

Informatika pro moderní fyziky (10)

Generování vektorových obrázků. Konfigurační soubory YML, formát JSON, použití cizích API

František HAVLŮJ

e-mail: haf@ujv.cz

ÚJV Řež

oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2020/2021

7. prosince 2020

- 1 Co jsme se naučili minule
- 2 Tvorba obrázků
- 3 Persistence dat a formát JSON
- 4 Použití cizích API

Obsah

- 1 Co jsme se naučili minule
- 2 Tvorba obrázků
- 3 Persistence dat a formát JSON
- 4 Použití cizích API

- použití ERb šablon (dokončení)
- základy tvorby obrázků v SVG

Obsah

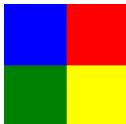
- 1 Co jsme se naučili minule
- 2 Tvorba obrázků**
- 3 Persistence dat a formát JSON
- 4 Použití cizích API

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)

Jednoduchý příklad SVG

```
<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>
```



Co už máme

- vykreslení šachovnice v SVG
- včetně souřadnic

Co musíme udělat

- načtu konfiguraci AZ ze souboru třeba do 2D pole
- budu mít hash s barvičkama (#rrggbb formát je skoro nejjednodušší, #000000 černá, #0000ff modrá)
- vykreslím do SVG - barva podle typu elementu, text typ elementu

Barvy elementů

Nejjednodušší je pro začátek mít to v nějakém hashi ve zdrojáku:

```
colors = {  
    "R1" => "#ff0000",  
    "R2" => "#ff4400",  
    ...  
}
```

Hash je pro tyto účely ideálním řešením – snadno získám barvu pro daný element `colors[element]`

Co s variantou nedefinovaných elementů

- v posledním souboru jsou navíc neznámé typy XX a YY – vhodně vyřešte:
- a) náhodná barva `rand, rand(123), "%02x" % 254`
- b) seznam barev pro neznámé typy
- vylepšení: načítáme barvy ze souboru! mohlo by to vypadat takto: (ale použijeme YAML - viz dále)

```
F4    ff0000  
R1    00ffff  
R2    00eeee
```

YAML - kamarád pro konfigurační soubory

- je trochu nuda pořád ručně načítat soubory a parsovat je, normálně se to tak nedělá – použiju standardizovaný formát souboru
- na konfigurační soubory je skvělý formát YAML (YML) – jednoduchý hash formátovaný odsazením (pozor - mezery, ne taby)

YAML - použití

- načtu standardní knihovnu `require "yaml"`
- parsování řetězce
`YAML.load(File.read("config.yml"))`
- uložení dat `File.write("config.yml",
data.to_yaml)`

YAML - kamarád pro konfigurační soubory

- přesuňte hash s barvami do konfiguračního souboru
- s hvězdičkou: RRGGBB, #RRGGBB i R,G,B

```
colors:  
  F4: ff0000  
  R1: 00ffff  
  R2: 00eeee
```

Obsah

- 1 Co jsme se naučili minule
- 2 Tvorba obrázků
- 3 Persistence dat a formát JSON**
- 4 Použití cizích API

Ukládání strukturovaných dat

- často mám data ve formě struktury (kombinace hash+pole, různý stupeň vnoření)
- z různých důvodů můžu chtít data uložit na disk a pak je znovu načítat
- (zejména efektivita zpracování, případně data z externích/webových zdrojů)
- bylo by dobré mít možnost uložit a načíst rovnou celý hash
- odpověď jsou strukturované metaformáty - YAML, XML, JSON

Práce s JSON

- v Ruby je k mání knihovna – `require "json"`
- generování JSON: `hash.to_json`
- čtení JSON: `JSON[data]`

Příklad – výsledky běhu kolem rybníka

- dva soubory – `ages.csv`, `times.csv`
- chci v jednom skriptu (tasku) načíst, spárovat a uložit
- a v jiném už rovnou načíst zpracovaná data
- a vypsat tabulku výsledků včetně ročníků narození
- pozor! JSON nezná symboly, uloží se jako řetězce

Obsah

- 1 Co jsme se naučili minule
- 2 Tvorba obrázků
- 3 Persistence dat a formát JSON
- 4 Použití cizích API**

K čemu to?

- spousta informací na webu je poskytována ve strojově čitelné formě
- API – rozhraní mezi aplikacemi
- s využitím webových služeb naše možnosti exponenciálně rostou (počasí, doprava, mapy, atd atd.)
- spousta věcí se dá udělat jako *mashup* – sice nic neumím, ale umím to dát dohromady

Typy / formáty

- URL – rovnou dostanu např. obrázek po zadání správného URL
- XML – velmi obecný, ale komplikovaný formát (“vypadá jako HTML”)
- JSON – velmi jednoduchý a kompaktní formát, vyvinutý pro JS (v podstatě jen číslo, řetězec, pole, hash)

URL API – google maps

- stačí správně vymyslet
- pozor na usage limits (v produkci je nutné lokální cache...)
- QR platba:

`http://qr-platba.cz/pro-vyvojare/restful-api/#generator-czech-image`

- Google Maps static API:

`https://developers.google.com/maps/documentation/staticmaps/`

Jednoduchý mashup: mapa o-závodů

- ORIS API – <http://oris.orientacnisporty.cz/API>
- úkol: vypíšme kalendář MTBO závodů v roce 2017
- <http://oris.orientacnisporty.cz/API/?format=json&method=getEventList&sport=3&datefrom=2017-01-01&dateto=2017-12-31>
- <https://developers.google.com/maps/documentation/static-maps/intro>
- API klíč – v praxi si musíte pořídit vlastní (ale nic to nestojí)
- AlzaSyCCADVlgtMz1a-GrFAHi_Jnvi2ofRIMuJk

A to je vše, přátelé!

