

- Basics -

> COMMENTS

```
# This is a comment
```

> ASSIGNMENTS

```
a = 42
```

> VARIABLES

```
a, b = [1 , 2]  
# a = 1, b = 2
```

- Built-in data types -

> STRINGS

```
'I'm a string'  
"with some #{RUBY(CODE)}"  
'without ruby code'
```

> NUMBERS

```
10  -42  123.45  1.2e-3
```

> SYMBOLS

```
:a_symbol  
# always with colon
```

> ARRAYS

```
['one' , 2 , 3.0]
```

```
access  
my_array[1]=2
```

> HASHES

```
# traditional syntax:  
{:key1 => value2, :key2 => value3}  
# or JSON syntax:  
{key1: value2, key2: value3}
```

```
access  
my_hash[:key1]=value2
```

- Blocks -

> BLOCKS

```
...do |block, arguments|  
    # ... do ... end for multiple lines of code  
end  
  
...{|argument, list| ... }  
# curly brackets for one-liner
```

- Methods and classes -

> DEFINING METHODS

```
def      name(list,of,arguments)  
    ...  
# body  
    return result  
# return value  
# „return“ is optional, if omitted  
# the last evaluated statement is the return value  
  
end
```

access

```
a.b(c) # or a.b_c  
# a is the object, b is the method,  
# c is the parameter  
# parentheses are optional
```

> DEFINITION OF CLASSES BY NAMESPACE

```
module MyModule  
    # CamelCase  
    class MyClass < ParentClass  
        # < means inherited  
    ...  
end
```

access

```
MyModule::MyClass
```

> CLASSVARIABLES

```
@@my_class_variable
```

> INSTANCEVARIABLES

```
@my_instance_variable
```