FUTURE
SIMPLE

RUBY 101

RUBY FEATURES

RUBY FEATURES

Interpreted

RUBY FEATURES

Interpreted

Dynamic typing

RUBY FEATURES

Interpreted

Dynamic typing

Object-Oriented

EXECUTION

```
ruby hello.rb
```

SYNTAX

```ruby
puts "Hello world!"
puts("Hello world!")
```

NUMBERS

```ruby
42.class   # => Fixnum
3.14.class # => Float
```

STRINGS

```
string = "some string"

string.upcase # => "SOME STRING"
```

ARRAYS

```
array = [1, 2, "three"]

array.first # => 1
array.last  # => "three"
```

RANGES

```
1..10

'a'..'z'
```

HASHES

```
hash = {
   :one => "jeden",
   :two => "dwa"
}
```

HASHES - Ruby 1.9

```ruby
hash = {
  one: "jeden",
  two: "dwa"
}
```

REGULAR EXPRESSIONS

```ruby
regex = Regexp.new('^\s*[a-z]')
regex = /^\s*[a-z]/

"string" =~ regex    # => true
regex    =~ "string" # => true
```

SYMBOLS

```
:some_symbol
```

CONTROL STRUCTURES

```ruby
i = 0
while i < 10
  puts i
  i += 1
end
```

CONTROL STRUCTURES

```
if car.speed > 130
  "you're speeding!"
elsif car.speed > 110
  "watch your speed"
else
  "you're ok"
end

"speeding" if car.speed > 130
```

CONTROL STRUCTURES

```ruby
case speed
  when 0..110
    "ok"
  when 110..130
    "watch your speed"
  else
    "speeding!"
end
```

EXCEPTION HANDLING

```
begin
  broken.code
rescue Exception => e
  puts "Error! #{e}"
ensure
  puts "This will always execute"
end
```

OBJECTS

```
car = Car.new
```

CLASSES

```
class Car

end
```

ATTRIBUTES

```
class Car
  attr_accessor :speed

end

car = Car.new
car.speed = 10
car.speed # => 10
```

CONSTRUCTOR

```ruby
class Car
  attr_accessor :speed

  def initialize
    @speed = 0
  end

end
```

METHODS

```
class Car
  attr_accessor :speed

  def initialize
    @speed = 0
  end

  def accelerate(speed)
    @speed += speed
  end

end
```

INHERITANCE

```ruby
class SportsCar < Car

end
```

INHERITANCE

```ruby
class SportsCar < Car

  def initialize
    super
    @speed = 100
  end

end
```

CLASS METHODS

```ruby
class Car

  def self.is_speeding?(car)
    car.speed > 130
  end

end

car = SportsCar.new
Car.is_speeding?(car) # => true
```

MODULES - mixins

```ruby
module Transport

  def load(content)
    @container ||= []
    @container.push(content)
  end

  def unload
    @container.slice! 0..-1
  end

end

class Car
  include Transport

end
```

MODULES - namespaces

```ruby
module SuperCars
  class Ferrari < Car
  end
end

ferrari = SuperCars::Ferrari.new
```

ITERATORS

```
[1, 2, 3].each do |num|
  puts num
end

[1, 2, 3].each { |num| puts num }
```

BLOCKS

```
class Array

  def each_odd
    for i in 0...self.length
      yield(self[i]) if self[i].odd?
    end
  end

end
```

```
[1,2,3,4,5,6].each_odd { |n| puts n }
```