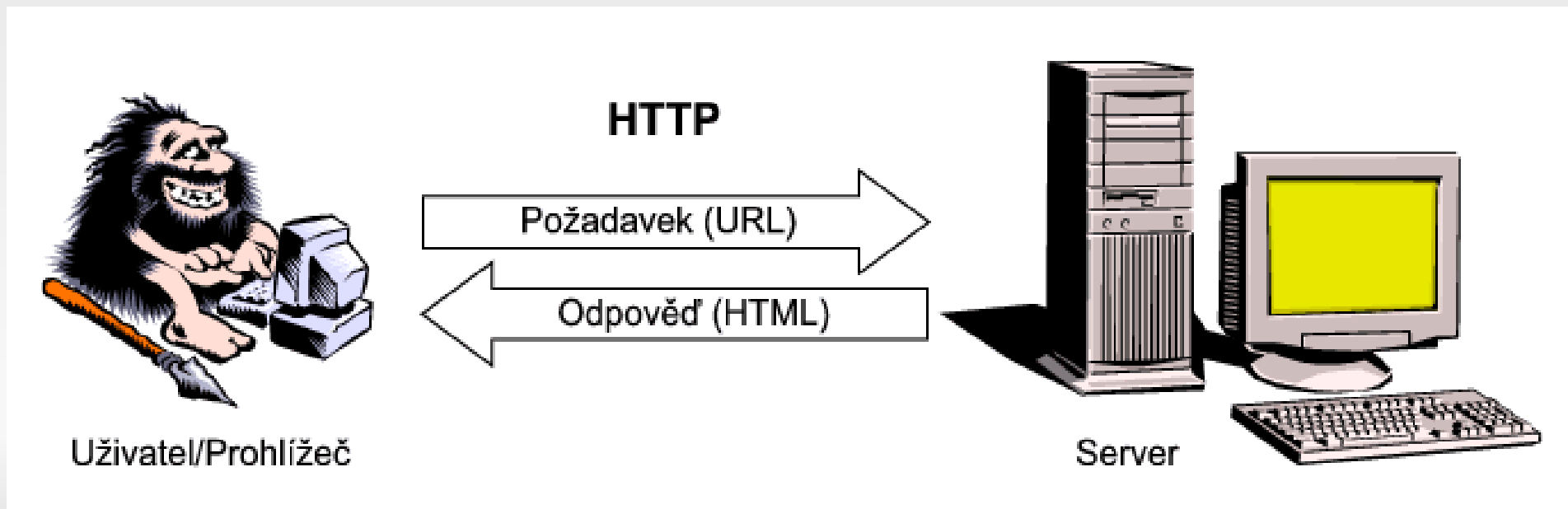


# Ruby on Rails



- framework
- vývoj aplikací pro WWW
- MVC - Model - View - Controller
- jazyk Ruby





- interpretovaný jazyk
  - dynamický, skriptovací
  - podobně jako PHP, Perl, Python
- objektově orientovaný
  - jednoduchá dědičnost
  - vše je objekt, včetně čísel (jako Smalltalk, Ada)
- závislost na konvencích
  - např. pojmenování prvků v modelu ovlivňuje pojmenování prvků ve view

# Hello world



```
#!/usr/bin/env ruby
```

```
puts "Hello world"
```

```
$ chmod a+x helloWorld.rb
```

```
$ helloWorld.rb #nebo ruby helloWorld.rb
```

```
Hello world
```

```
$
```

# Základy Ruby



- Všechno je objekt
- Proměnné nejsou typované - neuvádí se datový typ
- Automatická alokace paměti
- Automatické odstraňování objektů z paměti - Garbage collector
- Komentáře s využitím #
- \n ukončuje statement

# Základy Ruby 2



- Proměnná  
teplota = 35

- Metoda  
def název

...

end

# Základy Ruby 3



- Řetězce  
veta = "Kolik je hodin"
- Zřetězení  
"Je " + 5.to\_s + " hodin"
- Další možné varianty
  - např. String(5)
  - nebo "Je #{hodina} hodin"

# Pole



```
pole = ["Toto", "je", "pole", 1, 2, 3]
pole2 = Array.new
puts pole2.length
pole2[0] = "Karel"
```

# Asociované pole



```
pizza = ["prvni" => "syr", "druhy" => "klobasa"]  
puts pizza["druhy"]
```



# Rozsahy



- `prvky = 1..4`
- `puts prvky.to_s`
- 
- `prvky2 = "alpha".."alphe"`

# Řídící struktury



- if
- elseif
- else
- end
- 
- unless (inverze if)
- case
- ternární operátor

# Operátory



- ==
- !=
- ||
- &&
- ...

# While



```
while ($_ != "q")  
  puts "Makame ..."  
  print "Ukoncete zadanim q:"  
  gets  
  chomp  
end
```

# Until



- `until ($_ == "q")`

# For



```
for hodnota in 1..10  
  puts hodnota.to_s  
end
```

```
for polozka in ["a1", "b1", "c555"]  
  puts polozka  
end
```

# Iterátory



upto  
step

times  
5.times do  
  puts "Vypis"  
end

each

# Metody



```
def vypisovac(a, b, c)
  puts a + " " + b + " " + c
end
```



# Metody



```
def vypisovac(a, b, c)
  puts a + " " + b + " " + c
  return 1
end
```

```
def vypisovac(a, b, c)
  a + b + c #vraci soucet - neni nutny return
end
```

# Metody



```
def vypisovac(a, b, c, *dalsi)
  puts a + " " + b + " " + c + " " + dalsi.join(" ")
  return 1
end
```

```
datum = Time.now
vypisovac("A", "B", "C", "kaer", "dsf", "dfs", 1,
datum)
```

# Použité zdroje



- Min-Yen Kan. Introduction to Ruby, WING Group Meeting, 9 Jun 2006
- <http://www.jakpsatweb.cz/katalog/hosting-ror.html>