Scala Expressions and Types

1-20-2012

Opening Discussion

- Minute essay comments:
  - Color coding in vi, not the run.
  - Windows is proprietary. Linux is free and open source.
  - Where did Hello World come from?
  - Can Linux run C variants?
  - Will you be able to write programs that hackers use to troll other people?
  - What do “thread” and “string” mean in the command line world?
  - Giving you file a name in vi.

More

- Will Windows make things more difficult?
- How do you download Java and Scala on a Mac?
- Significance of wildcards.
- Differences between vi and other editors.
- Making a command cheat-sheet.
- Compiling while in vi?
- How long does it take me to write a program for my research? Do I have students write them or do I?
- Disconnecting from a machine – logout or exit.
- In this course we will code in vi until the last few weeks.

Even More

  - Speed of the class.
  - Sugar-cube sized supercomputers not likely to find their way into your tea for a while.
  - What are reliable careers aside from “law and gardening”? (Neither law nor gardening is safe.)
- IcP Solutions

Scala REPL

- If you just type scala and don't provide a file name to run, it will drop you into the REPL (Read, Evaluate, Print Loop).
- Here you can enter individual commands and have them run.
- It is great for testing things out and getting to learn the language.

Key Terms

- Token – A set of characters that has meaning to the language.
- Statement – A set of tokens that give a complete instruction.
- Expression – Tokens put together that produce a value.
- Type – All values have types. A type is a set of values and the operations allowed on them.
- Literal – A token that represents a value.
  - Numeric, String, Character, Boolean

Statements and Semicolon Inference

- In Scala, as with most programming languages, programs are made by putting together statements.
- In Scala, any expression is a valid statement as are a few other constructs.
- Statements end with semicolons, but they will be inferred at the end of a line if they make sense so you rarely type them.

Operators

- We can build longer expressions by putting literals together with operators.
- Let's start off by playing with some of the numeric operations you are probably familiar with.
  - +, -, *, /
- You can get the remainder after division with %.

Objects

- An object is defined to be information along with the things you can do with that information.
- The information in an object is called the properties.
- The actions are called methods.
- In Scala, even things like Int are objects and have methods on them.

Methods

- The normal way to call a method in Scala (and most other object-oriented languages) is to put a period after the object and follow it with the method name.
- The REPL will do tab completion and list methods for you.
- Let's look at the methods on some basic types and try calling them.

Arguments

- Some methods need additional information to work.
- To give this to the method we pass in arguments.
- Arguments are put in parentheses and separated by commas if there is more than one.
- The parentheses are generally optional in Scala if there is no argument.

Operator Syntax

- All the “operators” in Scala are really just methods.
- Scala allows any method with zero or one arguments to be called with an operator syntax.
- That means you leave off the dot and the parentheses.
- If a method takes no arguments you can call it without the dot.

Minute Essay

- What in today's lectures was confusing? Was there anything that surprised you?