

Ruby Names

```
# p007dt.rb

# Ruby names help to distinguish their intended use.

# Instance variable name starts with a @ sign.
@my_ivar

# Class variable name starts with a @@ sign.
@@my_cvar

# Global variable name starts with a $ sign.
$my_gvar

# Constant names starts with an uppercase letter.
# Class names, Module names and Constants use capitalization.
puts ConstantName
puts Float::DIG
puts Float::MAX

# Method names should begin with a lowercase letter.
# Use _underscores to separate multiword method or variable names.
def my_method;end

# ?, ! and = are the only weird method suffixes allowed.
def is_my_method?;end

# ! or bang labels a method as dangerous-specifically.
def my_dangerous_method!;end

def my_assignment_method;end

# # Ruby is dynamic
x = 7          # integer
x = "house"    # string
x = 7.5        # real

# # In Ruby, everything you manipulate is an object
'I love Ruby'.length
# 11
```

```
# The basic types in Ruby are:
# Numeric (subtypes include Fixnum, Integer, and Float)
# http://www.ruby-doc.org/core-1.9.3/Numeric.html

# Fixnum http://www.ruby-doc.org/core-1.9.3/Fixnum.html
# Integer http://www.ruby-doc.org/core-1.9.3/Integer.html
# Float http://www.ruby-doc.org/core-1.9.3/Float.html

# String
# http://www.ruby-doc.org/core-1.9.3/String.html

# Array
# http://www.ruby-doc.org/core-1.9.3/Array.html

# Hash
# http://www.ruby-doc.org/core-1.9.3/Hash.html

# Object
# http://www.ruby-doc.org/core-1.9.3/Object.html

# Symbol
# http://www.ruby-doc.org/core-1.9.3/Symbol.html

# Range
# http://www.ruby-doc.org/core-1.9.3/Range.html

# Regexp.
# http://www.ruby-doc.org/core-1.9.3/Regexp.html
```

```

s = 'hello'
s.class
# String

puts 'I am an object = ' + self.to_s
# I am an object = main

puts 'I am in class = ' + self.class.to_s
# I am in class = Object

print 'The object methods are = '
# The object methods are =

puts self.private_methods.sort
# Array
# Complex
# Float
# Integer
# Rational
# String
# __callee__
# __method__
# `
# abort # continues

5.times { puts "Mice!\n" } # more on blocks later
# Mice!
# Mice!
# Mice!
# Mice!
# Mice!

"Elephants Like Peanuts".length
# 22

rice_on_square = 1

64.times do |square|
  puts "On square #{square + 1} are #{rice_on_square} grain(s)"
  rice_on_square *= 2
end

# On square 1 are 1 grain(s)
# On square 2 are 2 grain(s)
# On square 3 are 4 grain(s)
# On square 4 are 8 grain(s) # continues

```