
Ruby Monstas



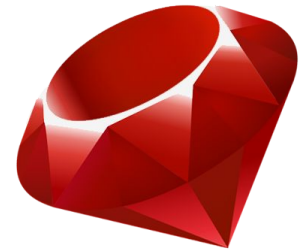
Session 5

Agenda

Starting Time

Control structures

Exercises



Starting Time

- 18.30?
- 19.00?

Control structures

Introduction

If

Sometimes we want to execute code depending on the situation.

If a certain situation is given, we want to execute a body of code.

The condition

```
if condition
  # the condition is true
end
```

condition is a **boolean**.

What could be a condition?

`1 < 2`

`"Hello" == "World"`

`"Hello".include?("llo")`

`[4, 5, 6].include?(7)`

`{"Marion" => 7}.has_key?("Marion")`

Anything that returns `true` or `false`!

If: example

```
irb(main):001:0> number = 42
```

```
=> 42
```

```
irb(main):002:0> if number > 10
```

```
irb(main):003:1>   puts "#{number} is greater than 10"
```

```
irb(main):004:1> end
```

```
42 is greater than 10
```

```
=> nil
```

If / else: the pattern

```
if condition
  # the condition is true
else
  # the condition is false
end
```

If, else: example

```
irb(main):001:0> number = 4
```

```
=> 4
```

```
irb(main):002:0> if number > 10
```

```
irb(main):003:1>   puts "#{number} is greater than 10"
```

```
irb(main):004:1> else
```

```
irb(main):005:1>   puts "#{number} is lower or equal to 10"
```

```
irb(main):007:1> end
```

```
4 is lower or equal to 10
```

```
=> nil
```

If / else: the pattern

```
if condition1
    # condition1 is true
elsif condition2
    # condition2 is true
else
    # none of the conditions are true
end
```

elsif

You can use as many elsif's as you like

If, elsif, else: example

```
irb(main):001:0> season = "spring"
=> "spring"
irb(main):002:0> if season == "summer"
irb(main):003:1>   puts "It's too hot!"
irb(main):004:1> elsif season == "winter"
irb(main):005:1>   puts "It's too cold!"
irb(main):006:1> else
irb(main):007:1>   puts "It's just right!"
irb(main):008:1> end
It's just right!
=> nil
```

Each: the pattern

```
array = []  
array.each do |element|  
  # do stuff with element  
end
```

Each: example

```
irb(main):001:0> names = ["Alice", "Bob", "Eve"]  
=> ["Alice", "Bob", "Eve"]  
irb(main):002:0> names.each do |name|  
irb(main):003:1>   puts name  
irb(main):004:1> end  
Alice  
Bob  
Eve  
=> ["Alice", "Bob", "Eve"]
```

Each: another example

```
irb(main):031:0> array = [1, 2, 3]
=> [1, 2, 3]
irb(main):032:0> array.each do |n|
irb(main):033:1*       n += 1
irb(main):034:1>   puts n
irb(main):035:1> end
2
3
4
=> [1, 2, 3]
```

While: the pattern

```
while condition do  
  # condition is true  
end
```

While: use

The while loop repeats a task until an expression returns false.

While: example

```
i = 0
```

```
while i < 5 do
```

```
    puts i
```

```
    i += 1
```

```
end
```

Self Study

- Research the “until”-Loop
 - Practise
 - Ask questions
-

Don't worry

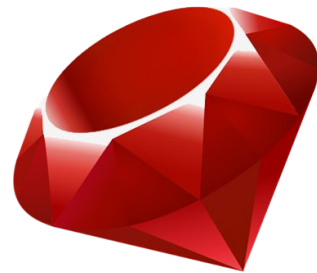
- Control structures take time to get used to
 - Don't be too hard on yourself
 - Keep practising
 - Ask for help
-

KEEP
Smiling

YOU ARE
Amazing

Tip: Work in pairs

Time to practice



Let's get to it!
