

Ques 1: Write a simple program for attr_reader, attr_writer and attr_accessor and explain about that.

Ans:

In Ruby, there is no direct external access to the internal state of objects. This means that it is necessary for you to provide public accessors for your internal objects.

Technically, the following code does that just fine:

```
class Message
  def initialize(m)
    @message = m
  end

  def get_message
    @message
  end

  def set_message(m)
    @message = m
  end
end

m = Message.new('foo')
#<Message:0x603bf0 @message="foo">
m.get_message
"foo"
m.set_message('bar')
"bar"
m.get_message
"bar"
```

However, this approach is almost never seen in code written by practicing Ruby. Instead, you'll see the preceding code example implemented like this:

```
class Message
  attr_accessor :message
  def initialize(m)
    @message = m
  end
end

m = Message.new('foo')
#<Message:0x5f3c50 @message="foo">
m.message
"foo"
m.message = "bar"
"bar"
m.message
"bar"
```

Aside from requiring less typing overall, this code is very clear and expressive, because it doesn't include the unnecessary get and set verbs. However, you might wonder how

to do data verification/protection with this approach.

If you need to add some special logic on write, you can still use `attr_reader` to provide the reading side of things and then use a custom method to handle the writing:

```
class Message
  attr_reader :message
  def message=(m)
    @message = m.dup
  end
end
```

On the other hand, if you need to do some handling on read but can afford to use the default writer, `attr_writer` is what you want:

```
class Message
  attr_writer :message
  def message
    @message.encode!("UTF-8")
  end
end
```

Of course, if you need both custom readers and writers, there is no need for the `attr_*` helpers. However, in this case, remember that unless there is a good reason to name things otherwise, use the methods `something()` and `something=()` instead of `get_something()` and `set_something()`.

=====

Ques 2: major difference b/w `attr_reader`, `attr_accessor`, `attr_writer`?

Ans: