CRIMEA DIGITAL GROUP ACADEMY

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ЛЗ #2 - Типы данных

Теория

Задание

Теория

Задание

Data Types - Integer

```ruby
32.class
# Integer

1 + 2
# => 3

2 - 1
# => 1

4 / 3
# => 1

2 * 2
# => 4
```

```ruby
2 ** 32
# => 4294967296

1 & 0
# => 0

0 | 1
# => 1

7 >> 1
# => 3
# 111>>1 =>11
```

```ruby
7 << 1
# => 14
# 111<<1=>1110

1 ^ 0
# => 1

1.to_s
# => "1"

2.to_f
# => 2.0
```

Data Types - Float

```ruby
32.99.class
# Float

Float("123.123")
# => 123

1.0 / 3.0
# => 0

1.5 * 2
=> 3.0

1.1 + 2.23
# => 3.33

2.124 - 1.111
# => 1.014
```

Data Types - Boolean

```ruby
true.class
# => TrueClass

false.class
# => FalseClass

true && false
# => false

true || false
# => true
```

Data Types - String

```ruby
"string".class
# => String

"OK 👌".encoding
# => #<Encoding:UTF-8>

'string'
# => "string"

'Hello world'.reverse
# => "dlrow olleH"
```

```ruby
"hello"[0]
# => “h”

"Hello"[0, 2]
# => "He"

"hello"[-1]
# => "o"

'"convenient"'
# => "\"convenient\""
```

Data Types - String

```
%{string}
# => "string"

%(string)
# => "string"

%[string]
# => "string"

%{"'\n'"'"}
# => "\"'\n'\"\n'\""
```

Data Types - String

```
"multiline
string"
# => "multiline\nstring"

<<-SQL

SELECT * FROM users
WHERE users.id = 2

SQL
# => "\nSELECT * FROM users\nWHERE users.id = 2\n\n"
```

Data Types - String

```
'foo' + 'bar'
# => "foobar"

'bang!' * 3
# => "bang!bang!bang!"

a = 1
b = 2
"#{a} + #{b} = #{a + b}"
# => "1 + 2 = 3"
```

```
%{#{a} + #{b}}
# => "1 + 2"

<<-INTERPOLATION
#{a} + #{b} = #{a + b}
INTERPOLATION
# => "1 + 2 = 3\n"
```

Data Types - Array

```ruby
[1, 2, 3, 4, 5]
# => [1, 2, 3, 4, 5]

[1 , :foo, 2, "bar", 3, true]
# => [1, :foo, 2, "bar", 3, true]

arr = []
arr << "bar"
arr.push(:foo)

# => ["bar", :foo]
arr.delete("bar")
arr.delete(:foo)
# => []
```

Data Types - Array

```ruby
arr[0]
# => "bar"

arr[1]
# => :foo

arr[0, 1]
# => ["bar", :foo]

arr.slice(1)
# => :foo

arr.slice(0, 7)
# => ["bar", :foo]
```

```ruby
arr.empty?
# => false

arr.size
arr.count
# => 2

arr.empty?
# => true

arr.each do |item|
  puts item
end
# bar
# foo
```

Data Types - Hash

```ruby
grades = { "Jane Doe" => 10, "Jim Doe" => 6 }

grades["John Doe"] = 7

puts grades["John Doe"]
# 7

puts grades
# { "Jane Doe" => 10, "Jim Doe" => 6, "John Doe" => 7 }

{ 1 => 1, "2" => 2, :foo => "bar", 7.3 => [] }
# => {1=>1, "2"=>2, :foo=>"bar", 7.3=>[]}
```

Data Types - Hash

```ruby
grades = { "Jane Doe" => 10, "Jim Doe" => 6 }

grades["John Doe"] = 7

puts grades["John Doe"]
# 7

puts grades
# { "Jane Doe" => 10, "Jim Doe" => 6, "John Doe" => 7 }

{ 1 => 1, "2" => 2, :foo => "bar", 7.3 => [] }
# => {1=>1, "2"=>2, :foo=>"bar", 7.3=>[]}
```

Data Types - Hash

```
student_1 = "John Doe"
student_2 = "Jane Doe"
student_3 = "Jim Doe"

score_1 = 7
score_2 = 9
score_3 = 3

{
  student_1 => score_1,
  student_2 => score_2,
  student_3 => score_3
}

grades["John Doe"] == score_1
# => true
```

Inspect - твой лучший друг

Иногда при возникновении проблем, хочется посмотреть что именно лежит в переменной, для этого есть метод **inspect**

```
{
  first_name: 'Ivan',
  last_name: 'Ivanov'
}.inspect
#=> "{:first_name=>\"Ivan\", :last_name=>\"Ivanov\"}"
```

Теория

Задание

SOFTWARE ENGINEERING

Задание

👉 Написать скрипт, который будет принимать число и слово, если слово заканчивается на “CS” - выводит на экран цифру 2 в степени (длины введенного слова), если не заканчивается - выводит слово задом наперед

👉 Написать скрипт, который будет выводить массив покемонов

- Спросит сколько добавить покемонов
- Указанное на предыдущем этапе число раз спросит имя и цвет каждого покемона
- Выведет в консоль массив содержащий хеши покемонов в формате

```
[{ name: 'Pikachu', color: 'Yellow' }]
```

* Старайтесь организовывать код в методы

КОНТАКТЫ

Для быстрого входа
на сайт

courses@crimeadigital.ru

tel: 8 (800) 551 44 86

https://crimeadigital.ru