Introducing Roo

Dr Garry Pettet
Software Engineer & Radiologist

Who Am I?

- UK Radiologist
- Xojo developer since 2004
- Interest in low-level problem solving
- This is my first XDC

Outline

- Background
- Technical Implementation
- Integration
- Walk Through
- Questions

What Is Roo?

- Dynamically typed, object-oriented language
- Expressive
- Extensible
- Embeddable
- Interpreted

What Is Roo?

Everything is an object

```
9.sqrt # 3
"hello world".capitalise # "HELLO WORLD"
35.5.integer? # False
```

What Is Roo?

Full class system

```
class Person:
  def init(name): # Note optional constructor.
    self.name = name

  def greet(who):
    print("Hello " + who)

class Doctor < Person:
  def init(name):
    super.init(name)

  def diagnose(who):
    print("OK " + who + ", let's see what's wrong with you")

var dave = Person("Dave")
dave.greet("Tony") # "Hello Tony"

var garry = Doctor("Garry")
garry.diagnose("Steven") # "OK Steven, let's see what's wrong with you"
```

What Is Roo?

Array and Hash (Dictionary) support

```
# Arrays of mixed types are supported.
var arr1 = ["Female", 36, True]
# Arrays can contain other arrays.
var arr2 = [1, 2, ["a", "b"]]

# Hashes (aka Dictionaries).
var person = {"name" => "Tony Stark", "rich?" => True}
person{"name"} # "Tony Stark"
```

Why Create Roo?

- Grew out of a home automation project
- Xojoscript limitations
- Academic curiosity:
  - What are the limits of Xojo?

Roo Vs Xojoscript

- Easier to pass objects between scripts and your app
- Extensive standard library
  - Regex
  - Networking
  - File manipulation
- Less verbose

Implementation

- Scanner
- Parser
- Analyser
- Interpreter

Scanner

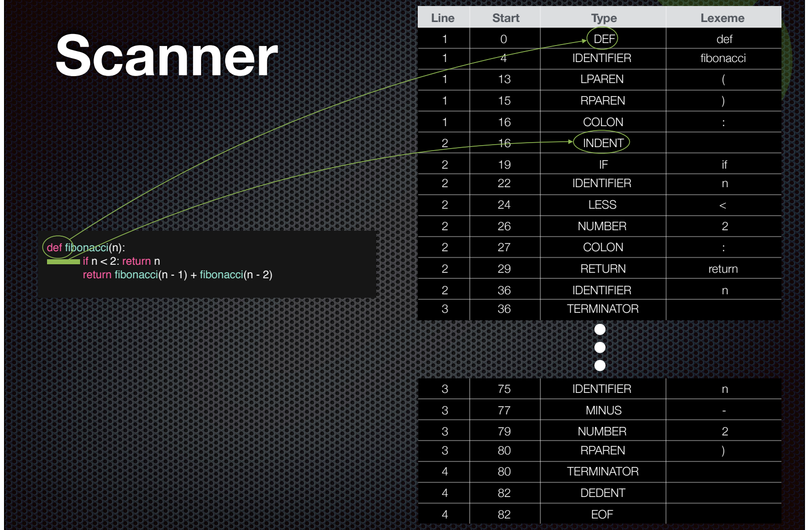
- Converts source code into **Tokens** for the Parser
- Tokens contain metadata such as line number, the source file, lexeme, etc
- Catches typographical and simple errors

Scanner

```
var name = 'Garry Pettet'
```

var	name	=	'Garry Pettet'
VAR_KEYWORD "var"	IDENTIFIER "name"	EQUALS "="	TEXT "Garry Pettet"

Scanner



```
def fibonacci(n):
    if n < 2: return n
    return fibonacci(n - 1) + fibonacci(n - 2)
```

Line	Start	Type	Lexeme
1	0	DEF	def
1	4	IDENTIFIER	fibonacci
1	13	LPAREN	(
1	15	RPAREN	)
1	16	COLON	:
2	16	INDENT	
2	19	IF	if
2	22	IDENTIFIER	n
2	24	LESS	<
2	26	NUMBER	2
2	27	COLON	:
2	29	RETURN	return
2	36	IDENTIFIER	n
3	36	TERMINATOR	

⋮

3	75	IDENTIFIER	n
3	77	MINUS	-
3	79	NUMBER	2
3	80	RPAREN	)
4	80	TERMINATOR	
4	82	DEDENT	
4	82	EOF	

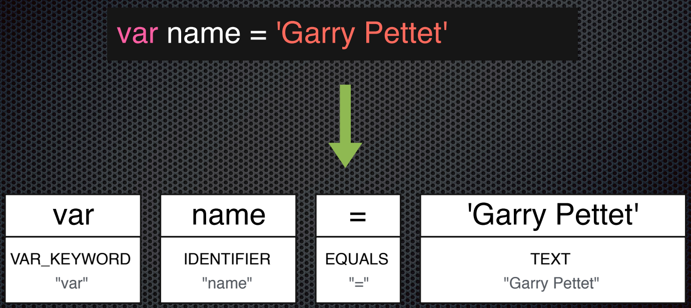
Scanner

- Other functions:
  - Cross-platform line endings
  - Multiline statements
  - Inclusion of multiple files

Parser

- Takes an array of **Tokens** and produces an array of **RooStmt** objects
- Enforces Roo's grammar
- Produces a "tree" of nodes that can be traversed

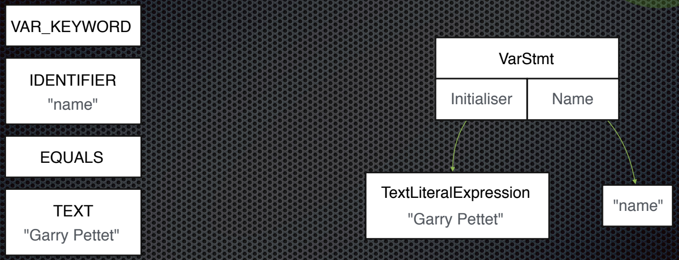
Parser



`var name = 'Garry Pettet'`

var	name	=	'Garry Pettet'
VAR_KEYWORD "var"	IDENTIFIER "name"	EQUALS "="	TEXT "Garry Pettet"

Parser



VAR_KEYWORD

IDENTIFIER
"name"

EQUALS

TEXT
"Garry Pettet"

Analyser

- Runs after the Parser but before the Interpreter
- Two roles:
    - Basic semantic analysis
    - Variable resolution and binding

Interpreter

- Traverses or "walks" the AST to visit each node
- Performs actions depending on the node type
- Manages variable allocation, scope, runtime error handling
- Essentially maps your code to Xojo code

Error Handling

- The Interpreter class raises its ErrorOccurred event if an error occurs during:
  - Scanning
  - Parsing
  - Analysis
  - Runtime

Script Safety

- Permit/deny network access on a per-URL basis or globally
- Enable/disable file system "safe mode"

Standard Library

- **Boolean**, **Number**, **Text**, **DateTime** and **Nothing** datatypes
- Built-in **Array** and **Hash** (aka Dictionary) support
- Classes and functions to handle:
  - Regular expressions
  - HTTP/HTTPS
  - File system access

Integration

- Drop in the folder containing the required classes
- Instantiate a **RooInterpreter**
- Add handlers for any events you're interested in
- Optionally extend the standard library with your own modules, classes or functions
- Call the **Interpret()** method on the **RooInterpreter**

Demo

What's Next?

- Bytecode virtual machine
  - Stack-based
  - Performance ++
- iOS support

Project homepage

https://roolang.org

Questions?

Garry Pettet

Website: https://garrypettet.com

GitHub: https://github.com/gkjpettet