Cheatsheets / **Learn Ruby**

Introduction to Ruby

Ruby string interpolation

In Ruby, *string interpolation* is used to insert the result of Ruby code into a string.

```
age = 30

print "Hi, my name is Cody, and I am #{age} years old"
# "Hi, my name is Cody, and I am 30 years old"
```

Get user input in Ruby

In Ruby, we can get the user's input using `gets.chomp`. `gets` is the method that is used to retrieve user input. Ruby automatically adds a new line after each bit of input, so `chomp` is used to remove that extra line.

```
print "Type your name and press Enter: "
name = gets.chomp

puts "My name is #{name}!"
```

Ruby Variables

In Ruby, a variable is a place to store values of almost any type including Integer, Boolean, String, Array, and Hashes. Each variable has its own name which cannot begin with a capital letter or a number and we use the equal sign for assigning a value to that variable.
The variable declaration does not require that you mention a specific data type.
The following program declares `my_var` variable and assigns the value `48`.

```
my_var = 48
```

Put and print command

`put` and `print` commands can be used to display text in the console.

```
print "Hello"
puts "This is written in a new line"
print "Still printing"
```

Code comments in Ruby

Commenting code helps programmers write free text that is commonly used to explain the code written, or can even be used to add TODO's to the code. There are two types of comments that can be written in Ruby:

- **Single line comments** start with a `#`.
- **Multi line comments** start with `=begin` and end with `=end`.

```
# I am a single line comment.

=begin
I am a multi line comment.
I can take as many lines as needed.
=end
```

Numeric data types in Ruby

In Ruby, the *Numeric* data type represents numbers including **integers** and **floats**.

```
# Integer value
x = 1

# Float value
y = 1.2
```

Arithmetic operations in Ruby

In Ruby, we can use arithmetic operators to evaluate mathematical expressions. The most common Ruby arithmetic operators are addition (+), subtraction (-), division(/), multiplication(*), exponentiation(**) and modulo(%).

```
print 1+3
# Addition: output 4

print 1-2
# Subtraction: output -1

print 9/3
# Division: output 3

print 2*3
# Multiplication: output 6

print 2**3
# Exponentiation: output 8

print 16%9
# Modulo: output 7
```

Ruby Object Methods

In Ruby, methods are built-in abilities of objects. To access an object's methods, you need to call it using a `.` and the method name.

```
var = "codecademy"

# Method to get the length of a string
print var.length # 10

# Method to get the string reversed
print var.reverse # ymedacedoc

# Method to convert all letters to
uppercase
print var.upcase # CODECADEMY
```

Strings in Ruby

Strings in Ruby are a sequence of characters enclosed by single quotation marks (‘’) or double quotation marks (“”).

```
# String 1
s1 = 'I am a single string!'

# String 2
s2 = "I am a double string!"
```

Boolean Data Types in Ruby

In Ruby, in order to represent values of truth about specific statements, we use Boolean variables. Boolean variables values are either `true` or `false`.

```
# Boolean true variable
year2019 = true

# Boolean false variable
year2018 = false
```

Ruby `.upcase` and `.downcase` Methods

Ruby strings have an `.upcase` and a `.downcase` method used to change their case. `.upcase` returns a version of the string in all uppercase, and `.downcase` returns a version in all lowercase.

```
puts "codecademy".upcase
# CODECADEMY

puts "Codecademy".downcase
# codecademy
```

Print Share