Web Application Architectures

Module 4: The Ruby Programming Language
Lecture 3: Objects and Variables

THE UNIVERSITY of NEW MEXICO

Objects

- Everything is Ruby is an object, the `Object` class is the parent class of all classes in Ruby. Its methods are therefore available to *all* objects unless explicitly overridden.

- An important method in the `Object` class is `class()`. It returns the "type" of an object.

```
> 1.class()    # => Fixnum
> 1.class      # => Fixnum
> 1.0.class    # => Float
> "Foo".class  # => String
```

- Notice how parentheses are optional – they are commonly omitted.

- The language syntax is sensitive to the capitalization of identifiers, in most cases treating capitalized variables as constants.

Variables

Ruby does not use variable declarations, if you assign a value to a literal, an "appropriate" variable named after that literal is created.

Ex.

```
> a = 2          # => 2
> a              # => 2
```

In this example, `a` has type `Fixnum`, this is an integer data in Ruby. The other integer type is `Bignum` (represents numbers of arbitrary size).

Ex.

```
> a = "2"        # => "2"
> a              # => "2"
```

Now `a` is a String variable.

Variables

- Important: All assignments are done by reference in Ruby. I.e, a variable just holds a reference to an object, and does not care about the type of the object.

- Ruby supports parallel assignment.
  Ex. You can easily swap the values stored in two variables:

```
> a = 2          # => 2
> b = 1          # => 1
> puts a, b      # 2
                 # 1
                 # => nil
a, b = b, a      # => [1, 2]
```

Variables

- Ruby uses simple naming conventions to denote the scope of variables:
  - `name` – could be a local variable.
  - `@name` – an instance variable.
  - `@@name` – a class variable.
  - `$name` – a global variable.

  The @ and $ sigils enhance readability by allowing the programmer to easily identify the roles of each variable.
- Furthermore, local variables must begin with a lowercase letter, and the convention is to use underscores, rather than camel case, for multi-word names.
- Constants are any name that starts with an uppercase letter, and the convention is to use underscores.
- Classes and modules are treated as constants, so they begin with uppercase letters, and the convention is to use camel case.