

# Informatika pro moderní fyziky (6)

## Chytré šablony a interaktivní dokumenty

František HAVLŮJ

*e-mail: haf@ujv.cz*

ÚJV Řež

oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2013/2014

19. listopadu 2013





- používání klávesových zkratk
- generování většího množství grafů, práce se všemi soubory v adresáři
- tvorba tabulek v LaTeXu, jejich vložení do dokumentu

V čem se liší tyto příkazy a co znamenají?

```
puts a  
puts "a"  
puts "#{a}"
```



- soubory `data_*.csv` - data ve třech sloupcích (datum, koncentrace kyseliny borité v chladivu, axiální offset)
- vyrobené tabulky (jistě si všichni zpracovali doma) (`ao_c??.tex`, `bc_c??.tex`)
- hezké grafy (`ao_c??.eps`, `bc_c??.eps`)

- slepit všechno do jednoho dokumentu – s hezkou strukturou, s vloženými
- de facto potřebujeme vyrobit celý LaTeX dokument automaticky (kromě hlavičky) vyrobit automaticky
- jaké máme možnosti?





- 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

- 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)

Jakýkoli Ruby příkaz, přiřazení, výpočet ...

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

Pokud chci něco vložit, stačí přidat rovnítko

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

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

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

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 %>
```

- 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)

- 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



## Příklad překladač ERb

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

### 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
```

```
\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!



- PDF zpráva je fajn, ale umožňuje jen “plochou” strukturu
- často je potřeba prezentovat tak velké množství informací, že to nejde jenom nalepit za sebe
- hodilo by se něco klikacího
- příklad – ASTRID (výstup z programu ANDREA)



- trochu divný jazyk, historicky vzniknul jako ideový koncept a omylem se z toho stala definitivní verze
- potvora s nejasnou syntaxí a dost netradiční
- naštěstí na většinu základního využití





