

# Learning Ruby - 9

Classes

# Creating Classes Manually

```
class Person

  def initialize( name, dob, nationality )
    @name = name
    @dob = dob
    @nationality = nationality
  end # of initialize.

end # of Person.

fred = Person.new( 'Fred Jones', '24/05/1950', :Irish )

fred
fred.methods.sort
fred.instance_variables
```

# Classes to Strings - Dynamically!

```
fred.to_s
```

```
class Person
```

```
  def to_s
```

```
    "#@name born on #@dob (@nationality)"
```

```
  end # of to_s.
```

```
end # of Person.
```

```
fred.to_s
```

# Class Get Methods

```
class Person

  def name
    @name
  end # of name.

  def dob
    @dob
  end # of dob.

  def nationality
    @nationality
  end # of nationality.

end # of Person.

fred.name
fred.to_s
fred.dob
```

# Class Set Methods

```
class Person

  def set_name( name )
    @name = name
  end # of set_name.

  def set_dob( dob )
    @dob = dob
  end # of set_dob.

  def set_nationality( nationality )
    @nationality = nationality
  end # of set_nationality.

end # of Person.

fred.name
fred.set_name( "Fred Reginald Jones" )
fred.to_s
```

# Working Less

```
class Person

  def name=( name )
    @name = name
  end # of name=.

  def dob=( dob )
    @dob = dob
  end # of dob=.

  def nationality=( nationality )
    @nationality = nationality
  end # of nationality=.

end # of Person.
```

# Working Even More Less

```
class Person
  attr_reader :name, :dob, :nationality
  attr_writer :name, :dob, :nationality

  def initialize( name, dob, nationality )
    @name = name
    @dob = dob
    @nationality = nationality
  end # of initialize.

  def to_s
    "#@name born on #@dob (@nationality)"
  end # of to_s.
end # of Person.

tom   = Person.new( "Thomas", "26/05/1945", :Irish )
dick  = Person.new( "Richard", "15/02/1980", :English )
harry = Person.new( "Harold", "02/11/1975", :American )

people = [ tom, dick, harry ]

people.each { |person| puts person.to_s }
```

# More ... Ruby So Far

- Creating classes in Ruby is almost too easy
- The **attr\_reader** and **attr\_writer** shortcuts are especially handy
- Of course, as Ruby supports OO, classes can **inherit** from other classes (and from more than one when you use *mixins*)
- **Public**, **Protected** and **Private** access controls are also available
- Chapter 3 of *The PickAxe* has all the details