

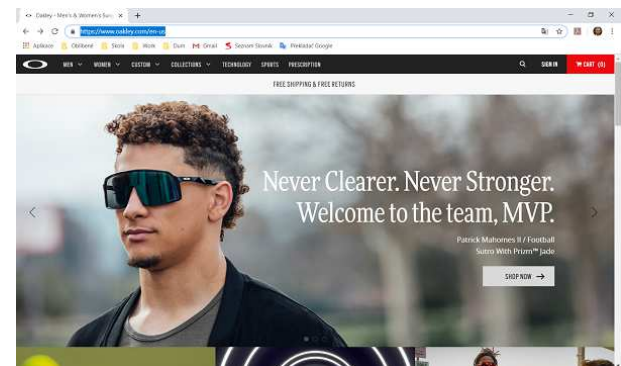
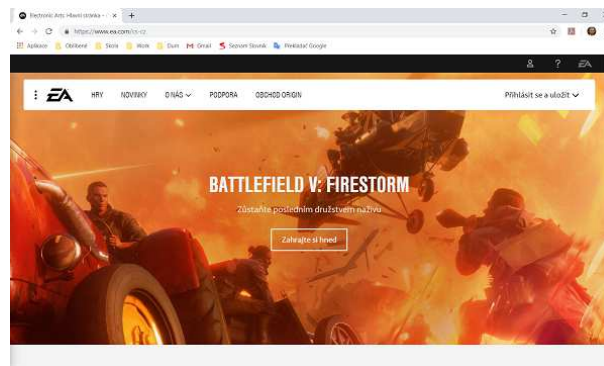
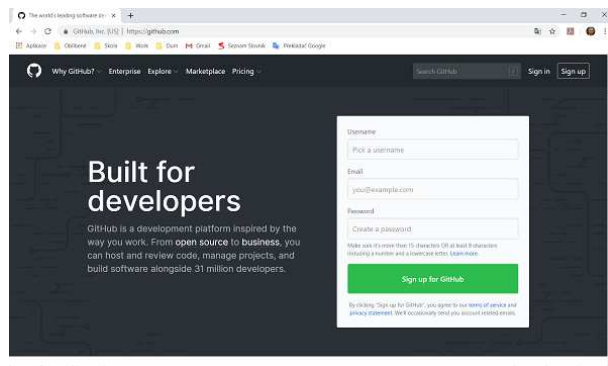
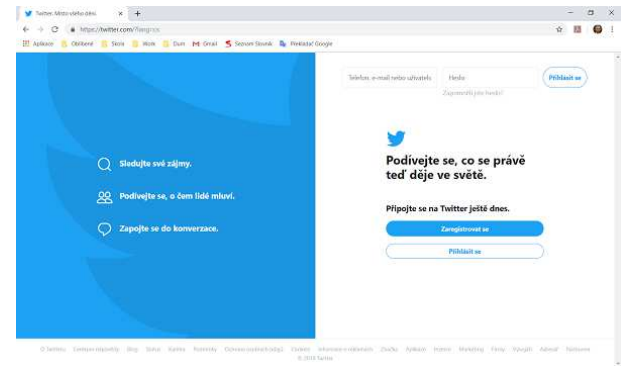
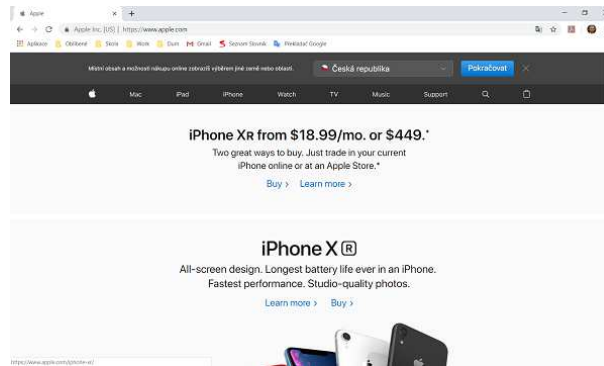
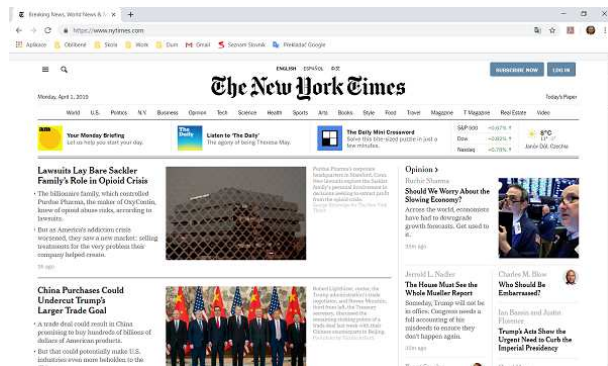
# Ruby

doc. Ing. Josef Chaloupka, Ph.D.



# Ruby on Rails

- Apple, Oakley, The New York Times, Twitter, Electronic Arts, Github...



# Ruby

- Ruby (rubín) 1993 – 1995 (první verze); autor >>> Yukihiro Matsumoto (Matz); Python – málo objektový, Perl – málo výkonný
- plně objektově orientovaný – vše v Ruby je objekt
- webový framework Ruby on Rails (náhrada PHP)
- psaní skriptů; velkých projektů (vše je objekt); CGI (Common Gateway Interface) skriptů; GUI (Graphical User Interface) aplikací; zpracování textu >>> podpora regulárních výrazů
- + interpretovaný jazyk; přenositelnost zdrojového kódu mezi OS; interaktivní režim; jednoduchá syntaxe; plná podpora objektově orientovaného programování - vše je objekt; dynamické datové typy; regulární výrazy; modularita zdrojových kódů; GUI a webové aplikace
- - interpretovaný jazyk

# Ruby On Rails

- framework pro vytváření webových aplikací v jazyku Ruby
- tvůrce >>> dánský programátor David Heinemeier Hansson >>> 2004
- <http://guides.rubyonrails.cz/> [http://guides.rubyonrails.org/getting\\_started.html](http://guides.rubyonrails.org/getting_started.html)
- implementace architektury **MVC** Model→View→Controller
- Výhody MVC
  - oddělení aplikační logiky od uživatelského rozhraní
  - znovupoužitelnost kódu
  - přehledná struktura kódu aplikace, která usnadňuje údržbu
- Model >>> reprezentuje data v aplikaci a pravidla pro práci s nimi; interakce s tabulkou v databázi; jedna tabulka = jeden model (většinou)
- View >>> uživatelské rozhraní >>> HTML kód s vloženým kódem Ruby; views poskytuje data prohlížeči, který zasílá aplikaci požadavky
- Controller >>> spojka mezi Modelem a View; zpracovává požadavky z webového prohlížeče; získává data z Modelů a posílá je do View

# Ruby On Rails

- podpora automatizovaného testování jednotlivých vrstev aplikace
- Apple, Oakley, The New York Times, Twitter, Electronic Arts, Github, Yellow Pages..
- „nový“ přístup programování; nutnost zbavit se starých návyků >>> efektivnější
- přístup **REST** (Representational State Transfer)

>>> architektura >>> vzor pro webové aplikace

webová aplikace >>> soubor zdrojů (resources\_) + HTTP „sloves“ (\_verbs)

nejsnazší a nejrychlejší způsob, jak modelovat entity a jejich vztahy

- DELETE /photos/17 >>> odkaz -> obrázek s ID 17; akce -> smazat
- HTTP nezná metodu delete – o to se však nemusíme starat

# Ruby On Rails

- podpora automatizovaného testování jednotlivých vrstev aplikace
- Apple, Oakley, The New York Times, Twitter, Electronic Arts, Github, Yellow Pages..
- „nový“ přístup programování; nutnost zbavit se starých návyků >>> efektivnější
- přístup **REST** (Representational State Transfer)

>>> architektura >>> vzor pro webové aplikace

webová aplikace >>> soubor zdrojů (resources\_) + HTTP „sloves“ (\_verbs)

nejsnazší a nejrychlejší způsob, jak modelovat entity a jejich vztahy

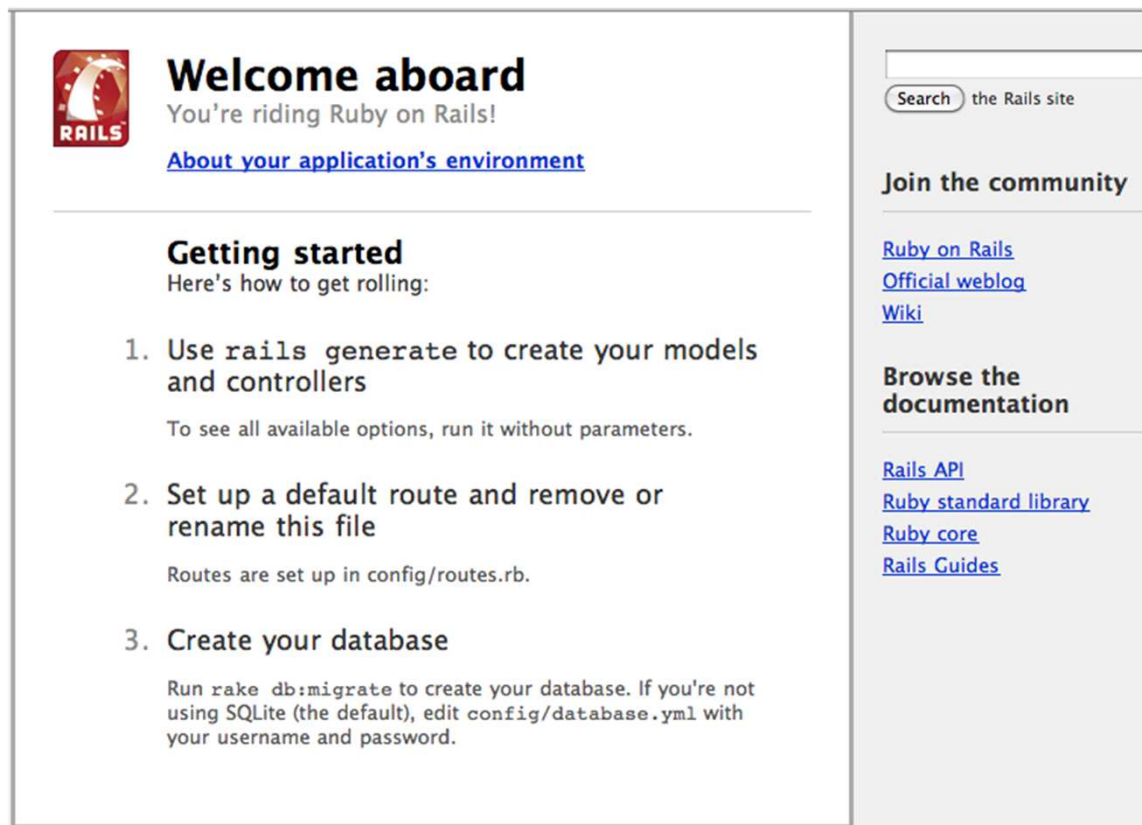
- DELETE /photos/17 >>> odkaz -> obrázek s ID 17; akce -> smazat
- HTTP nezná metodu delete – o to se však nemusíme starat

# Ruby On Rails

- Nová aplikace:
- 1) instalace balíčkovacího systému RubyGems: Gem -> klenot  
\$ gem install rails
- 2) vytvoření aplikace (blog) – help pro rails >>> rails new -h  
\$ rails new blog  
vytvoří adresář blog s podadresáři
- 3) instalace potřebných gemů: definice potřebných gemů pomocí nástroje Bundler  
bundle install
- 4) konfigurace databáze >>> MySQL, SQLite3, PostgreSQL...
- 5) vytvoření databáze  
\$ rake db:create

# Ruby On Rails

- Hello, World:
- 1) spuštění webového serveru (WEBrick, Phusion Passenger, Thin nebo Unicorn)  
\$ rails server



Výchozí informační stránka Rails: <http://localhost:3000>



# Ruby On Rails

- Hello, World:
- 2) vytvoření Controlleru a View:  
\$ rails generate controller home index  
  
změna v souboru index: <h1>Hello, World!</h1>
- 3) nastavení domovské stránky  
smazání původního souboru: \$ rm public/index.html  
soubor config/routes.rb, změna řádku: root :to => "home#index"



# Ruby - proměnné

- Naučte se Ruby -> Martin Šín -> <http://www.linuxexpres.cz/praxe/naucte-se-ruby>
- [https://www.tutorialspoint.com/execute\\_ruby\\_online.php](https://www.tutorialspoint.com/execute_ruby_online.php)

## **Ruby:**

```
print ["banán", "citron", "ananas"].sort.last.capitalize  
// vypíše Citron
```

## **PHP:**

```
$a = Array("banán", "citron", "ananas");  
sort($a);  
print ucwords( end($a) );  
// vypíše Citron
```

# Ruby - proměnné

```
mujText = 'Jsem nejlepší na světě.'
```

```
puts mujText
```

```
jmeno = 'Martin Šín'
```

```
puts 'Ty musíš být ' + jmeno
```

```
puts 'Říká se, že ' + jmeno + ' je ten nejlepší.'
```

```
prom = 3 * (2 + 6)
```

```
puts prom
```

```
var1 = 8
```

```
var2 = var1
```

```
puts var1
```

```
puts var2
```

```
var1 = 'osm'
```

```
puts var1
```

```
puts var2
```

# Ruby - práce s textem

```
puts 'Ahoj světě!'
```

```
puts 'Sklenku vína ' + 'si dám moc rád.' # Sklenku vína si dám moc rád.
```

```
puts 'Popis cesty je následující:'  
puts 'doleva ' * 5 + 'doprava ' * 2  
#doleva doleva doleva doleva doleva doprava doprava
```

```
puts 12 + 12    #24  
puts '12' + '12' #1212  
puts '12 + 12'  #12 + 12  
puts 2 * 5      #10  
puts '2' * 5    #22222  
puts '2 * 5'    #2 * 5
```

```
puts 'Zadej \'heslo: \' ' # Zadej ' heslo:'
```

```
puts '12' + 12  
#text6.rb:1:in `+': can't convert Fixnum into String (TypeError) from ...
```

```
komentář blok =begin ... =end
```

# Ruby - práce s textem

```
puts '12' + 12
```

```
#text6.rb:1:in `+': can't convert Fixnum into String (TypeError) from ...
```

```
puts '12'.to_i + 12      #24
```

```
puts '12' + 12.to_s     #1212
```

```
puts '12'.to_f          #12.0
```

```
puts '99.999'.to_f      #99.999
```

```
puts 20                  #put string
```

```
puts 20.to_s
```

```
puts '20'
```

```
puts 'Ahoj, jaké je tvoje jméno?'
```

```
jmeno = gets           #string + Enter
```

```
puts 'Ahoj, ' + jmeno + ' je pěkné jméno, rád tě poznávám!'
```

```
Ahoj, jaké je tvoje jméno?
```

```
Martin
```

```
Ahoj, Martin
```

```
je pěkné jméno, rád tě poznávám!
```

# Ruby - práce s textem

```
puts 'Ahoj, jaké je tvoje jméno?'  
jmeno = gets.chomp  
puts 'Ahoj, ' + jmeno + ' je pěkné jméno, rád tě poznávám!'
```

```
puts 'Ahoj, jaké je tvoje jméno?'  
jmeno = gets  
puts 'Ahoj, ' + jmeno.chomp + ' je pěkné jméno, rád tě poznávám!'
```

```
Ahoj, jaké je tvoje jméno?  
Martin  
Ahoj, Martin je pěkné jméno, rád tě poznávám!
```

```
var1 = 'radar'  
var2 = 'mississippi'  
puts var1.reverse      #radar  
puts var2.reverse      #ippississim
```

#nefunguje s diakritikou! - uložit jako UTF8!

# Ruby - práce s textem

```
slovo = 'ABBrakaDaBRa'  
puts slovo  
puts slovo.upcase           # ABBRAKADABRA  
puts slovo.downcase        # abbrakadabra  
puts slovo.swapcase        # abbRAKAdAbrA  
puts slovo.capitalize     # Abbrakadabra
```

```
sirka = 20                      #80  
text = '--> Text <--'  
puts text.ljust sirka          #--> Text <--  
#puts text.ljust(sirka)  
puts text.center sirka         #  --> Text <--  
puts text.rjust sirka          #    --> Text <--
```

# Ruby - matematické výrazy

```
puts 5**2          # 25
puts 36**0.5        # 6.0
puts 7/3            # 2
puts 7%3            # 1
```

```
puts ((2-5).abs)    #3
```

```
puts rand           # default 0 až 1
puts(rand(100))     # 0 až 99
```

objekt **Math**

```
puts(Math::PI)      # 3.14159265358979
puts(Math::E)       # 2.71828182845905
puts(Math.cos(Math::PI/3)) # 0.5
puts(Math.tan(Math::PI/4)) # 1.0
puts(Math.log(Math::E**2)) # 2.0
puts((1 + Math.sqrt(5))/2) # 1.61803398874989
```

**+= -= \*= /= %= \*\*=**

**&** (bit and) **|** (or) **^** (xor) **~** (not) **<<** (left shift) **>>** (right shift)



# Ruby - podmínka, cykl

```
puts 'Ahoj, jak se jmenuješ?'
jmeno = gets.chomp
puts 'Ahoj ' + jmeno + '!'
if jmeno == 'Josef'
    puts 'To je ale pěkné jméno!'
elsif jmeno == 'Jan'
    puts 'To je ale ještě dobré jméno!'
else
    puts 'Nechceš se nechat přejmenovat?'
end

== != > < >= <= and && or || not !

c = 2
case c
    when 1
        puts 'jedna'
    when 2
        puts 'dva'
    else
        puts 'nevím'
end
```

# Ruby - cykl, pole

```
while 1
  puts 'Nejhorší programátor!'
end #Ctrl + c
```

until = while !(výraz)

```
for i in 0..10
  puts i
end
```

0..10 -> 0 až 10 krok 1; **10.step(0, -2)** -> 10 až 0 krok -2

```
jmena = ['Martin', 'Honza', 'Alena']
puts jmena[0] #Martin
```

výpis pole:

```
jmena.each do |jmeno|
  puts jmeno
end
```

# Ruby - pole

```
jmena = ['Martin', 'Honza', 'Alena']  
puts jmena.join(' ', ' ') # Martin, Honza, Alena
```

```
napoje = ['mléko', 'voda', 'pivo']  
puts napoje.join(' :-( ' ) + ' :-( ') # mléko :-( voda :-( pivo :-(
```

přidání položky

```
napoje.push 'víno'  
puts napoje[3] # víno
```

odebrání poslední položky

```
napoje.pop  
napoje.pop
```

zobrazení poslední položky

```
puts napoje.last #voda
```

zobrazení počtu položek v poli

```
puts napoje.length # 2
```

# Ruby - metody, procedury, funkce

```
def pozdrav  
  puts 'Ahoj!'
```

```
end
```

```
pozdrav  
pozdrav
```

parametry

```
def mocnina cislo  
  vysledek = cislo * cislo  
end  
puts mocnina(10)
```

```
def mocnina cislo  
  vysledek = cislo * cislo  
  vysledek1 = 5  
end
```

```
def mocnina cislo  
  vysledek = cislo * cislo  
  vysledek1 = 5  
  vysledek  
end
```

# Ruby - metody, procedury, funkce

```
def pronasob (cislo1, cislo2)
```

```
  vysledek = cislo1 * cislo2
```

```
end
```

```
puts pronasob(10, 5)
```

```
def test
```

```
  i = 100
```

```
  j = 200
```

```
  k = 300
```

```
return i, j, k
```

```
end
```

```
var = test
```

```
puts var.length
```

```
puts var
```

```
předání parametrů
```

```
def procPar (*par)
```

```
  for i in 0...par.length
```

```
    puts par[i]
```

```
  end
```

```
end
```

```
procPar "Text", 5, 12.7
```

# Ruby - třídy

**new** -> vytvoření nového objektu dané třídy

```
a = Array.new + [17] + [20]
```

```
b = String.new + 'Ahoj'
```

```
c = Time.new
```

```
puts 'a = ' + a[0].to_s + ' ' + a[1].to_s
```

```
puts 'b = ' + b.to_s
```

```
puts 'c = ' + c.to_s
```

```
# a = 17 20
```

```
# b = Ahoj
```

```
# c = 2014-04-10 12:29:21 +0200
```

# Ruby - třídy

**hash** -> neuspořádaná skupina prvků

poleBarev = [] # to samé jako metoda Array.new

hashBarev = {} # to samé jako metoda Hash.new

poleBarev[0] = 'červeně'

poleBarev[1] = 'zeleně'

poleBarev[2] = 'modře'

hashBarev['text'] = 'červeně'

hashBarev['čísla'] = 'zeleně'

hashBarev['klíčová slova'] = 'modře'

poleBarev.each do |barva|

puts barva

end

puts

hashBarev.each do |vyuziti, barva|

puts vyuziti + ': ' + barva

end

každý prvek označen symbolem -> text, znak, číslice

# Ruby - třídy

## rozšíření třídy

```
class Integer
```

```
  def to_czech  
    if self == 5  
      cesky = 'pet'  
    else  
      cesky = 'padesatpet'  
    end  
  
    cesky  
  end  
  
end
```

```
puts 5.to_czech      # pet  
puts 55.to_czech    # padesatpet
```



# Ruby - třídy

## vytvoření třídy

```
class Kostka
```

```
  def hod  
    1 + rand(6)  
  end
```

```
end
```

```
hody_kostkou = [Kostka.new, Kostka.new, Kostka.new]
```

```
hody_kostkou.each do |kostka|  
  puts kostka.hod  
end
```

# Ruby - třídy

proměnné instancí

instance -> konkrétní objekt dané třídy

```
class Kostka
  def hod
    @cislo = 1 + rand(5)
  end

  def vysledek
    @cislo
  end
end
```

```
kostka = Kostka.new
```

```
kostka.hod
puts kostka.vysledek
```

```
kostka.hod
puts kostka.vysledek
```

# Ruby - třídy

proměnné instancí

```
class Kostka
```

```
  def initialize          #metoda - inicializace
    hod
  end
```

```
  def hod
    @cislo = 1 + rand(5)
  end
```

```
  def vysledek
    @cislo
  end
```

```
end
```

```
puts Kostka.new.vysledek
puts Kostka.new.vysledek
```

# Ruby - třídy

proměnné instancí

```
class Kostka
  def initialize
    hod
  end
```

```
  def vysledek
    @cislo
  end
```

**private**

```
  def hod
    @cislo = 1 + rand(5)
  end
end
```

```
kostka = Kostka.new.vysledek
puts kostka
kostka.hod      #Error
```