

---

# Ruby Monstas



---

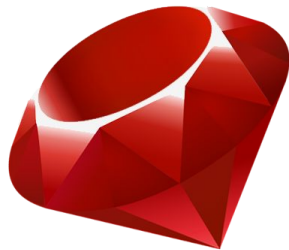
## Session 4

---

# Agenda

---

Interactive Recap  
Control structures  
Exercises





**Scott Hanselman**  @shanselman



Non-technical people, here's a secret. We tech folks have no idea what the problem is. We just try to narrow it down, removing variables.



13h

*“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.” - Linus Torvalds*

*“There are two ways to write error-free programs; only the third one works.” - Alan J. Perlis*

---

# 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
```

# 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
```

# Loop

---

A **loop** cycles through code until we tell it to stop with **break**.

---

# Loop: the pattern

---

```
loop do
  if condition
    break
  end
end
```

---

# Loop: Example

---

```
irb(main):001:0> counter = 1
=> 1
irb(main):002:0> loop do
irb(main):003:1>   puts counter
irb(main):004:1>   counter = counter + 1
irb(main):005:1>   if counter > 3
irb(main):006:1>     break
irb(main):007:1>   end
irb(main):008:1> end
1
2
3
=> 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"]
```

---

# Demo

---

Todo Script

---

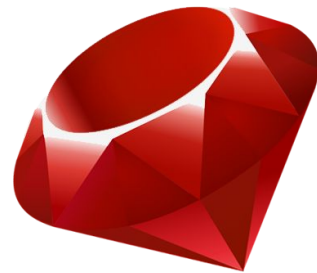
**Tip: Work in pairs**

---

---

---

# Time to practice



---

Let's get to it!

---