

# Informatika pro moderní fyziky (9) inteligentní šablony, procvičení ERb+LaTeX, tvorba vektorové grafiky

František HAVLŮJ

*e-mail: haf@ujv.cz*

ÚJV Řež

oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2017/2018

6. prosince 2017

1 Na šablony chytře

2 Tvorba obrázků

# Obsah

- 1 Na šablony chytře
- 2 Tvorba obrázků

# Úskalí šablon

- snadno umíme nahradit jeden řetězec druhým
- trochu méně pohodlné pro větší bloky textu
- navíc by se hodila nějaká logika (cyklus) přímo v šabloně
- naštěstí jsou na to postupy

# ERb (Embedded Ruby)

- lepší šablona - “aktivní text”
- používá se například ve webových aplikacích
- hodí se ale i na generování latexových dokumentů, resp. všude, kde nám nesejde na whitespace
- poměrně jednoduchá syntax, zvládne skoro všechno (viz předmět MAA3)

# Základní syntaxe ERb (1)

Jakýkoli Ruby příkaz, přiřazení, výpočet ...

```
<% a = b + 5 %>  
<% list = ary * ", " %>
```

## Základní syntaxe ERb (2)

Pokud chci něco vložit, stačí přidat rovnítko

```
<%= a %>
```

```
<%= ary[1] %>
```

```
<%= b + 5 %>
```

## Základní syntaxe ERb (3)

Radost je možnost použít bloky a tedy i iterátory apod.  
v propojení s vkládaným textem:

```
<% (1..5).each do |i| %>  
Number <%= i %>  
<% end %>  
<% ary.each do |x| %>  
Array contains <%= x %>  
<% end %>
```



## ERb – shrnutí

- dobrý sluha, ale špatný pán
- můžu s tím vyrobit hromadu užitečných věcí na malém prostoru
- daň je velké riziko zamotaného kódu a nízké přehlednosti (struktura naprosto není patrná na první pohled, proto je namístě ji držet maximálně jednoduchou)

## Důležité upozornění

- oddělení modelu a view
- přestože lze provádět zpracování dat a výpočty přímo v ERb, je to nejvíc nejhorší nápad
- je chytré si všechno připravit v modelu (tj. v Ruby skriptu, kterým data chystáme)
- a kód ve view (tj. v ERb šabloně) omezit na naprosté minimum

# Jak ze šablony udělat výsledek

## Příklad překladač ERb

```
require 'erb_compiler'  
  
erb(template, filename, {:x => 1, :y => 2})
```

## Příklad – kreslení grafů z minula

### template.gp

```
set terminal png
set output "plot_<%=n%>.png"
plot "data_<%=n%>.csv"
```

```
(1..10).each do |i|
  erb("template.gp", "plot_#i.gp", {:n => i})
end
```

# Takže v latexu třeba

```
\subsection{Koncentrace kyseliny borité}  
  
<% files.each do |f| %>  
  
\subsubsection{Kampaň <%= f.split('_').last %>}  
\begin{center}  
\includegraphics[width=0.8\textwidth]{<%= f %>_bc.eps}  
\end{center}  
<% end %>
```

## A teď už to jenom dejte dohromady...

- 1 připravit si základní kostru dokumentu v latexu
- 2 převést na šablonu: mít seznam souborů, správně generovat kapitoly
- 3 vyrobit grafy
- 4 vložit grafy do šablony
- 5 vyrobit tabulky
- 6 vložit tabulky do šablony
- 7 A JE TO!

# Obsah

- 1 Na šablony chytře
- 2 Tvorba obrázků

## Zadání dnešní úlohy

- pro zadanou textovou mapu AZ VR1 potřebuju udělat hezký obrázek
- co druh, to barvička, rozumně zacházet s odstíny (palivo různě modré, R/B/E tyče různě červené, zelené, fialové)



# 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

# 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)

## 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 přiřadit `text-anchor="middle"`, aby to byl dolní prostředek

# Postup

- načtu ze souboru třeba do 2D pole
- budu mít hash s barvičkama
- vykreslím do SVG

A to je vše, přátelé!

