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 2014/2015
29. října 2013

Obsah

- používání klávesových zkratek
- úvod do LaTeXu
- tvorba tabulek v LaTeXu, jejich vložení do dokumentu

Obsah

Co už máme

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

Co nám ještě chybí

- 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?

Vektorové a rastrové formáty

- rozdíl asi každý zná
- pro tisk buď potřebuju vektor anebo 300 dpi rastr
- pokud to není fotka nebo sken, tak je vždycky lepší vektor než rastr
- LaTeX i Gnuplot podporují EPS (Encapsulated Postscript), další populární formáty jsou AI, WMF, DWG
- PDF je metaformát (ale hodně se používá i pro vektory...)

Obsah

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 překladu ERb

```
require_relative 'erb_compiler.rb'

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

Lokální proměnné v ERb šabloně

funkce `erb` je chytře napsaná tak, že umožňuje vložit libovolné proměnné pomocí hashe (tedy v příkladu výše budou v šabloně definovány proměnné `x` a `y`)

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

Výhody HTML+JS

- 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)

Výhody HTML+JS

- absolutní přenositelnost
- použitelné ve webových aplikacích i offline
- celkem snadno se zařídí, aby to vypadalo dobře
- JS zajišťuje slušný stupeň interaktivity

HTML

- poměrně rozumný jazyk
- byť trochu “ukecaný”, tak struktura je dostatečně jasná a přehledná

javaScript

- 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 není potřeba mu moc rozumět (já jsem důkazem)

Dnes jen samotné HTML

- příklad viz report.html
- základem jsou tagy – stromová struktura
- `<tag> ... </tag>`
- `<tag atribut=”hodnota”> ... </tag>`
- základní strukturu (hlavička) obšlehněte ze vzoru

Úkol trochu nadlouho - ale jednoduchý

- to samé co je v LaTeXu udělat v HTML
- příště doplníme interaktivitu

A to je vše, přátelé!