

# Accessors in Ruby (Automatic)

- Accessors are a way to create getter and setter methods without explicitly defining them in a class

## 3 type of accessor methods

1) attr\_reader 2) attr\_writer 3) attr\_accessor

## attr\_reader (Attribute reader) method

- attr\_reader automatically generates a getter method for each given attribute (argument)

Getter method

```
def name  
  @name  
end
```

```
def email  
  @email  
end
```

```
def contact  
  @contact  
end
```

explicit method

auto generated getter method using attr\_reader

```
attr_reader :name, :email, :contact
```

initialize (arg1, arg2, arg3)  
↓  
n number of getter methods are produced

## ⇒ Things to remember

- You can only access those instance variables which you have passed to attr\_reader. (Video 3:16)
- You can only use attr\_reader in a class using it outside a class gives you no method error //

## attr\_writer method

attr\_writer automatically generates a setter method for each given attribute argument.)

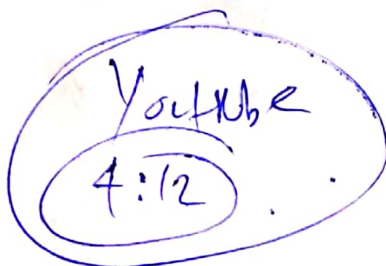
### Things to Remember:

- you can only update the value of these instance variables which you have passed to attr\_writer.
- you can only use attr\_writer in a class. Using attr\_writer outside a class will ~~be~~ give you a NoMethodError.

```
def name=(name)
  @name = name
end
```

```
def email=(email)
  @email = email
end
```

```
def contact=
def contact=(contact)
  @contact = contact
end
```



• you can use attr\_reader & attr\_writer one by one. i.e., if you want to only set value use attr\_writer if you want to update value use attr\_writer

⇒ attr\_writer : name, : email, : contact

↓  
if you write (no content)

attr\_writer : name, : email

↓  
then while updating

```
puts obj1.name
puts obj1.name = 'Alex'
puts obj1.email = '@'
puts obj1.contact = '5'
```

Prints error



~~Except method every thing is object in java~~

after-accessor (attr-read + attr-write)

combination of attr-read + attr-write