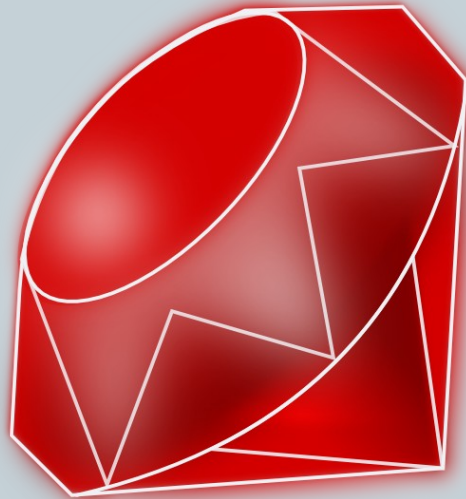


# Ruby



# O que é ruby?



**Ruby** é uma linguagem de programação:

- Interpretada.
- Dinamica e fracamente tipada(duck typing).
- Orientada a objetos.

# Primeiros passos



- Como checar se a máquina tem o Ruby instalado?
  - `ruby -v`
- Como usar um terminal ruby?
  - `irb`
- Como executar um arquivo de script ruby?
  - `ruby olamundo.rb`

# Strings



- Para definir uma string:
  - "Hello World"
  - 'Hello World'
- Concatenar strings
  - "hello" + "world"

# Números



- Operadores aritméticos

- Soma: +
- Subtração: -
- Multiplicação: \*
- Divisão inteira: /
- Resto da divisão: %
- Potência: \*\*
- Incremento: +=
- Decremento: -=

# Operadores relacionais



- Maior que:  $>$
- Menor que:  $<$
- Igual a:  $==$
- Diferente de:  $!=$
- Maior ou igual a  $\geq$
- Menor ou igual a  $\leq$

# Operadores lógicos



- **E:** & ou &&
  - Ex.: (5>3 && 1<2)
- **Ou:** | ou ||
  - Ex.: (6>2 || 4>3)
- **Não:** !
  - !(a==2)

# Variáveis



- Variáveis locais:
  - Começam com algum caractere entre 'A' a 'Z' ou 'a' a 'z'.
  - Só podem ser chamadas se forem instanciadas.
- Variáveis de instância:
  - Começam com o caractere @.
  - Geralmente são usado dentro de classes, métodos ou módulos.



# Estruturas condicionais



- Se:

```
if (< condição> )  
    < com ando>  
end
```

- Se..Senão

```
if (< condição> )  
    < com andos para condição verdadeira>  
else  
    < com andos para condição falsa>  
end
```

# Estruturas condicionais



- Se... senão se  
**if** (< condição> )  
    < com andos se a condição é verdade>  
**elseif** (< segunda condição> )  
    < com andos se a segunda condição é verdadeira>  
**end**

# Arrays



- Criar um array

```
nom es = Array.new
```

- Adicionar elementos na Array

```
nom es << "Renato"
```

```
nom es << "Palloma"
```

```
nom es << "Pablo"
```

- Criar um array com elementos

```
nom es = ["Renato", "Palloma", "Pablo"]
```

# Arrays



- Percorrer um a array

```
nom es.each do |nom e|  
  puts nom e  
end
```

Ou

```
nom es.each { |nom e| puts nom e }
```

# Hash



- Hash é uma estrutura de chave e valor

- Criar uma hash:

```
user = Hash.new
```

- Atribuir valores a chave:

```
user[:nome] = "Renato"
```

```
user[:email] = "renato@jalencar.com"
```

- Criar uma hash com valores:

```
user = Hash.new(:nome => "Renato",  
:email => "renato@jalencar.com")
```

# Métodos



- Definindo um método

```
def hello_world  
  puts "hello world"  
end
```

- Chamando um método

```
hello_world
```

- Método com retorno

```
def calcula_soma(num 1, num 2)  
  num 1 + num 2  
end
```

- Chamando um método com retorno

```
soma = calcula_soma(1,2)
```

# Classes



- Definir uma classe

# arquivo processo.rb

```
class Processo  
end
```

- Instanciar objeto de uma classe

# arquivo criar\_processo.rb

```
processo = Processo.new
```

# Herança



```
# pessoa.rb  
class Pessoa  
end
```

```
# pessoa_fisica.rb  
class PessoaFisica < Pessoa  
end
```



# Referências



- Guia do usuário Ruby, Ruby Brasil.  
<http://ruby-br.org/tutoriais/?t=1&p=0>,  
acessado em 25, jun de 2014.