progress engine
we will develop it

#tceh

2016

Врубиться в Ruby

Лекция 3

Лекция 3 - базовые знания

- Условные операторы
- Циклы и управление ходом цикла
- Что такое ООП
- Классы, модули, методы
- Области видимости
- Исключения

Условные операторы

Операторы if/else/elsif

```
if firstname == "John" && lastname == “Snow" then
    print "Hello Lord Commander!"
end
```

Операторы if/else/elsif

```
if firstname == "John" && lastname == "Snow"
    puts "Hello Lord Commander!”
else
    puts “Tell me your name, stranger!”
end
```

Операторы if/else/elsif

```
if firstname == "John" && lastname == "Snow"
   puts "Welcome Lord Commander!”
elsif firstname == “Tyrion” && lastname == “Lannister”
   puts “What are you doing here?”
else
   puts “Tell me your name, stranger!”
end
```

Тернарный оператор

customerName **==** "Fred" **?** "Hello Fred" **:** "Who are you?"

[condition] ? [true expression] : [false expression]

unless

```
if i < 10
   puts "Student failed"
else
   puts "Student passed"
end
```

```
unless i >= 10
    puts "Student failed"
else
    puts "Student passed"
end
```

Циклы

Циклы **for, times, upto, downto**

```
for i in 1..8 do
    puts i
end
```

Циклы **for, times, upto, downto**

5.**times** { |i| puts i }

Циклы **for, times, upto, downto**

```
(0..5).each do |i|
   puts "Value of local variable is #{i}"
end
```

Циклы **for, times, upto, downto**

```
1.upto(5) { |i| puts i }
```

Циклы **while, until**

```
i = 0
while i < 5 do
   puts i
   i += 1
end
```

Циклы **while, until**

```
i = 0
while i < 5
  puts i
  i += 1
  break if i == 2
end
```

Циклы **while, until**

```
i = 0
until i == 5
   puts i
   i += 1
end
```

Отличие until от while - цикл выполняется до тех пор, пока *условие* не станет ***true***

Циклы **while, until**

```
i = 0
num = 5
begin
   puts("Inside the loop i = #{i}” )
   i +=1;
end until i > num
```

Выражения **break, next, redo, retry**

```
for i in 0..5
   if i > 2 then
      break
   end
   puts "Value of local variable is #{i}"
end
```

Выражения **break, next, redo, retry**

```
for i in 0..5
   if i < 2 then
      next
   end
   puts "Value of local variable is #{i}"
end
```

Выражения **break, next, redo, retry**

```
for i in 0..5
   if i < 2 then
      puts "Value of local variable is #{i}"
      redo
   end
end
```

Выражения **break, next, redo, retry**

```
for i in 1..5
   retry if  i > 2
   puts "Value of local variable is #{i}"
end
```

Case

```
car = "Patriot"

manufacturer = case car
  when "Focus" then "Ford"
  when "Navigator" then "Lincoln"
  when "Camry" then "Toyota"
  when "Civic" then "Honda"
  when "Patriot" then "Jeep"
  when "Jetta" then "VW"
  when "Cayene" then "Porsche"
  when "Outback" then "Subaru"
  when "520i" then "BMW"
  when "Tundra" then "Toyota"
  else "Unknown"
end

puts "The " + car  + " is made by "  + manufacturer
```

Exceptions

```
begin
# -
rescue OneTypeOfException
# -
rescue AnotherTypeOfException
# -
else
# Other exceptions
ensure
# Always will be executed
end
```

```
begin
   file = open("/unexistant_file")
   if file
      puts "File opened successfully"
   end
rescue
      file = STDIN
end
print file, "==", STDIN, "\n"
```

```ruby
filename = "/unexistant_file"
begin
  file = open(filename)
  if file
    puts "File opened successfully"
  end
rescue
  filename = "existant_file"
  retry
end
```

raise Exception

raise

или

raise "Error Message"

или

raise ExceptionType, "Error Message"

или

raise ExceptionType, "Error Message" condition

raise Exception

```
begin
    puts 'I am before the raise.'
    raise 'An error has occurred.'
    puts 'I am after the raise.'
rescue
    puts 'I am rescued.'
end
puts 'I am after the begin block.'
```

ensure

```
begin
   #.. process
   #..raise exception
rescue
   #.. handle error
ensure
   #.. finally ensure execution
   #.. This will always execute.
end
```

ensure

```
begin
  raise 'A test exception.'
rescue Exception => e
  puts e.message
  puts e.backtrace.inspect
ensure
  puts "Ensuring execution"
end
```

ООП

Классы

```
class Human
    some_code
end
```

Классы - инициализация

```
class Human
    def initialize(first_name, last_name)
        @name = [first_name, last_name].join(“ ”)
    end
end
```

Классы - методы

```
class Human
    def initialize(first_name, last_name)
        @name = [first_name, last_name].join(“ ”)
    end

    def printName()
       puts @name
    end
end
```

Переменные

Глобальные переменные

```
$global_variable = 10
class Class1
  def print_global
     puts "Global variable in Class1 is #$global_variable"
  end
end

class Class2
  def print_global
     puts "Global variable in Class2 is #$global_variable"
  end
end

class1obj = Class1.new
class1obj.print_global
class2obj = Class2.new
class2obj.print_global
```

Переменные экземпляра

```
class Customer
   def initialize(id, name, addr)
      @cust_id=id
      @cust_name=name
      @cust_addr=addr
   end
   def display_details()
      puts "Customer id #@cust_id"
      puts "Customer name #@cust_name"
      puts "Customer address #@cust_addr"
    end
end

# Create Objects
cust1=Customer.new("1", "John", "Wisdom Apartments, Ludhiya")
cust2=Customer.new("2", "Poul", "New Empire road, Khandala")

# Call Methods
cust1.display_details()
cust2.display_details()
```

Переменные класса

```
class Customer
   @@no_of_customers=0
   def initialize(id, name, addr)
      @cust_id=id
      @cust_name=name
      @cust_addr=addr
   end
   def display_details()
      puts "Customer id #@cust_id"
      puts "Customer name #@cust_name"
      puts "Customer address #@cust_addr"
    end
    def total_no_of_customers()
       @@no_of_customers += 1
       puts "Total number of customers: #@@no_of_customers"
    end
end

# Create Objects
cust1=Customer.new("1", "John", "Wisdom Apartments, Ludhiya")
cust2=Customer.new("2", "Poul", "New Empire road, Khandala")

# Call Methods
cust1.total_no_of_customers()
cust2.total_no_of_customers()
```

Локальные переменные

Локальные переменные начинаются со строчной буквы.

Видимость локальной переменной определяется её классом, методом или модулем.

Константы

```
class Example
   VAR1 = 100
   VAR2 = 200
   def show
      puts "Value of first Constant is #{VAR1}"
      puts "Value of second Constant is #{VAR2}"
   end
end

# Create Objects
object=Example.new()
object.show
```

Псевдо-переменные

self - ссылка на текущий инстанс класса
__FILE__ - ссылка на текущий файл
__LINE__ - ссылка на текущую строку

Домашнее задание

1 - Прочитать про ООП
http://nashbridges.me/introducing-ruby-oop

2 - Прочитать про public, private, protected методы
http://rubyblog.com.ua/2016/05/public-protected-private-in-ruby

3 - Прочитать про замыкания
https://sites.google.com/site/sitsiliyaror/blogs/zamykania

4 - Изучить основы html и css
http://webdesign-master.ru/blog/html-css/4.html
http://css.manual.ru/articles/css_basics

Вопросы?

+7 (926) 889-16-32

alec@alec-c4.com

http://alec-c4.com/