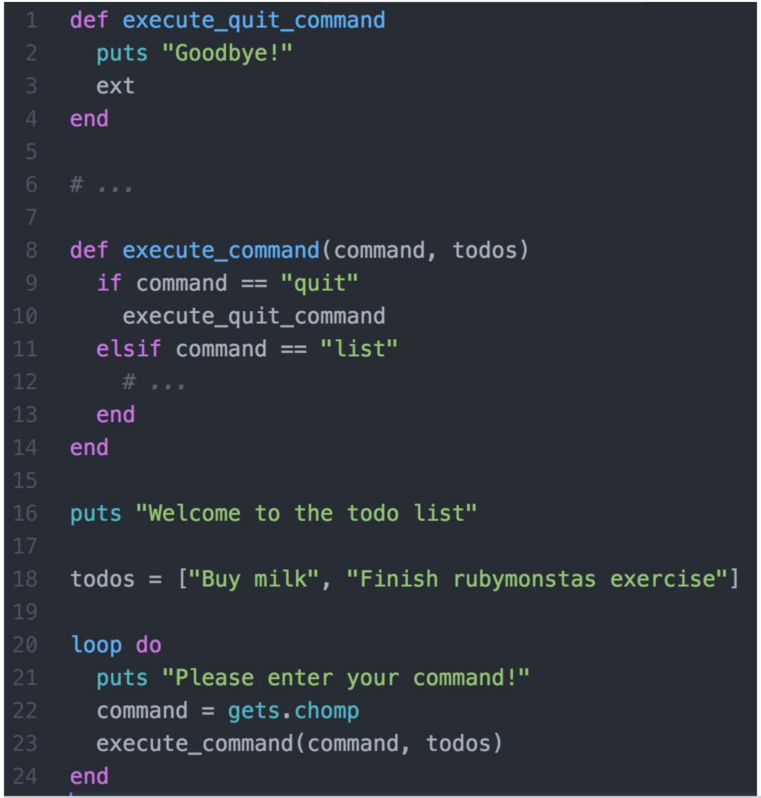
Ruby Monstas

Session 6

Agenda

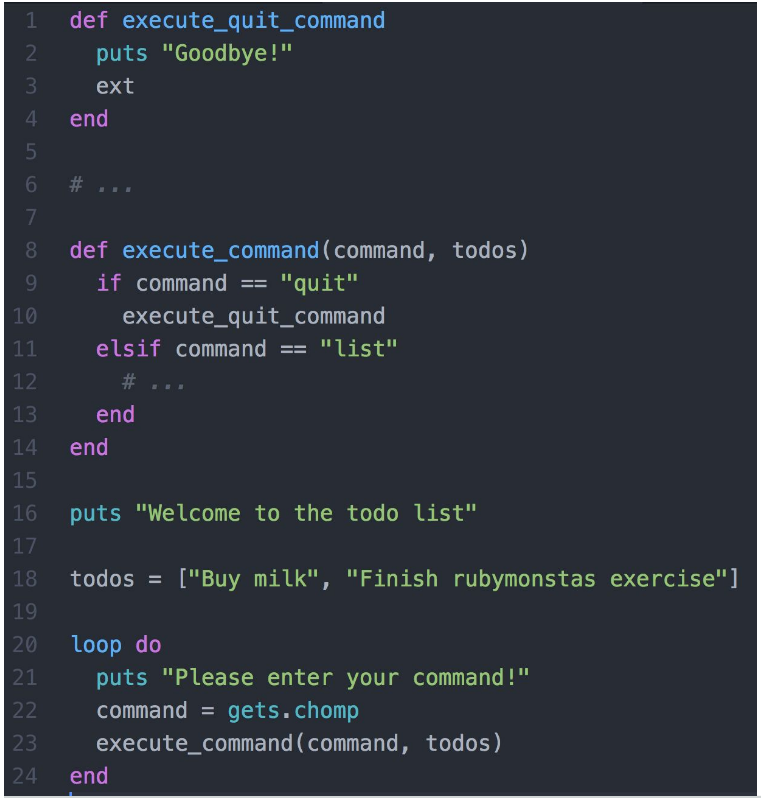
- Error messages
- Interactive Error message exercise
- Boolean Operators
- Exercises

Anatomy of a stack trace



```ruby
def execute_quit_command
  puts "Goodbye!"
  ext
end

# ...

def execute_command(command, todos)
  if command == "quit"
    execute_quit_command
  elsif command == "list"
    # ...
  end
end

puts "Welcome to the todo list"

todos = ["Buy milk", "Finish rubymonstas exercise"]

loop do
  puts "Please enter your command!"
  command = gets.chomp
  execute_command(command, todos)
end
```

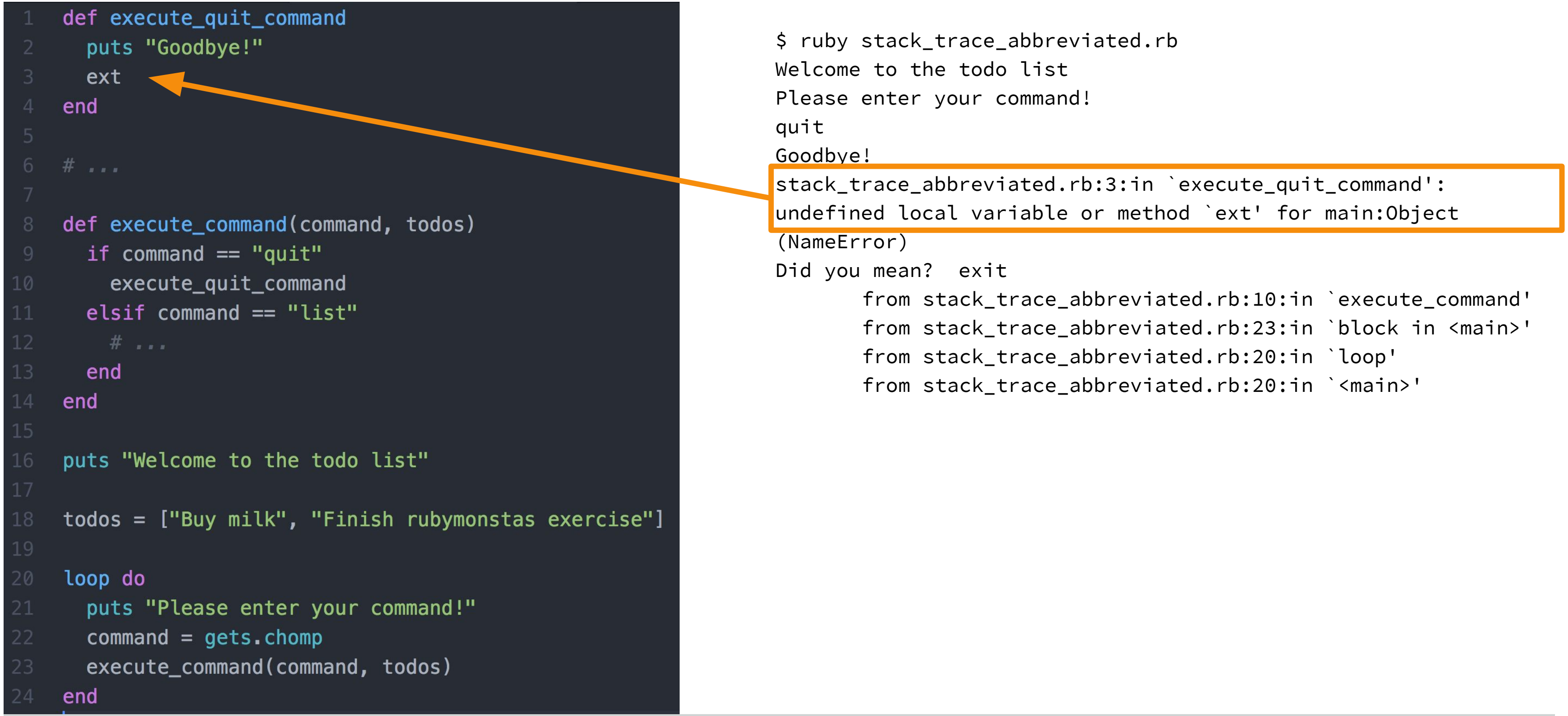
Anatomy of a stack trace



```ruby
 1  def execute_quit_command
 2    puts "Goodbye!"
 3    ext
 4  end
 5
 6  # ...
 7
 8  def execute_command(command, todos)
 9    if command == "quit"
10      execute_quit_command
11    elsif command == "list"
12      # ...
13    end
14  end
15
16  puts "Welcome to the todo list"
17
18  todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20  loop do
21    puts "Please enter your command!"
22    command = gets.chomp
23    execute_command(command, todos)
24  end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
        from stack_trace_abbreviated.rb:10:in `execute_command'
        from stack_trace_abbreviated.rb:23:in `block in <main>'
        from stack_trace_abbreviated.rb:20:in `loop'
        from stack_trace_abbreviated.rb:20:in `<main>'
```

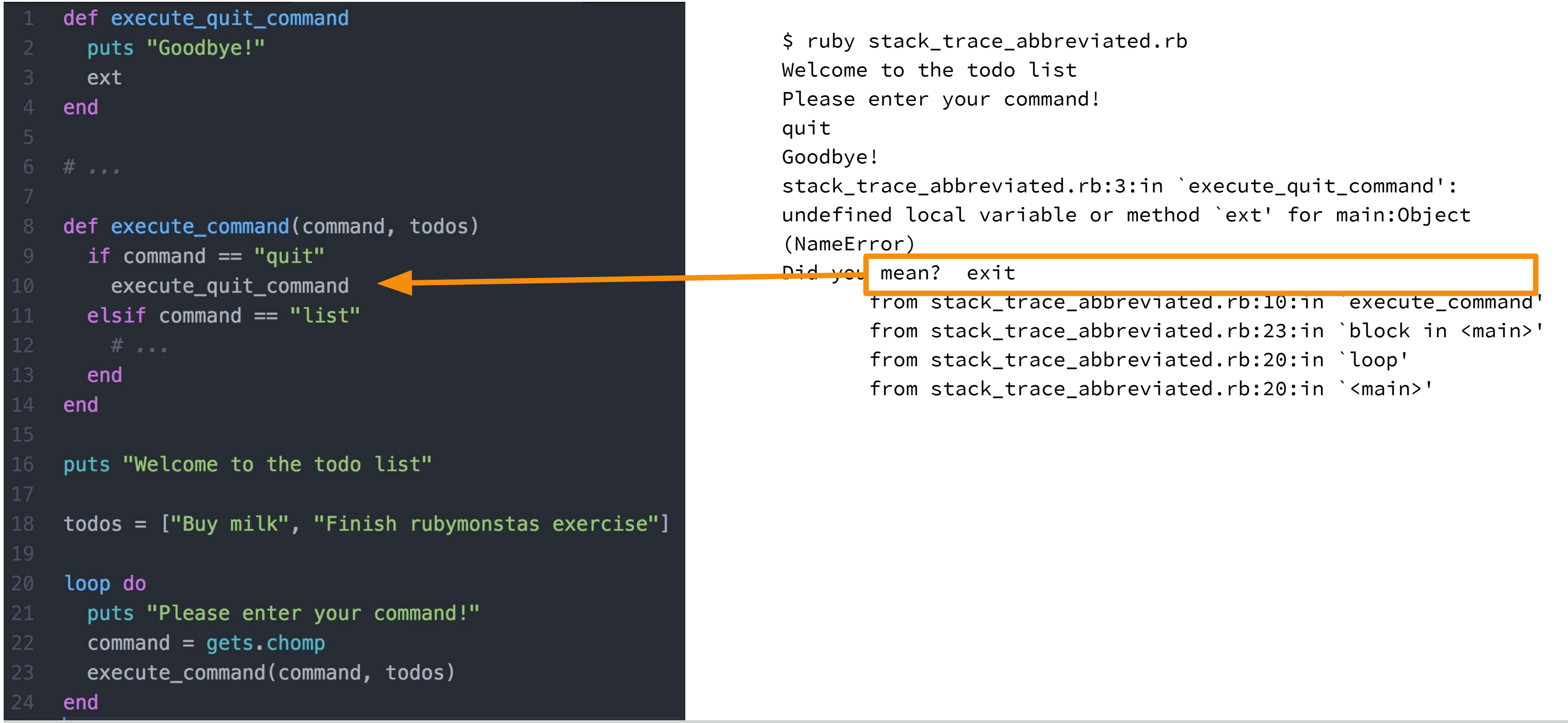
Anatomy of a stack trace



```ruby
 1  def execute_quit_command
 2    puts "Goodbye!"
 3    ext
 4  end
 5
 6  # ...
 7
 8  def execute_command(command, todos)
 9    if command == "quit"
10      execute_quit_command
11    elsif command == "list"
12      # ...
13    end
14  end
15
16  puts "Welcome to the todo list"
17
18  todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20  loop do
21    puts "Please enter your command!"
22    command = gets.chomp
23    execute_command(command, todos)
24  end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
        from stack_trace_abbreviated.rb:10:in `execute_command'
        from stack_trace_abbreviated.rb:23:in `block in <main>'
        from stack_trace_abbreviated.rb:20:in `loop'
        from stack_trace_abbreviated.rb:20:in `<main>'
```

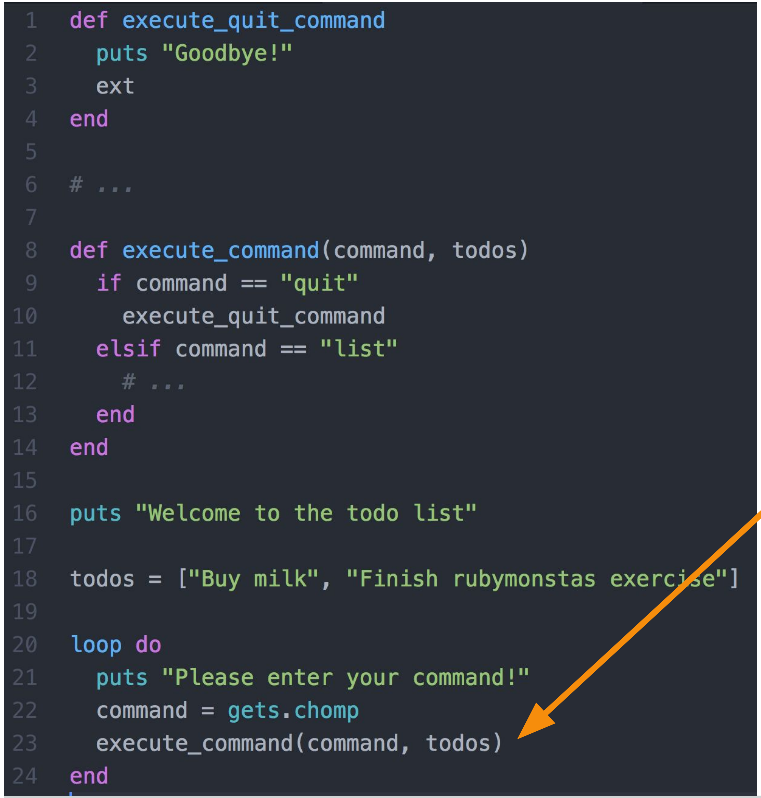
Anatomy of a stack trace



```ruby
1  def execute_quit_command
2    puts "Goodbye!"
3    ext
4  end
5
6  # ...
7
8  def execute_command(command, todos)
9    if command == "quit"
10     execute_quit_command
11   elsif command == "list"
12     # ...
13   end
14 end
15
16 puts "Welcome to the todo list"
17
18 todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20 loop do
21   puts "Please enter your command!"
22   command = gets.chomp
23   execute_command(command, todos)
24 end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
       from stack_trace_abbreviated.rb:10:in `execute_command'
       from stack_trace_abbreviated.rb:23:in `block in <main>'
       from stack_trace_abbreviated.rb:20:in `loop'
       from stack_trace_abbreviated.rb:20:in `<main>'
```

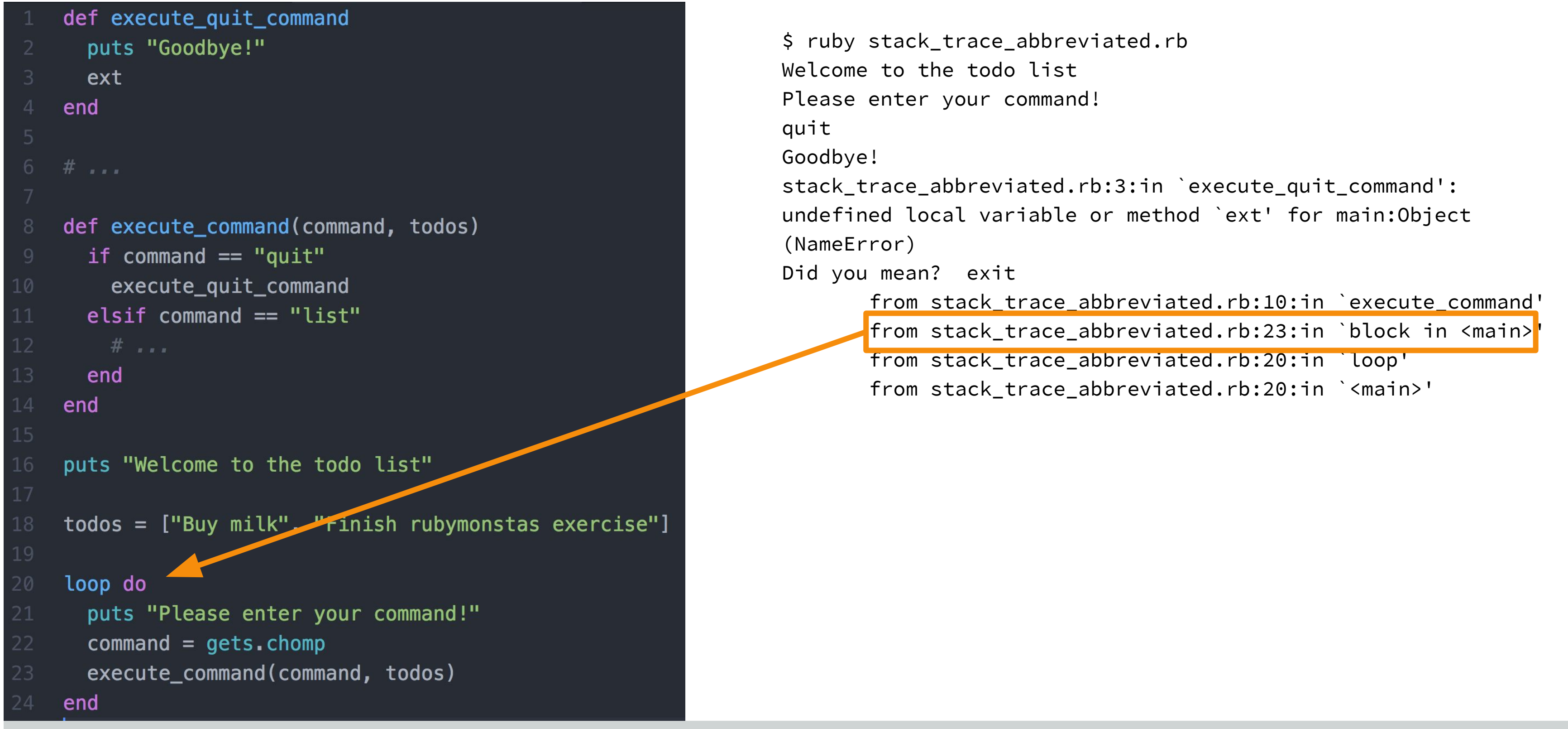
Anatomy of a stack trace



```
 1  def execute_quit_command
 2    puts "Goodbye!"
 3    ext
 4  end
 5
 6  # ...
 7
 8  def execute_command(command, todos)
 9    if command == "quit"
10      execute_quit_command
11    elsif command == "list"
12      # ...
13    end
14  end
15
16  puts "Welcome to the todo list"
17
18  todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20  loop do
21    puts "Please enter your command!"
22    command = gets.chomp
23    execute_command(command, todos)
24  end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
          from stack_trace_abbreviated.rb:10:in `execute_command'
          from stack_trace_abbreviated.rb:23:in `block in <main>'
          from stack_trace_abbreviated.rb:20:in `loop'
          from stack_trace_abbreviated.rb:20:in `<main>'
```

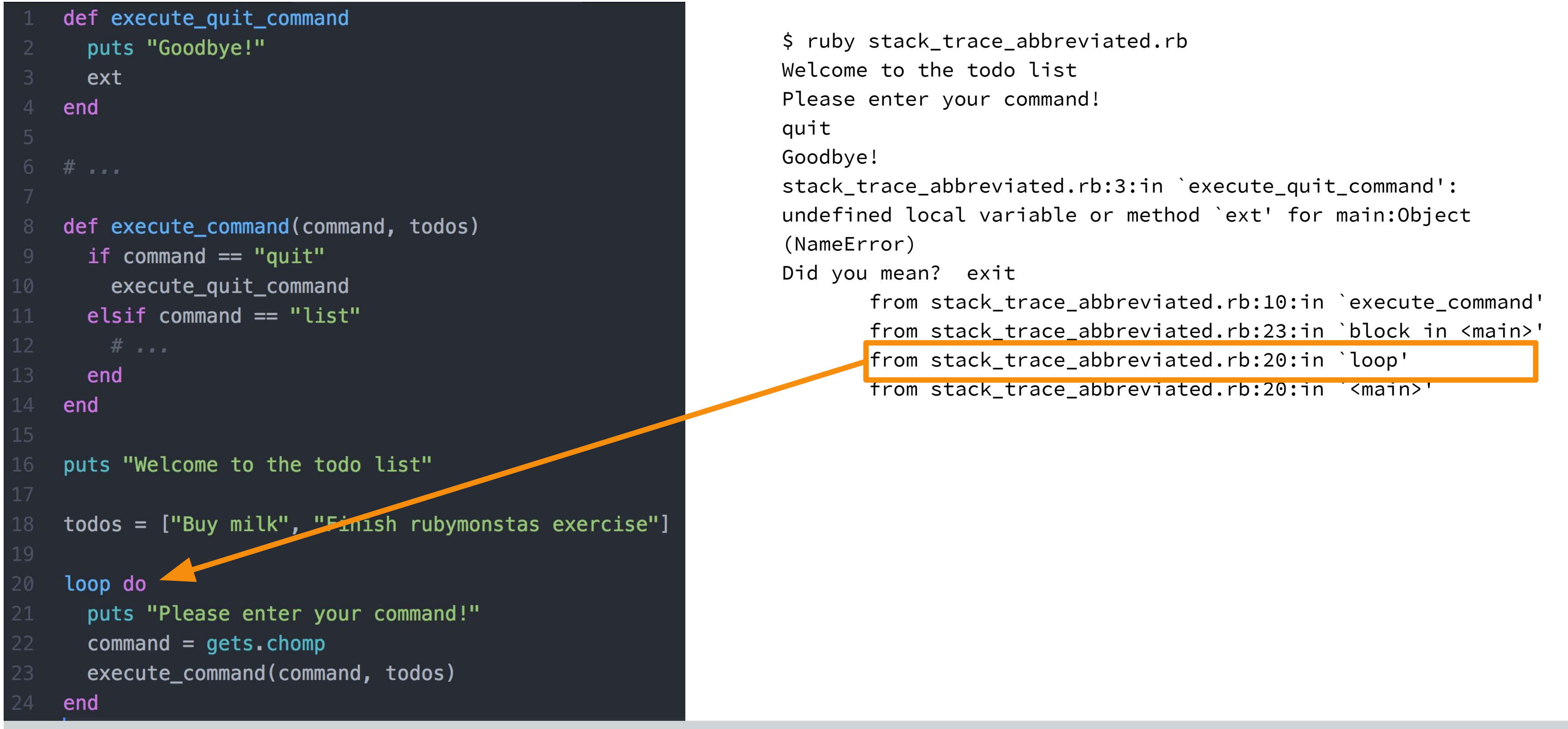
Anatomy of a stack trace



```
1   def execute_quit_command
2     puts "Goodbye!"
3     ext
4   end
5
6   # ...
7
8   def execute_command(command, todos)
9     if command == "quit"
10      execute_quit_command
11    elsif command == "list"
12      # ...
13    end
14  end
15
16  puts "Welcome to the todo list"
17
18  todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20  loop do
21    puts "Please enter your command!"
22    command = gets.chomp
23    execute_command(command, todos)
24  end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
        from stack_trace_abbreviated.rb:10:in `execute_command'
        from stack_trace_abbreviated.rb:23:in `block in <main>'
        from stack_trace_abbreviated.rb:20:in `loop'
        from stack_trace_abbreviated.rb:20:in `<main>'
```

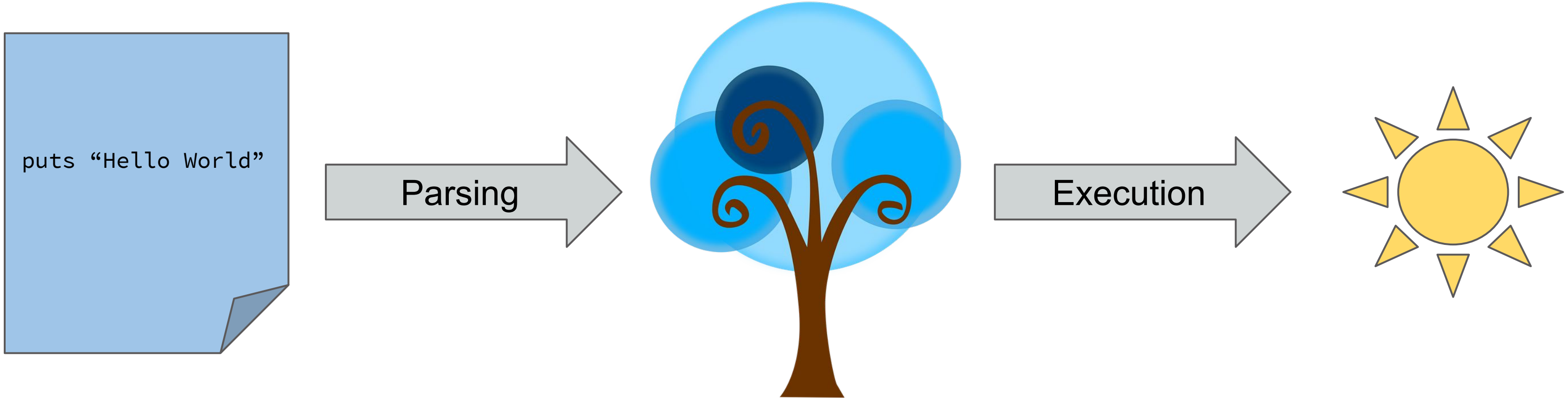
Anatomy of a stack trace



```ruby
1  def execute_quit_command
2    puts "Goodbye!"
3    ext
4  end
5
6  # ...
7
8  def execute_command(command, todos)
9    if command == "quit"
10     execute_quit_command
11   elsif command == "list"
12     # ...
13   end
14 end
15
16 puts "Welcome to the todo list"
17
18 todos = ["Buy milk", "Finish rubymonstas exercise"]
19
20 loop do
21   puts "Please enter your command!"
22   command = gets.chomp
23   execute_command(command, todos)
24 end
```

```
$ ruby stack_trace_abbreviated.rb
Welcome to the todo list
Please enter your command!
quit
Goodbye!
stack_trace_abbreviated.rb:3:in `execute_quit_command':
undefined local variable or method `ext' for main:Object
(NameError)
Did you mean?  exit
        from stack_trace_abbreviated.rb:10:in `execute_command'
        from stack_trace_abbreviated.rb:23:in `block in <main>'
        from stack_trace_abbreviated.rb:20:in `loop'
        from stack_trace_abbreviated.rb:20:in `<main>'
```

How does Ruby execute code?



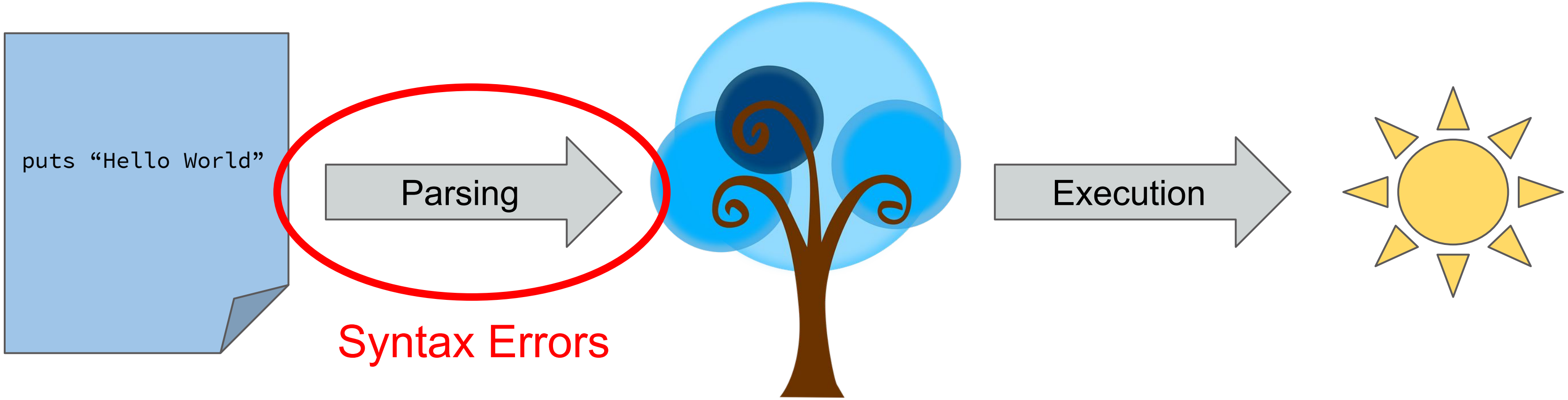
```
puts “Hello World”
```

Parsing

Execution

Abstract Parse Tree

How does Ruby execute code?



```
puts “Hello World”
```

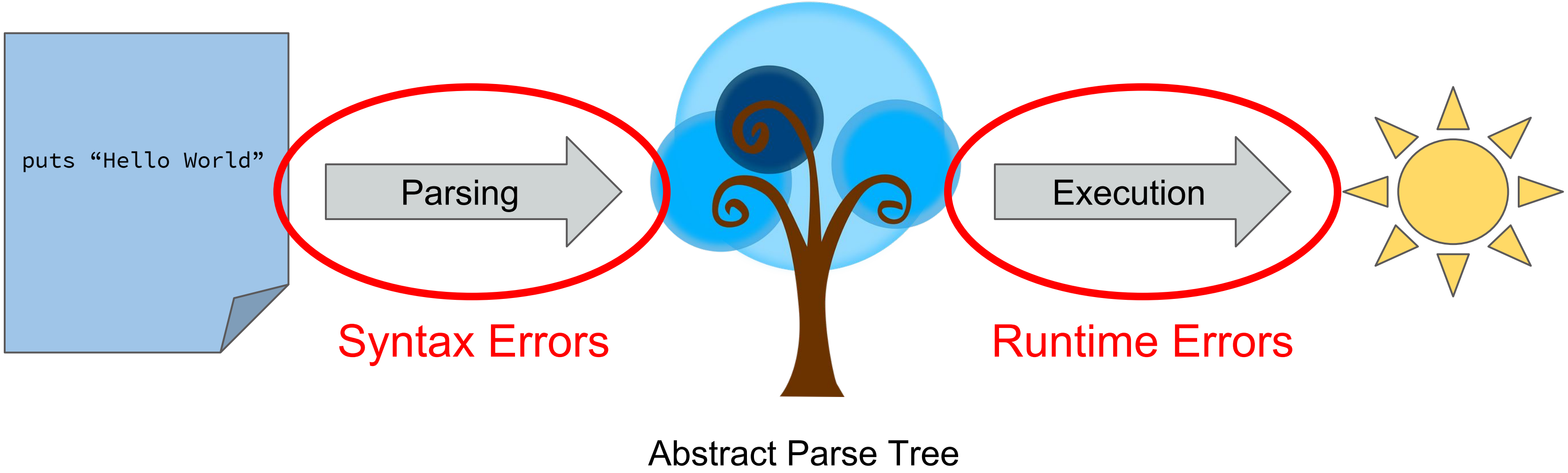
Parsing

Syntax Errors

Abstract Parse Tree

Execution

How does Ruby execute code?



puts “Hello World”

Parsing

Syntax Errors

Abstract Parse Tree

Execution

Runtime Errors

Syntax error

- Occurs before actually executing your code
- Your Ruby file does not adhere to the syntax rules of Ruby
- Syntax highlighting can help avoid these errors

```
def break_words(stuff)
  words = stuff.split(' ')
  return words
ed
```

```
session-6-error-exercise-solution.rb:85: syntax error,
unexpected end-of-input, expecting keyword_end
```

Syntax error

```
[1, 2, 3].each
  puts integer
end
```

```
session-6-error-exercise-solution.rb:60: syntax error,
unexpected keyword_end, expecting end-of-input
```

Runtime errors

- Occurs during execution of your code
- Includes a backtrace
- Logic, naming, unexpected input, …

```ruby
def criticise_user_name(name)
  proper_name_length = 9

  if name.length == proper_name_length
      puts "Now that's a proper name!"
  els
      puts "This name is too long or too short!"
  end
end
```

```
session-6-error-exercise-syntax-fixed.rb:54:in `criticise_user_name': undefined local variable or method
`els' for main:Object (NameError)
    from session-6-error-exercise-syntax-fixed.rb:97:in `<main>'
```

Interactive error message exercise

Boolean Operators

Not

```
if !condition
  # the condition is false
end
```

```
if condition1 != condition2
  # condition1 and condition2 are not equal
end
```

condition is a **boolean**.

Not: example

```
irb> name = "John"
=> "John"
irb> is_guest = false
=> false
irb> if !is_guest
irb>   puts "Welcome back, #{name}"
irb> end
Welcome back, John
=> nil
```

AND

```
if condition1 && condition2
  # condition1 and condition2 are true
end
```

condition is a boolean.

AND: example

```
irb> age = 26
=> 26
irb> if age > 25 && age < 50
irb>   puts "Your age is between 25 and 50"
irb> end
Your age is between 25 and 50
=> nil
```

OR

```
if condition1 || condition2
  # condition1 or condition2 is true, or both are true
end
```

```
condition is a boolean.
```

OR: example

```
irb> name = "root"
=> "root"
irb> is_admin = true
=> true
irb> if name == "root" || is_admin
irb>   puts "You have super powers!"
irb> end
You have super powers!
=> nil
```

Time to practice

Let’s get to it!