtedajší majster sveta v šachu.

Dnes sú počítače podstatne výkonnejšie a softvér dokáže viac, no stále sú najlepší ľudskí hráči veľmi podobní svojou výkonnosťou s počítačmi. Prečo je to tak? Človek sa nezmenil, no počítače sú omnoho výkonnejšie. Dôvod je veľmi jednoduchý – šach samotný je komplikovaný. A to, čo by počítač vždy označil za chybu, sa môže ukázať ako výborný ťah po niekoľkých ťahoch. Je to komplikovaná veda a len tí najlepší šachisti dokážu poraziť počítačové programy na vyššej náročnosti. Človek má desať prstov. Naučili sme sa používať desiatkovú sústavu. Počítač reaguje na jednoduché jednotky a nuly. Napríklad ide prúd a nejde prúd. Z toho sa dá urobiť akýkoľvek program, vrátane akéhokoľvek obsahu. Ideálnou kombináciou je kombinovať silu počítača a kreativitu človeka. Aj o tom je programovanie. Využívať algoritmy tak, aby všetko náročné robil počítač, zatiaľ čo človek môže robiť niečo zaujímavejšie ako komplikované výpočty.

Dobrým príkladom je tzv. Debuggovanie, alebo jednoducho povedané vychytávanie múch a rôznych nedostatkov. Dobre to ukazuje záchrana princeznej, ktorá je napríklad aj v tomto známom vtipe: https://devhumor.com/content/uploads/images/October2016/toggl-how-to-save-the-princess-in-8-programming-languages.jpg

Výhodou programovania je to, že sa dá vyskúšať čokoľvek. Ak kód nefunguje, môže sa na to ísť iným spôsobom. Pridať pár riadkov a vyskúšať, pričom pridané riadky sa dajú kedykoľvek znovu odstrániť. Jeden človek tak môže v priebehu minúty vyskúšať množstvo rôznych riešení. Povedzme, že chceme zachrániť princeznú. Niekedy máme dvoch rytierov, ale ani jedného koňa. V horšom prípade máme dvoch drakov, ale ani jednu princeznú. Alebo namiesto zachránenia princeznej program vyhodnotí, že treba zachrániť draka pred princeznou.

Problém môže vzniknúť napríklad pri farebnej schéme webovej stránky. Autor chce zmeniť farbu odkazov z modrej na zelenú. Prepíše hodnoty, ale odkazy sú stále modré. Prečo? Neuložil sa dokument so stránkou? Nebola obnovená stránka v prehliadači? Iná časť kódu je „silnejšia“ a má prednosť, čím spôsobuje, že odkazy sú naďalej modré? Chýba bodka, zátvorka či úvodzovka? Alebo sú odkazy odteraz červené? Aj to sa môže stať. Vo väčšine prípadov sa však dá vrátiť k tomu, čo fungovalo predtým a postupne hľadať chybu. Dá sa presne vystopovať, kedy to prestalo fungovať.

Reálny problém, ktorý sa stal: Stránka nefunguje správne, niektoré jej časti sa nezobrazia vôbec. Problém bol po dlhom uvažovaní a skúšaní v tom, že sa JavaScript spustil skôr, ako sa mal. Prebehol v čase, keď ešte neboli zadefinované všetky časti, ktoré mal spracovať. Je to niečo podobné, ako keď dáte ráno murárovi úlohu, aby vymuroval stenu. Vybije sa vám mobil a až večer zistíte, že všetky tehly na stavbe sú ešte stále v inom okrese.

Kapitola X: Kostra v skrini

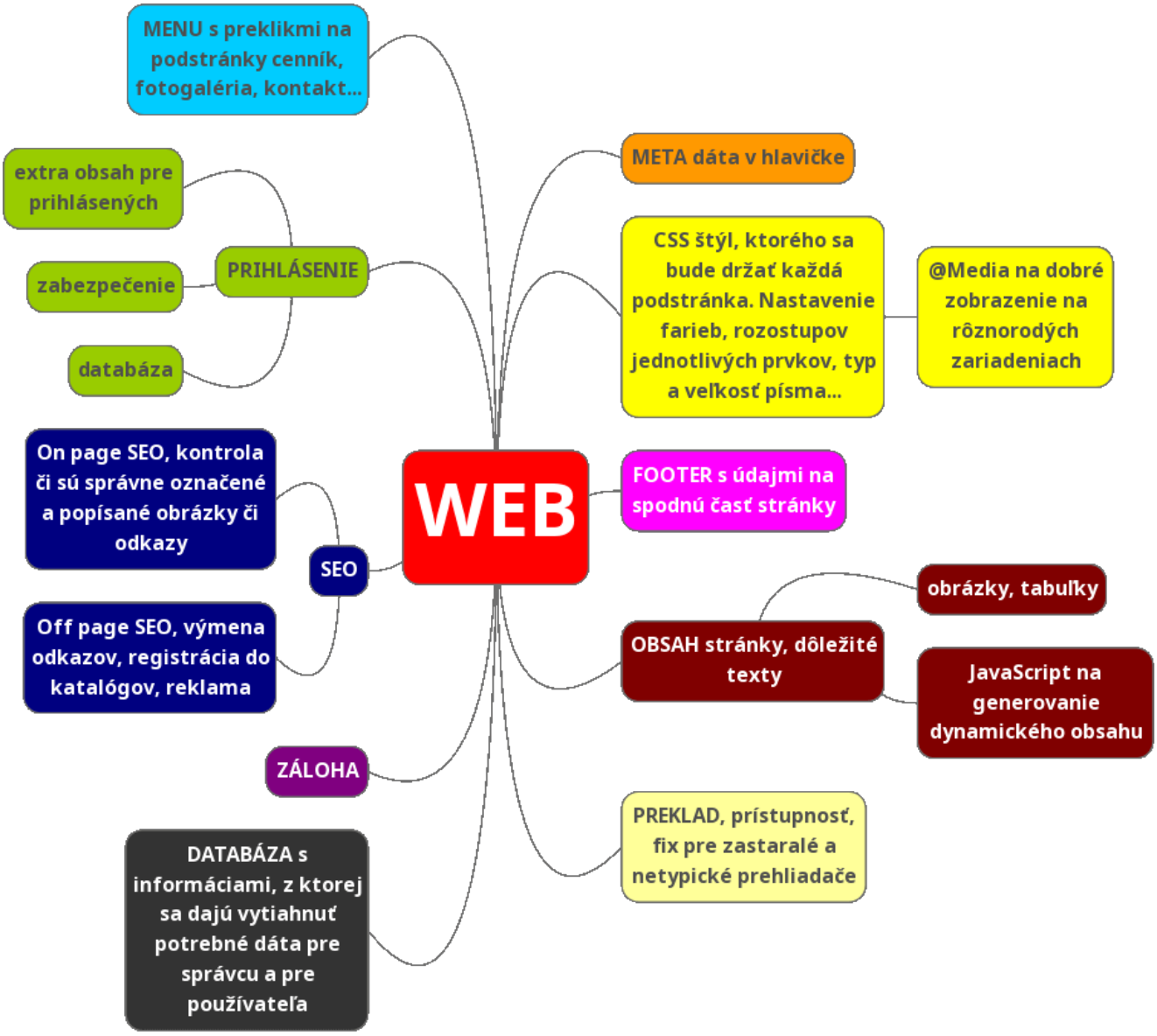
Pamätáte si na Johna, ktorý bol spomínaný v druhej kapitole? Ten, čo staval dom, ale inak ako ostatní. Postavenie domu samozrejme má byť od základov, ale on na tom išiel inak. Zohnal si človeka (alebo inú bytosť), ktorá pripraví základy, podlahy, nosné steny, strechu a dá to všetko dohromady. V podstate je jedno, kde začne, ak dokáže získať všetko potrebné a urobiť z toho dobrý celok. Na to ale potrebuje to hlavné, skúsenosti. John ide urobiť web pre svoju kamarátku Betku, ktorá ponúka ubytovanie pre turistov. Má skúsenosti, lebo v minulosti robil web pre inú kamarátku, Aničku. Samozrejme, že každá má iné požiadavky, možno mieri aj na iných návštevníkov a zároveň nový web vzniká oveľa neskôr ako ten prvý, na ktorom sa aj John učil. Web pre Aničku bol jednoduchý. Niečo sa začalo písať, do toho niečoho sa pridali obrázky, pár odkazov a celé sa to urobilo tak, ako bolo vtedy normálne a v móde. Tabuľky. Veľa tabuliek. Pre Johna je jasné, že nemôže urobiť jednoduchú kópiu prvej stránky, lebo chce prekonať originál a znovu sa chce aj niečo naučiť. Začne tak s kostrou. A to je dobrý nápad.

Kostra, štruktúra alebo iný názov je dôležitá pre prehľadnosť. Betka bude mať istotne radosť, ak bude stránka pekná, prehľadná a ak si ju môže niekto pozrieť bez ohľadu na to, či to bude cez počítač alebo telefón. Taktiež chce stránku, cez ktorú si môže niekto urobiť rezerváciu s tým, že bude mať aj ona dobrý prehľad o všetkom potrebnom. Ako na to? Tu je jednoduchý nápad, ako by to malo vyzerať:

1. Osobné stretnutie, prebratie možností, spísanie požiadaviek.
2. Začiatok projektu, dodanie textov, fotografii, informácii o okolí.
3. Návrh riešenia. V tomto prípade webová stránka, ktorá bude vyzerať dobre na rôznych zariadeniach (responzívna), čo vyžaduje dobre urobené CSS.
4. Vytvorenie prvej verzie stránky s tým, že sa zároveň testuje jej použiteľnosť. Na to postačia akékoľvek texty. Môže sa napríklad využiť a upraviť niektorá HTML+CSS šablóna.
5. Vkladanie správnych textov, obrázkov a všetkého ostatného cez pár dobre urobených funkcii (PHP)
6. Príprava vyšších funkcii, ako je prihlásenie a rezervácie.
7. Kompletné otestovanie pred spustením a kontrola stránky, či je vhodne urobená a neobsahuje technické či textové chyby.
8. Spustenie stránky a príprava stránky na to, aby sa dala ľahko nájsť cez Google, katalógy a podobne (SEO).

Samozrejme, tento postup sa dá zjednodušiť, ale výsledkom bude produkt, ktorý bude menej zaujímavý alebo viac chybový v porovnaní s tým, čo ponúkne konkurencia. Viac technológii sa použije preto, aby sa dal projekt udržovať. Dobrá kostra tiež zabezpečí, že stránka nebude jeden obrovský dokument na stovky riadkov, ale presný opak. Krátky dokument, ktorý si vždy a veľmi prehľadne vytiahne len to, čo potrebuje. Tak môže byť hlavný súbor, súbor s dlhými textami, súbor s menu a tak podobne. Ak sa zmení súbor s menu, ovplyvní to každú podstránku a nebude sa musieť meniť množstvo súborov.

Ukážka kostry webovej stránky



Keď sa urobí dobrá kostra projektu, tak nemusí všetko robiť jeden človek.

Peter vie výborne robiť s CSS a postará sa o to, aby bola stránka esteticky pekná a ľahko čitateľná.

Anežka vie písať dobré, pútavé texty. Nie je žiadna programátorka, ale jej texty sa veľmi hodia.

Matúš je expert na to, aby všetko fungovalo ako má. Pripraví databázu, zálohovanie a menu.

John sa postará o zvyšok. Až na obrázky. Povie si, že to je práca pre grafika.

Lukáš je grafik a postará sa o to, aby boli obrázky presne orezané, pôsobili jednotne a aby boli dosť malé na to, aby sa dostatočne rýchlo načítali aj ľuďom s pomalým pripojením na internet.

Ukážka HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>Hlavný nadpis</h1>

<p>Tu je veľmi jednoduchý text s klikateľným
odkazom na <a
href="https://google.com">Google.com</a>.
Samozrejme sa dá pridať akýkoľvek text,
ktorý môže byť <b>hrubým</b>,
<u>podčiarknutým</u> či napríklad <i>šikmým
písmom</i>. Alebo napríklad <b><u><i>všetkým
naraz</i></u></b>.</p>

<h2>Menší nadpis nad nákupným zoznamom</h2>
<ul>
<li>chlieb</li>
<li>mlieko (<b>polotučné!</b>)
<li><font color="green">zelené</font>
jablká</li>
</ul>

</body>
</html>
```

Hlavný nadpis

Tu je veľmi jednoduchý text s klikateľným odkazom na Google.com. Samozrejme sa dá pridať akýkoľvek text, ktorý môže byť **hrubým**, podčiarknutým či napríklad *šikmým písmom*. Alebo napríklad ***všetkým naraz***.

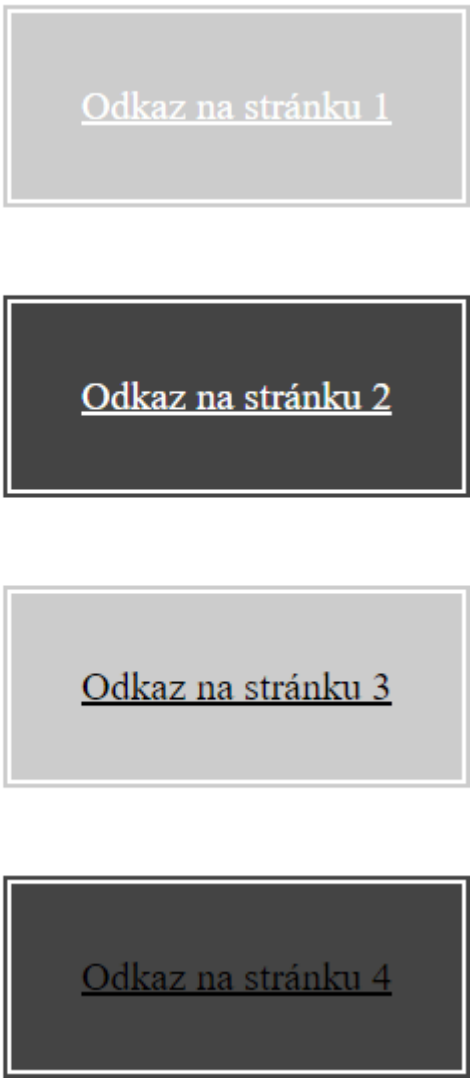
Menší nadpis nad nákupným zoznamom

- chlieb
- mlieko (**polotučné!**)
- zelené jablká

```
<html>
<body>

<h1>Hlavný nadpis</h1>

<p>Tu je veľmi jednoduchý text s klikateľným odkazom na
<a href="https://google.com">Google.com</a>. Samozrejme sa dá pridať akýkoľvek text, ktorý
môže byť <b>hrubým</b>, <u>podčiarknutým</u> či napríklad <i>šikmým písmom</i>. Alebo
napríklad <b><u><i>všetkým naraz</i></u></b>.</p>

<h2>Menší nadpis nad nákupným zoznamom</h2>
<ul>
<li>chlieb</li>
<li>mlieko (<b>polotučné!</b>)
<li><font color="green">zelené</font> jablká</li>
</ul>

</body>
</html>
```

Ukážka CSS

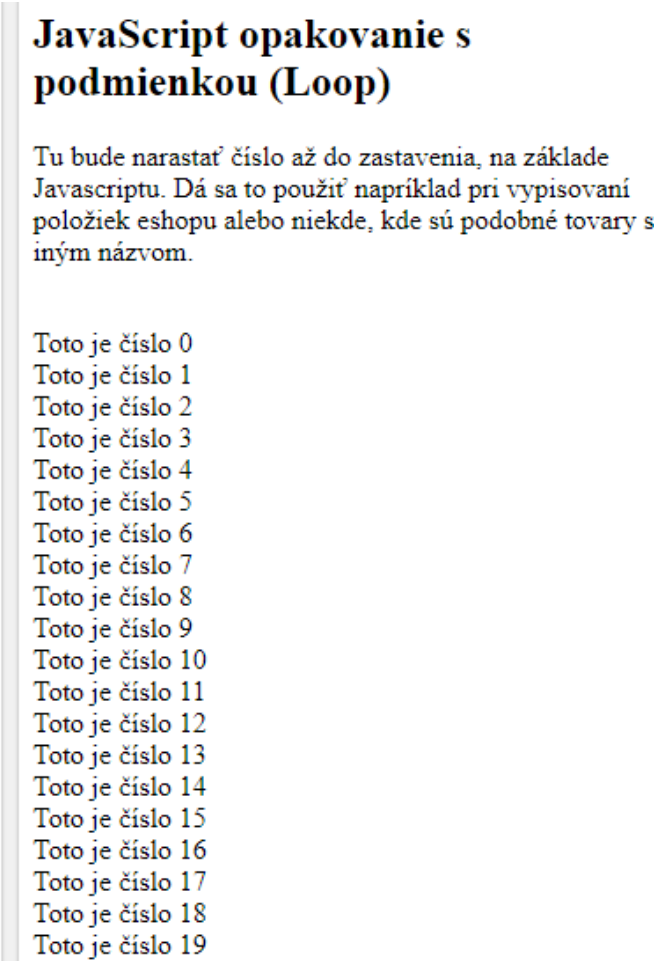
```
<html>
<head>
<style>
a:link, a:visited {
  color: white;
  padding: 35px;
  margin: 20px;
  display: inline-block;
  font-size: 20px;
  border: 6px double white;
}
a.svetle {background-color: #cccccc;}
a.tmave {background-color: #444444;}
a.ciernytext {color: black;}
</style>
</head>
<body>
<a class="svetle" href="stranka1.html">Odkaz
na stránku 1</a>
<a class="tmave" href="stranka2.html">Odkaz
na stránku 2</a>
<a class="ciernytext svetle"
href="stranka3.html">Odkaz na stránku 3</a>
<a class="ciernytext tmave"
href="stranka4.html">Odkaz na stránku 4</a>
</body>
</html>
```




```
<html>
<head>
<style>
a:link, a:visited {
  color: white;
  padding: 35px;
  margin: 20px;
  display: inline-block;
  font-size: 20px;
  border: 6px double white;
}
a.svetle {background-color: #cccccc;}
a.tmave {background-color: #444444;}
a.ciernytext {color: black;}
</style>
</head>
<body>
<a class="svetle" href="stranka1.html">Odkaz na stránku 1</a>
<a class="tmave" href="stranka2.html">Odkaz na stránku 2</a>
<a class="ciernytext svetle" href="stranka3.html">Odkaz na stránku 3</a>
<a class="ciernytext tmave" href="stranka4.html">Odkaz na stránku 4</a>
</body>
</html>
```

Ukážka JavaScript

```
<html>
<body>

<h2>JavaScript opakovanie s podmienkou
(Loop)</h2>


<p>Tu bude narastať číslo až do zastavenia,
na základe Javascriptu. Dá sa to použiť
napríklad pri vypisovaní položiek eshopu
alebo niekde, kde sú podobné tovary s iným
názvom.</p>
<p id="demo"></p>

<script>
var text = "";
var i = 0;
while (i < 20) {
  text += "<br>Toto je číslo " + i;
  i++;
}
document.getElementById("demo").innerHTML =
text;
</script>

</body>
</html>
```



JavaScript opakovanie s podmienkou (Loop)

Tu bude narastať číslo až do zastavenia, na základe Javascriptu. Dá sa to použiť napríklad pri vypisovaní položiek eshopu alebo niekde, kde sú podobné tovary s iným názvom.

Toto je číslo 0
Toto je číslo 1
Toto je číslo 2
Toto je číslo 3
Toto je číslo 4
Toto je číslo 5
Toto je číslo 6
Toto je číslo 7
Toto je číslo 8
Toto je číslo 9
Toto je číslo 10
Toto je číslo 11
Toto je číslo 12
Toto je číslo 13
Toto je číslo 14
Toto je číslo 15
Toto je číslo 16
Toto je číslo 17
Toto je číslo 18
Toto je číslo 19

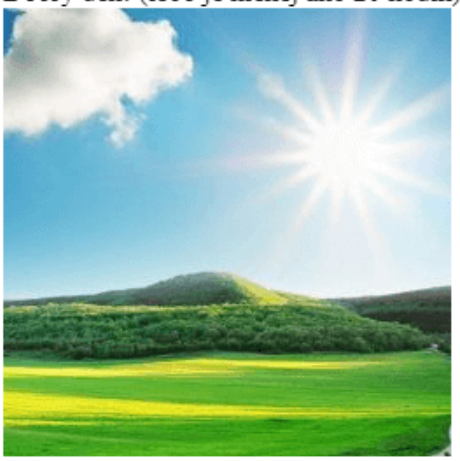
```
<html>
<body>

<h2>JavaScript opakovanie s podmienkou (Loop)</h2>


<p>Tu bude narastať číslo až do zastavenia, na základe Javascriptu. Dá sa to použiť napríklad pri vypisovaní položiek eshopu alebo niekde, kde sú podobné tovary s iným názvom.</p>
<p id="demo"></p>

<script>
var text = "";
var i = 0;
while (i < 20) {
  text += "<br>Toto je číslo " + i;
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Ukážka PHP

```
<html>
<body>

<?php
$t = date("H");
echo "<p>Server hlási, že momentálne je " .
$t;
echo " hodín, a tak ti praje:</p>";

if ($t < "10") {
  echo "Dobré ráno! (lebo je menej ako 10
hodín)";
} elseif ($t < "20") {
  echo "Dobrý deň! (lebo je menej ako 20
hodín)<br /><img src='obrazok-den.jpg'>";

} else {
  echo "Dobrú noc! (lebo nie je splnená ani
jedna z podmienok)";
}
?>

</body>
</html>
```

Server hlási, že momentálne je 19 hodín, a tak ti praje:

Dobrý deň! (lebo je menej ako 20 hodín)




```
<html>
<body>

<?php
$t = date("H");
echo "<p>Server hlási, že momentálne je " . $t;
echo " hodín, a tak ti praje:</p>";

if ($t < "10") {
  echo "Dobré ráno! (lebo je menej ako 10 hodín)";
} elseif ($t < "20") {
  echo "Dobrý deň! (lebo je menej ako 20 hodín)<br /><img src='obrazok-den.jpg'>";

} else {
  echo "Dobrú noc! (lebo nie je splnená ani jedna z podmienok)";
}
?>

</body>
</html>
```

Ukážka Python

```
# Tento program prepočíta teplotu zo stupňov
celzia (C) na fahrenheity (F)

print('Stačí zmeniť čísla v tejto časti,')
print('program to okamžite prepočíta')
print(' ')
celsius1 = 100
celsius2 = 200
celsius3 = 1538

fahrenheit = (celsius1 * 1.8) + 32
print('%0.1f C = %0.1f F' %
(celsius1,fahrenheit))

fahrenheit = (celsius2 * 1.8) + 32
print('%0.1f F zodpovedá %0.1f C' %
(fahrenheit,celsius2))

fahrenheit = (celsius3 * 1.8) + 32
print('Teplota %0.1f C (%0.1f F) taví
železo' %(celsius3,fahrenheit))
```

```
Stačí zmeniť čísla v tejto časti,
program to okamžite prepočíta

100.0 C = 212.0 F
392.0 F zodpovedá 200.0 C
Teplota 1538.0 C (2800.4 F) taví železo
```

```
# Tento program prepočíta teplotu zo stupňov celzia (C) na fahrenheity (F)

print('Stačí zmeniť čísla v tejto časti,')
print('program to okamžite prepočíta')
print(' ')
celsius1 = 100
celsius2 = 200
celsius3 = 1538

fahrenheit = (celsius1 * 1.8) + 32
print('%0.1f C = %0.1f F' %(celsius1,fahrenheit))

fahrenheit = (celsius2 * 1.8) + 32
print('%0.1f F zodpovedá %0.1f C' %(fahrenheit,celsius2))

fahrenheit = (celsius3 * 1.8) + 32
print('Teplota %0.1f C (%0.1f F) taví železo' %(celsius3,fahrenheit))
```

Užitočné odkazy

Webové stránky, ktoré vysvetľujú rôzne programovacie jazyky, vrátane príkladov:

https://www.w3schools.com/

https://www.learn-html.org/, https://www.learn-js.org/, https://www.learn-php.org/ a ďalšie

https://www.javatpoint.com/

https://www.jakpsatweb.cz/

https://www.zdrojak.cz/

https://www.root.cz/

Vypisovať veľké množstvo stránok nemá veľký zmysel, pretože to, čo potrebujete, často nájde Google v priebehu sekundy. Stačí sa len dobre opýtať. Tak napríklad, ak by ste chceli vedieť, aký je v CSS rozdiel medzi padding a margin, stačí sa spýtať Google. Toto zadanie:

https://www.google.com/search?q=css+padding+vs+margin

ma doviedlo sem:

https://www.w3schools.com/css/css_boxmodel.asp

Podobné knihy z Greenie knižnice:

http://greenie.elist.sk/knihy/linux-ako-nieco-navyse

http://greenie.elist.sk/knihy/linuxove-distribucie

Samotná Greenie knižnica používa napríklad JavaScript pri kvízoch a kompetný web je prehľadný HTML/CSS kód.

Youtube toho vie naučiť veľa. Napríklad tuto Yablko pekne preberá ako sa píše HTML kód:

https://www.youtube.com/watch?v=54bRQFfKlwk&ab_channel=Yablko

Taktiež PHP, kde ukáže spojenie programovania a jeho obľúbeného seriálu:

https://www.youtube.com/watch?v=kMq5Ix-srRg&ab_channel=Yablko

Martin Bugár má svoj názor na programátorské myslenie a ukáže, ako urobiť napríklad pyramídu:

https://www.youtube.com/watch?v=nyP6EEAiSrA&ab_channel=Kurzyprogramovania