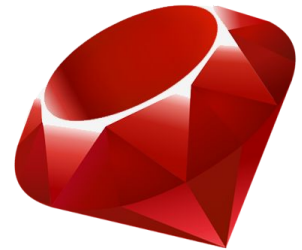

Ruby Monstas



Session 2

Agenda

- Recap
- Advanced Command Line
- Variables
- Booleans
- Arrays
- Exercises



Recap

```
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 [-?] project [-l layout_name] [-p [server_name[:user_name:password]]
[-o output_folder_name] [-d]

Option Description:
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.

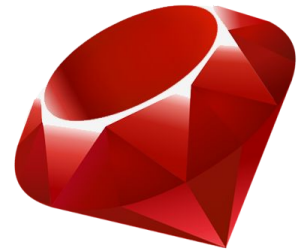
-?, -h Requests online help.

Examples:
RHCL c:\project\myhelp.xpj
RHCL "c:\my project\myhelp.xpj"
RHCL c:\MY_DOC~1\MY_PROJ~1.XPJ
RHCL c:\project\myproject.xpj -d
RHCL c:\project\myhelp.xpj -l "Microsoft HTML Help"
RHCL c:\project\myhelp.xpj -l WebHelp -p
RHCL c:\project\myhelp.xpj -l FlashHelp -p server1:user1:password1 -p server2
RHCL c:\project\myhelp.xpj -l "My Layout" -o "c:\Output\My Layout Output"

C:\Program Files\Adobe\RoboHelp 6.0\RoboHTML>
```

Numbers, Calculations, Strings, Methods

- 1,2,3
- 1.5, 3.09, 4.12
- 1+2
- 1/2
- “Lorem Ipsum”
- ‘Im a sentence’.length



Advanced Command Line

Copy files and directories

cp	Copy files and folders
mv	Move files and folders, also rename files and folders
rm	Remove files
touch	Create a file if it does not exist already
man	Manual pages, documentation for commands
cat	Concatenate and print files
less	Display file

Variables

Introduction

Definition

- Memory locations
- Can hold any data

```
irb(main):001:0> a = 5
```

```
=> 5
```

```
irb(main):002:0> b = "Hello World"
```

```
=> "Hello World"
```

Naming Conventions

- Small letters
 - No white space
 - If the variable name has several words, use underscore
 - Cannot start with a number
-

Good variable names

- `user_name`
 - `date_of_birth`
 - `list_of_favourite_drinks`
-

Bad/invalid variable names

- 5_things
 - mynewvariable
 - AnotherVariableName
-

Assigning Variables

- Assignment operator: =
- Assigning variables: assign an object to a variable

```
irb(main) :001:0> a = 5
```

```
=> 5
```

Use of Variables

- Refer to objects via variables

```
irb(main):003:0> a + 4
```

```
=> 9
```

```
irb(main):004:0> b.length
```

```
=> 11
```

Overwrite Variables

- If you assign a new object to the same variable, the new object is valid

```
irb(main):002:0> b = "Hello World"
```

```
=> "Hello World"
```

```
irb(main):005:0> b = "Another String"
```

```
=> "Another String"
```

Assigning Variables to Variables

- Variables can be assigned to other variables

```
irb(main) :001:0> a = 5
```

```
=> 5
```

```
irb(main) :006:0> x = a
```

```
=> 5
```

puts

- “puts” is used to display the results of Ruby code

```
irb(main):008:0> puts "We print a string"
```

```
We print a string
```

```
=> nil
```

```
irb(main):009:0> puts b
```

```
Another String
```

```
=> nil
```

gets

- “gets” is used to get user input

```
irb(main):010:0> name = gets
```

```
Marion
```

```
=> "Marion\n"
```

```
irb(main):011:0> name
```

```
=> "Marion\n"
```

Booleans

Introduction

Boolean values

- The truth!
 - true
 - false

Wait, I've seen that before!

- Functions (methods) can return booleans

```
irb(main):013:0> "Ruby is great".include? "Ruby"
```

```
=> true
```

```
irb(main):014:0> "This is great".include? "Ruby"
```

```
=> false
```

```
irb(main):016:0> 1.odd?
```

```
=> true
```

```
irb(main):017:0> 3.even?
```

```
=> false
```



-
- Ruby code convention: Functions that return a boolean value end with a ?

```
irb(main):023:0> b
```

```
=> "Another String"
```

```
irb(main):024:0> b.empty?
```

```
=> false
```

```
irb(main):026:0> x
```

```
=> 5
```

```
irb(main):027:0> x.integer?
```

```
=> true
```

Booleans and variables

```
irb(main):029:0> is_b_empty = b.empty?
```

```
=> false
```

```
irb(main):030:0> is_b_empty
```

```
=> false
```

Comparison operators

==	Are the left and right side equal ?
!=	Are the left and right side not equal ?
<	Is the left side smaller than the right side?
>	Is the left side bigger than the right side?
<=	Is the left side smaller than or equal to the right side?
>=	Is the left side bigger than or equal to the right side?

Comparison operators

```
irb(main):074:0> 5 == 4
```

```
=> false
```

```
irb(main):075:0> 5 != 4
```

```
=> true
```

```
irb(main):076:0> 1 < 2
```

```
=> true
```

```
irb(main):077:0> 1 > 2
```

```
=> false
```

```
irb(main):078:0> 7.1 <= 7.2
```

```
=> true
```

```
irb(main):079:0> 7.1 >= 7.2
```

```
=> false
```

Arrays

Introduction

Arrays

- lists of things

```
irb(main):036:0> [1, 1, 3, 5, 8]
```

```
=> [1, 1, 3, 5, 8]
```

```
irb(main):038:0> my_array = [7, "hello", 5.9]
```

```
=> [7, "hello", 5.9]
```

Accessing array items

```
irb(main):043:0> my_array = [7, "hello", 5.9]
```

```
=> [7, "hello", 5.9]
```

```
irb(main):044:0> my_array[1]
```

```
=> "hello"
```

```
irb(main):045:0> my_array.last
```

```
=> 5.9
```

```
irb(main):046:0> my_array.first
```

```
=> 7
```

Adding to an array

- “shovel” operator

```
irb(main):039:0> my_array << "new item"
```

```
=> [7, "hello", 5.9, "new item"]
```

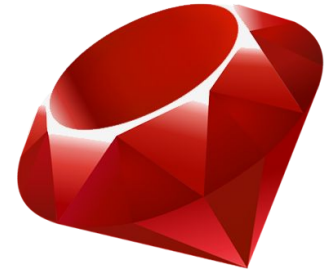
```
irb(main):041:0> my_array << x
```

```
=> [7, "hello", 5.9, "new item", 5]
```

Deleting from an array

```
irb(main):055:0> my_array
=> [7, "hello", 5.9, "new item", 5]
irb(main):056:0> my_array.delete_at(2)
=> 5.9
irb(main):057:0> my_array
=> [7, "hello", "new item", 5]
irb(main):058:0> my_array.delete("hello")
=> "hello"
irb(main):059:0> my_array
=> [7, "new item", 5]
```

Time to practice



Let's get to it!
