

Tvorbba webu

Polzicování pomocí CSS



Martin Urza



Motivace



- Proč používat CSS layout?
 - Formátování a rozložení webu je oddělené od obsahu.
 - Web v CSS layoutu umí prohlížeče zobrazovat o něco rychleji (tabulky je nutné napřed celé načíst).
 - Pro dynamicky generované stránky lze za pomoci CSS snadno vytvořit různé vzhledy pro různé uživatele.
 - Frikulíni (free, cool & in) používají zásadně CSSko ;o))
- Chcete-li tvořit layout stránek pomocí kaskádových stylů, je nejprve nutné fungování CSS pochopit; není to sice nic obzvláště těžkého, avšak nelze se pouštět do pozicování bez chápání principu fungování CSS.
 - Kdo nechápe CSS, nechť si doplní znalosti.



<div>

🔪 formátování využíváme element div

- Principiálně tvoříme pozicování pomocí kaskádových stylů tak, že obalíme jednotlivé prvky (texty, odkazy, obrázky, formuláře, ...) do elementů div; těm potom nastavíme vlastnostmi kaskádových stylů umístění na stránce a velikost.
- Tyto divy lze vkládat do sebe (a typicky je to žádoucí) a to i vícenásobně (div v divu v divu...). Mnoho lidí, kteří se CSS layout učí, na tuto možnost zapomíná a komplikují si tím práci.
- Celé pozicování pomocí kaskádových stylů je o tom, jakým způsobem určovat pozici a velikost divů, což lze dělat mnoha různými způsoby.

Vlastnost position



- Vlastnost position má čtyři hodnoty:
 - Static – prvek je umístěn tam, kde by byl původně.
 - Relative – prvek je posunut z původní pozice (vlastnost top určuje posunutí dolů, vlastnost left doprava), kde ale stále zůstává prázdné místo o velikosti původního prvku. Nově umístěný prvek překryje všechny ostatní, neposune je.
 - Absolute – prvek je z dokumentu vyjmut (ostatní prvky nejsou posunuty – jako kdyby tam prvek vůbec nebyl) a je umístěn na pozici podle vlastností top a left.
 - Fixed – prvek je vyjmut z dokumentu (jako u absolute) a umístěn na fixní pozici v okně (neroluje se stránkou) prohlížeče; **nefunguje ve starých explorerech!!**

Vlastnosti top, left, bottom, right



- Pro "position: static" jsou tyto vlastnosti ignorovány.
- Možné hodnoty jsou auto (žádné posunutí), délka (v pixelech) a procenta.
- Tyto vlastnosti určují vzdálenost elementu od okrajů „omezujícího prvku“. Omezujícím prvkem může být buď nadřazený element (existuje-li a je-li pozicován absolutně nebo relativně).
- Top definuje vzdálenost od horního okraje (posunutí směrem dolů), left od levého, bottom od spodního a right od pravého.
- Všem těmto vlastnostem je možné zadávat i záporné hodnoty (a chování je očekávatelné).



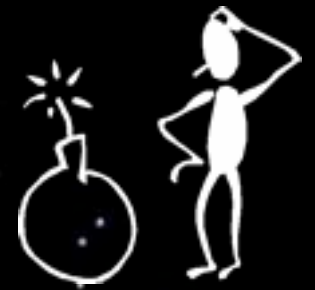
Velikost a umístění prvků

- Krom top, left, right a bottom má CSS ještě další dvě vlastnosti pro určování pozice objektů; jde o width a height (hodnoty mohou být pixely či procenta).
- Z oněch šesti vlastností jsou některé redundantní.
 - Dvě vlastnosti z trojice left, width a right určují třetí.
 - Totéž platí pro trojici top, height a bottom.
 - Velikost a umístění prvků lze jednoznačně určit čtyřmi vlastnostmi (dvěma z každé skupiny).
 - Uvedení celé trojice způsobí typicky ignorování jedné z vlastností; potíž je, že není moc jasné, která to bude.
- Nejčastěji se objekt pozicuje pomocí vlastností width, left, height a top.

Cvičení

- Napište stránku, na které bude pouze smajlík z divů; **všechny** následující podmínky musí být splněny:
 - Smajlík má dvě čtvercové (64x64 px) zelené oči.
 - Pusu smajlíku tvoří čtyři čtvercové (32x32 px) červené divy (pokuste se vytvořit úsměv, nebo zamračení).
 - Nos smajlíku je tvořen růžovým čtvercem (32x32 px).
 - Smajlík musí být **horizontálně** na středu stránky a to i v případě, že se okno prohlížeče rozšíří či zúží (pokud se okno zúží příliš, smajlík může být zdeformován).
 - Hint: Problémy s nosem může řešit záporný margin.
 - Vertikální umístění může být libovolné (rozumné).
- Budou potřeba CSS vlastnosti z minulé přednášky.

Problémy absolutního pozicování



- Opravdu hloupý způsob rozvržení stránky je umístit všechny prvky absolutně a zadat délky v pixelech.
- O něco lepší je použití procent; neřeší to ale všechny problémy, protože u některých prvků není předem známá velikost (je určena až podle obsahu).
- V případě, že chceme napsat stránky s proměnnou výškou a zápatím v nějakém divu, pochopitelně nelze určit pozici zápatí absolutně (protože nevíme, jakou výšku bude mít obsah).
- Dalším problémem absolutního pozicování může být i to, že jsou z toku dokumentu úplně vyjmuty a jsou vykreslovány přes ostatní obsah.



Vlastnost float

- Vlastnost float má tři hodnoty (none/left/right), jež určují, jak bude objekt obtékán.
 - Hodnota left/right znamená, že bude objekt umístěn vlevo/vpravo a obsah jej bude obtékat zprava/zleva.
 - None je klasické nastavení, žádné obtékání.
- Více objektů se stejně nastaveným obtékáním (left či right) se naskládá horizontálně vedle sebe, dokud se vejdou.
 - Tak lze udělat např. menu (float: left) a obsah stránky.
- Vlastnost clear má čtyři hodnoty (none/left/right/both), která říká, že prvek nesmí obtékat (tedy má být zařazen až pod obtékaný) a z které strany.

Cvičení



- Napište stránku, která má záhlaví (místo nahoře na nadpis), levé menu (fixní šířky), obsah (zbytek šířky stránky) a zápatí (dole bude například copyright na středu).
- **Důležité je, aby zápatí bylo vždy přímo pod menu a obsahem bez ohledu na to, co z toho je zrovna vyšší** (je-li tedy vyšší menu, je zápatí zarovnané pod menu, je-li vyšší obsah, zápatí bude zarovnané pod obsahem).
- Při testování stránky musíte pochopitelně vložit do menu i obsahu nějaký text – ten však nemusí dávat smysl, jde jen o layout.

Horizontální centrování plovoucích prvků



- Neznáme-li šířku centrovaného prvku (nebo se tato může měnit) a známe-li šířku okrajů, je horizontální centrování jednoduché. Nastavíme prostě elementu `margin-left` a `margin-right` na stejnou hodnotu (lze ji zadat i v procentech), čímž definujeme šířku okrajů a zbylá šířka rodičovského elementu je vyhrazena pro šířku centrovaného prvku.
- Příklad stylu centrovaného prvku:
 - `"margin-left: 25%; margin-right: 25%"` Vycentrovaný prvek má šířku 50% nadřazeného tagu a 25% okraje.
 - `"margin-left: 16px; margin-right: 16px"` Vycentrovaný prvek má 16px okraj a šířku rodiče minus 32px.



Horizontální centrování pevných prvků

- Chcete-li vycentrovat element s neměnnou známou šířkou, nastavte tomuto elementu onu šířku (můžete i v procentech) a `margin-left` i `margin-right` nastavte na hodnotu `auto`. To samo o sobě ještě nestačí, navíc je nutné nastavit rodičovskému (nadřazenému) tagu vlastnost `text-align: center` (netýká se jen textu, ale všech prvků); to ale zdědí i centrovaný tag, což lze napravit nastavením `text-align: left`.
- Příklad:
 - `"margin-left: auto; margin-right: auto; width: 64%; text-align: left"` (centrovaný prvek)
 - `"text-align: center"` (rodičovský/nadřazený prvek)



Vertikální centrování pevných prvků

- Chceme-li výšku centrovaného prvku $N\%$, nastavíme `height: N%` a `top: ((100-N)/2)%` (za N dosadíte číslo).
- Má-li být výška centrovaného prvku Npx , nastavíme `height: Npx`, `top: 50%` a `margin-top: ((-1) * N/2)px`.
- Aby „fungovaly“ vlastnosti `top` a `height`, pochopitelně je nutné nastavit `position: absolute`.
- Příklady:
 - `"position: absolute; height: 64%; top: 18%"`
 - `"position: absolute; height: 128px; margin-top: -64px; top: 50%"`
 - Je pochopitelně možné prvky posouvat i jinam než na střed.

Vertikální centrování plovoucích prvků



- Obecně jde o velmi ošklivý problém, který zatím asi není možné vyřešit bez obezliček, hacků a špinavých triků.
- Největší potíž je v tom, že řešení pro MSIE se výrazně liší od řešení pro ostatní prohlížeče; navíc v případě, že se tyto kódy sloučí do jednoho, vzájemně se vyruší a přestanou fungovat zcela.
- Budete-li někdy chtít vertikálně vycentrovat plovoucí objekt proměnlivé výšky, googlujte; snad bude časem možné najít lepší řešení, než je k dispozici teď (to je velmi kostrbaté, proto jej neuvádím).
- Pokuste se tomuto problému pokud možno vyhnout.

Závěr



- Na rozdíl od programování, které je činností tvůrčí, je tvorba webu činnost spíše tupá. Programátorská práce spočívá v myšlení; tvůrce webu zejména hledá.
- Když máte problém, pravděpodobně jej měl již někdo dřív, vyřešil ho a o řešení se podělil se světem, takže nejlepší rada zní: **googlovat, googlovat, googlovat!**
- Obecně nenarazíte na problém, jenž by neměl řešení, ale budete-li se tvorbou webu zabývat, občas potkáte problém, jehož řešení je přinejmenším úchylné.
- CSS má stovky vlastností, nebylo možné probrat vše, doporučuji vyhledat např.: display, cursor, overflow, z-index a další....

Rekapitulace



- CSS layout poskytuje mnohé výhody a rozhodně má smysl pomocí CSS pozicovat všechny větší weby; když už ne pro přehlednost, pak kvůli rychlosti.
- **Nezapomínejte na vkládání divů do jiných divů!!**
Některé věci bez toho řešit téměř nejdou, jiné velmi obtížně. Na každé trochu komplikovanější stránce se s tím setkáte.
- Kdykoliv máte problém, lze jej snadno vygooglovat.
 - Temnou stranou síly je, že mnohdy takové návody píší lidé, kteří o tom nic neví, neboť web může tvořit každý moula (a tím spíš o tom poučovat ostatní).
- Pozor na nekompatibilitu prohlížečů!!