

Programming for Everybody

6. Hashes & Symbols

Arrays vs Hashes

Collections store a list of elements in a single variable, and can contain any data type (numbers, strings, booleans, other arrays, other objects, etc.)

Arrays are simple collections, where *each element can be accessed only by its positions* (called index)

Hashes are more complex collections, where *each element is labelled with a keyword, and can be accessed with that specific word* (the key)

Arrays vs Hashes

Arrays

```
my_array = [6, "a string", true, ["inside a sub-array!", 4, false], -12]
```

indexes

0

1

2

3

4

Hashes

```
my_hash = {
```

“cat” => “Garfield”,

“dog” => “Snoopy”,

“mouse” => “Mickey”

}

keys

Symbols

As we've seen until now, **strings** can be used as keys of our hashes (with the so-called *rocket syntax* 🚀)

But there is a more specific data type to represent the keys (labels) of an hash: **symbols**

Always remember that a symbol is different from a string!

Symbols can live outside of an hash, and can be assigned to variables and used as any other data type:

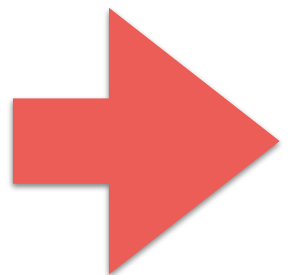
```
my_symbol = :this_is_a_symbol
```

Symbols as Hash Keys

From now on, when you deal with hashes, *try to not use strings as keys anymore*, together with the rocket syntax

Always try to use **symbols**, as they are a data type used mainly for keys, and Ruby gets also faster with symbol keys instead of string keys!

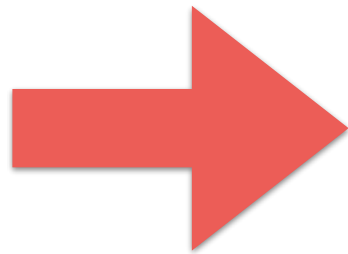
Also, instead of using the “rocket syntax” (it’s old syntax, and was replaced from Ruby 1.9), just place a colon after (in an hash) or before (outside the hash) the symbols



Symbols as Hash Keys

```
my_hash = {  
  "cat" => "Garfield",  
  "dog" => "Snoopy",  
  "more words" => "a value"  
}
```

```
cat_name = my_hash["cat"]
```



```
my_hash = {  
  cat: "Garfield",  
  dog: "Snoopy",  
  more_words: "a value"  
}
```

```
cat_name = my_hash[:cat]
```

No more rocket syntax!

Let's see some live
examples!

Thank you! :)