

---

# Introducing Ruby



---

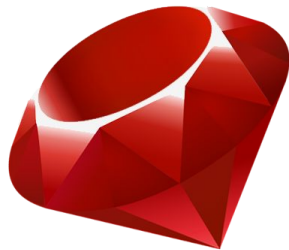
## First Session

---

# Agenda

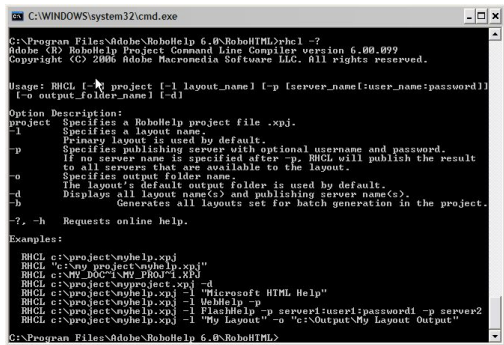
---

- Command line
- Numbers
- Strings
- Exercises



---

# Command Line



```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\Adobe\RoboHelp 6.0\RoboHTML>rhcl -?
Adobe (R) RoboHelp Project Command Line Compiler version 6.00.099
Copyright (C) 2006 Adobe Macromedia Software LLC. All rights reserved.

Usage: RHCL [-s] project [-l layout_name] [-p [server_name[:user_name:password]]]
           [-o output_folder_name] [-d]

Option Description:
-s project Specifies a RoboHelp project file .xpj.
-l Specifies a layout name.
  Primary layout is used by default.
-p Specifies publishing server with optional username and password.
  If no server name is specified after -p, RHCL will publish the result
  to all servers that are available to the layout.
-o Specifies output folder name.
  The layout's default output folder is used by default.
-d Displays all layout name(s) and publishing server name(s).
-b Generates all layouts set for batch generation in the project.
-y, -h Requests online help.

Examples:
RHCL c:\project\myhelp.xpj
RHCL "c:\my project\myhelp.xpj"
RHCL c:\my doc\myproj1.xpj -l "Microsoft HTML Help"
RHCL c:\project\myhelp.xpj -l "Microsoft HTML Help"
RHCL c:\project\myhelp.xpj -l WebHelp -p
RHCL c:\project\myhelp.xpj -l FileHelp -p server1:user1:password -p server2
RHCL c:\project\myhelp.xpj -l "My Layout" -o "c:\Output\My Layout Output"
```

---

## Introduction

---

# What is the command line?

---

- Textual interaction with the computer
  - Also called “Terminal”, “Shell”, “bash”, “console”, “PowerShell”
  - Can look differently depending on operating system or user settings
-

# How does it work?

---

1. Computer “asks” for command
  2. You input a command, followed by the return key
  3. Computer executes the command
  4. Computer prints the output of the command
-

# Command line

---

- Always in a **working directory**
  - You can do most things you can do in a graphical interface
  - You can do a lot of things quicker than with a graphical interface (once you've practiced a bit)
  - Command followed by arguments (separated by spaces)
  - Cryptic commands might seem scary at first
-

# Directories

---

- Where am I?
  - `pwd`
  - “Print working directory”
- How do I go somewhere else?
  - `cd`
  - “Change directory”
  - Examples
    - `cd testdirectory`
    - `cd ..`
    - `cd ~`

# Directories

---

- How do I create directories?
    - `mkdir`
    - “make directory”
    - e.g. `mkdir rubymonstas`
  - How do I delete directories?
    - `rmdir`
    - “remove directory”
    - e.g. `rmdir rubymonstas`
-



# Directories and files

---

- How do I see what's inside a directory?
  - `ls`
  - “list”
  - Examples
    - `ls`
    - `ls rubymonstas`
    - `ls ..`
    - `ls ~`

# Ruby-specific commands

---

- How do I run a Ruby script?
    - `ruby`
    - e.g. `ruby myscript.rb`
  - How do I try something or play around with Ruby?
    - `irb`
    - “interactive ruby shell”
    - Command line for Ruby
-

---

# Numbers



---

## Introduction

---

# Numbers

---

- Integers: whole numbers  
→ 1, 2, 3...
- Floats: decimal numbers  
→ 1.5, 3.09, 4.12...



# Calculations

---

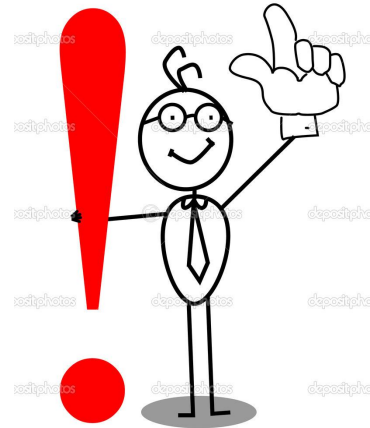
- Addition  $\rightarrow +$
- Subtraction  $\rightarrow -$
- Multiplication  $\rightarrow *$
- Division  $\rightarrow /$
- Modulo  $\rightarrow \%$



# Division - Attention!

---

- Division of Integers
- Division of Floats
- $6 / 4 = ?$
- $6 / 4.0 = ?$



# Modulo

---

- Gives the remainder of a division
  - $13 \% 6 = 1$
  - $21 \% 8 = 5$
- Why the hell...?



---

# Strings



---

## Introduction

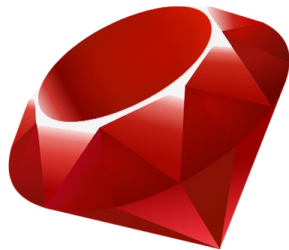
---



# String Definition

---

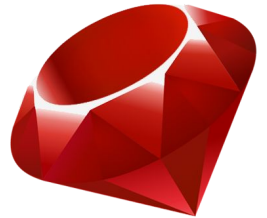
- Words or phrases
- Single quotes ‘ . . . ’
- Double quotes “ . . . ”



# Methods

---

- “Talk to” an object
- Manipulate an object
- Objects and methods are separated by a .



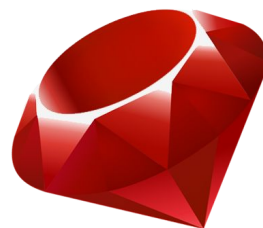
# String Methods

---

- `.length`

```
> "I love Ruby".length  
=> 11
```

→ includes white spaces

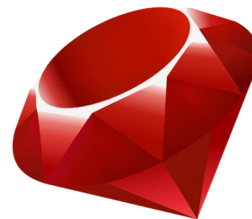


---

- `.reverse`

```
> "I love Ruby".reverse
```

```
=> "ybuR evol I"
```



---

- `.include?`

```
> "I love Ruby".include?("Ruby")  
=> true
```

```
> "I love Ruby".include?("Rails")  
=> false
```

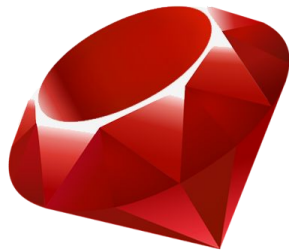


---

- `.upcase`

```
> "Ruby".upcase
```

```
=> "RUBY"
```



---

- `.downcase`

```
> "Ruby".downcase
```

```
=> "ruby"
```

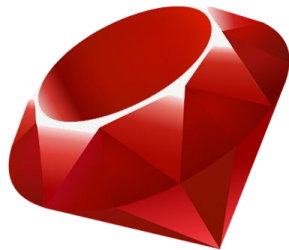


---

- `.capitalize`

```
> "ruby".capitalize
```

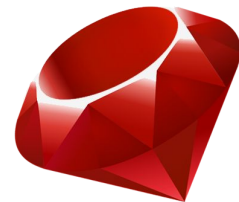
```
=> "Ruby"
```





---

# Ruby



# Documentation

---

Everything is online!

---

# Ruby documentation

---

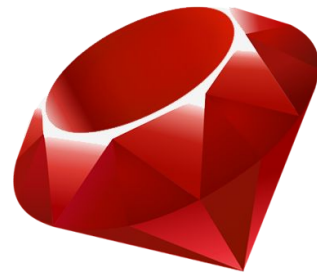
- Google!
- For example “ruby string reverse”

<http://ruby-doc.org/core/String.html>

---

---

# Time to practice



---

Let's get to it!

---