

# Informatika pro moderní fyziky (8)

## CSS - stylování dokumentů, SVG - tvorba obrázků, složitější interaktivní dokument

František HAVLŮJ

*e-mail: haf@ujv.cz*

ÚJV Řež

oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2013/2014

3. prosince 2013

- 1 Co jsme se naučili minule
- 2 Něco o kódování
- 3 Stylování dokumentů
- 4 Tvorba obrázků

# Obsah

- 1 Co jsme se naučili minule
- 2 Něco o kódování
- 3 Stylování dokumentů
- 4 Tvorba obrázků

- HTML pro běžné použití
- procvičení ERb
- základy použití JS

# Obsah

- 1 Co jsme se naučili minule
- 2 Něco o kódování**
- 3 Stylování dokumentů
- 4 Tvorba obrázků

# Kódování

- znaky v počítači: 1 znak = 1 byte = 256 možností
- cca polovina je “normální text”, zbytek jsou tak trochu speciální znaky
- jsou tam ale jen ‘západní’ znaky, na středoevropské se nedostalo
- co teprv azbuky, japonština, čínština, ...

# Kódování

- varianta 1: nahrazovat druhou polovinu znaků tím, co zrovna potřebuju (ěščř...)
- výhoda: nezabírá místo, pořád platí znak=byte, jednoduché řešení
- nevýhoda: pro různé jazyky různá kódování
- další nevýhoda: na leckterý jazyk (neevropský) to naprosto nestačí

# Kódování

- varianta 2: přejít na reprezentaci jednoho znaku více byty
- výhoda: podstatně se rozšíří počet znaků, takže není nutné 'přepínat'
- nevýhoda: zvětšuje velikost textu
- další nevýhoda: míra rozsypanosti čaje při špatné interpretaci se zvyšuje (je možné způsobit i nečitelnost 'obyčejných' znaků)



## Kódování - obecné problémy

- soubory neobsahují informaci o tom, v jakém kódování jsou, takže je nutné dodávat metadata
- manipulace s kódováním je otravná, obtížná a snadno způsobí problémy
- je nutné synchronizovat/řešit kódování na všech vrstvách aplikace (soubory / databáze / databázový klient / aplikace / server / klient)
- ... pokud existuje 100 různých variant, řeší se to obvykle tím, že se vymyslí nějaká 101. ...

# Kódování

- naštěstí to (minimálně v euroamerickém prostředí) už dneska nemusíme řešit, existuje jednotný standard – Unicode (UTF-8)
- starší systémy občas poskytují data v jiných kódováních, je dobré vědět, že pro češtinu se můžete setkat s Windows/CP1250 a ISO-9981-2
- pokud budete chtít pracovat s exotickými jazyky (Afrika, Oceánie), budete asi muset využít UTF-16/32 (naneštěstí opět více variant)
- z historických důvodů je nejednotnost také pro japonská kódování
- to hlavní: používejte UTF-8. Tečka. Nezapomeňte: v editoru, v hlavičce HTML, v hlavičce LaTeXu, všude...

# Obsah

- 1 Co jsme se naučili minule
- 2 Něco o kódování
- 3 Stylování dokumentů**
- 4 Tvorba obrázků

# CSS

- jazyk pro popis vzhledu HTML dokumentu
- v nejjednodušším přístupu definuje styly pro jednotlivé typy elementů
- dále umožňuje definovat tzv. třídy (skupiny elementů, pomocí atributu `class` v HTML) a také styly pro konkrétní elementy (`id`)
- složitější selekory – vnoření, souslednost atd.

## CSS - jednoduchý příklad

- základní selektory: *element*, *.třída* a *#id*
- základní syntaxe *vlastnost: hodnota*;

```
a {  
  color: blue;  
  text-decoration: none;  
  font-weight: bold;  
}  
#core_map { float:left; }
```

## CSS - kam s ním

- přímo k tagu (`<div style="display:none">`) – možná dobré na rychlé ladění/patlání, ale vesměs vždy špatně; jedinou výjimkou je `display:none` pro elementy, které mají být vidět až později, tam to jinak nejde
- v hlavičce (head) HTML dokumentu: `<style type="text/css">...</style>`
- v externím souboru (obvykle jediné správné řešení) – pomocí tagu `link` v hlavičce
- my vystačíme se `style` tagem v hlavičce

## Rychlé procvičení

- vezměte si svoje krásné HTML z minula a doplňte do něj trochu toho stylování
- minimum: změnit font (`font-family`), nastavit rozumné velikosti písma a barvy
- zrušit podtrhávání odkazů (`text-decoration`)
- bonus: jiná barva odkazů na přepínání tabulka/graf

# Obsah

- 1 Co jsme se naučili minule
- 2 Něco o kódování
- 3 Stylování dokumentů
- 4 Tvorba obrázků**



## Zadání dnešní úlohy

- každý den data z 1-9 detektorů (`data/*.csv`)
- detektor má svoji polohu v AZ (VR-1 Vrabec, 8x8 čtvercových pozic) – včetně data je uvedena na prvním řádku CSV souboru
- je potřeba hezky zobrazit na každý den mapu AZ a grafy signálů z detektorů
- viz `html/document.html`

## Jak na obrázky

- pěkný formát na tvorbu vektorových obrázků je SVG (Scalable Vector Graphics)
- je to dobrá věc především na internet – všechny prohlížeče ho umí
- stejně jako HTML je postaven na XML, takže už to vlastně umíme

## Jednoduchý příklad

```
<svg width="320" height="320" xmlns="http://www.w3.org/2000/svg" version="1.1">  
  <rect x="0.0" y="0.0" width="40.0" height="40.0" fill="blue" />  
  <rect x="40.0" y="0.0" width="40.0" height="40.0" fill="red" />  
  <rect x="0.0" y="40.0" width="40.0" height="40.0" fill="green" />  
  <rect x="40.0" y="40.0" width="40.0" height="40.0" fill="yellow" />  
</svg>
```



## SVG – co a jak

- souřadný systém z levého horního rohu
- je potřeba udat celkovou šířku a výšku
- zatím nám stačí obdélník – tag `rect`
- pozor, je to striktní XML, tedy je nutné `rect` tag uzavřít (!)
- vyzkoušejte – nejdřív jen tak, potom vygenerovat 8x8 mapu (zatím klidně prázdnou)

## Jak vložit do HTML

- jsou různé metody, jak vložit ze souboru
- my se bez toho v pohodě obejdeme - vložíme přímo (`IO.read`)
- v tu chvíli je totiž mj. možné naplácat na SVG objekty javascriptové handlers
- ... tedy na čtverečku můžu mít onclick

## Zpracovat data

- pro každý CSV soubor chceme mít graf (gnuplot/png)
- pro každý den mapu (nejdřív obyčejnou, potom klikací, pak třeba s textem)

## Další JS chytrosti

- v jQuery už známe `$ ( '#id' )`
- ale ve skutečnosti jde použít jakýkoli CSS selektor, takže třeba `$ ( 'p' )`
- pokročilý CSS selektor – vnoření: `#my_list img` vybere všechny obrázky (`img`) které jsou uvnitř elementu s `id my_list`
- ... použiju v situacích, kdy chci schovat nějakou množinu elementů a pak jeden z nich zobrazit (tj. když mám hromadu obrázků a chci, aby byl vidět jen jeden)

## Další CSS chytrosti

- normálně se jednotlivé elementy řadí pod sebe
- můžu místo toho použít tzv. floating, kdy se začnou elementy řadit nalevo nebo napravo
- efekt znáte např. z webových galerií, kde mi fotky vyplní celou šířku okna a jdou po řádcích
- `float:left`
- barvu pozadí nastavím např. `background-color:red` nebo `background-color:#dddddff`



## Další SVG chytrosti

- kromě `rect` se bude hodit také `text`
- jako `text` se zobrazí obsah příslušného elementu
- opět použiju atributy `x`, `y` (levý dolní roh) a můžu přihodit `text-anchor="middle"`, aby to byl dolní prostředek
- pozor, `text` mi překryje čtvereček, takže budu muset zopakovat `onclick`!

Co jsme se naučili minule  
Něco o kódování  
Stylování dokumentů  
Tvorba obrázků

A to je vše, přátelé!

