

# Ruby & Rake

# Ruby - Revisão de Sintaxe

# String

```
"Hello, World!"
```

# Interpolação de Strings

```
name = "Rodrigo"  
"Hello, #{name}"
```

# Funções

```
def greet(name)
  return "Hello, #{name}"
end
```

```
def greet name
  "Hello, #{name}"
end
```

# Símbolos

`:name`, `:anything`

Símbolos são mais eficientes que strings. Duas strings com o mesmo conteúdo são dois objetos diferentes, mas para qualquer nome existe apenas um objeto Symbol. Isso pode economizar tempo e memória.<sup>1</sup>

---

<sup>1</sup>[http://guru-sp.github.io/tutorial\\_ruby/simbolos.html](http://guru-sp.github.io/tutorial_ruby/simbolos.html)

# Hash

```
hash = {  
  :name => "Rodrigo Chaves",  
  :age => 20,  
  :birthday => "12/04/1996"  
}  
  
hash[:name] == "Rodrigo Chaves"
```

# Rake

Rake é Ruby Make, é um *standalone* feito em ruby utilizado para substituir o utilitário do unix `make`. Ele usa `rakefiles`, que são arquivos feitos com uma lista de tarefas. Um exemplo clássico é a utilização do `rake` para fazer atividades administrativas.



## Instalando o Rake

```
$ gem install rake
```

## Criando meu arquivo rake

`rakefile, Rakefile, rakefile.rb, Rakefile.rb`

# Tasks

```
task :name  
end
```

Cada task é responsável para fazer uma tarefa.

# Dependências

```
task :name => [:dep1, :dep2]  
end
```

# Argumentos

```
task :t, [args] => [deps]
```

# Exemplos

```
$ rake -T
(in /home/.../rake)
rake clean           # Remove any temporary products.
rake clobber         # Remove any generated file.
rake clobber_rdoc    # Remove rdoc products
rake contrib_test    # Run tests for contrib_test
rake default         # Default Task
rake install         # Install the application
rake lines           # Count lines in the main rake file
rake rdoc            # Build the rdoc HTML Files
rake rerdoc          # Force a rebuild of the RDOC files
rake test            # Run tests
rake testall         # Run all test targets
```

# Exercícios

1. Crie uma tarefa rake que compila um programa C.
2. Crie uma tarefa rake que compila 2 programas C e gera o executável.
3. Simular duas tarefas de Rails relacionadas ao banco de dados, em um banco de dados.
  - ▶ 1. criar o arquivo .txt
  - ▶ 2. preencher o arquivo .txt com dados de exemplo
4. Criar um modulo de ruby e criar um tarefa rake para carregá-lo no interpretador ruby.

# Referências

- ▶ <http://www.stuartellis.name/articles/rake/>
- ▶ [https://en.wikipedia.org/wiki/Rake\\_%28software%29](https://en.wikipedia.org/wiki/Rake_%28software%29)
- ▶ [http://guides.rubyonrails.org/command\\_line.html#rake](http://guides.rubyonrails.org/command_line.html#rake)
- ▶ <http://lukaszwrobel.pl/blog/rake-tutorial>