

3. ПЕРЕМЕННЫЕ И ПРИСВАИВАНИЕ

До сих пор, каждый раз, когда мы выводили с помощью `puts` строку или число, всё, что мы выводили, исчезало. Я имею в виду, если мы хотели напечатать что-то дважды, мы должны были вводить это дважды с клавиатуры:

```
puts '...ты можешь сказать это снова...'
puts '...ты можешь сказать это снова...'

...ты можешь сказать это снова...
...ты можешь сказать это снова...
```

Вот было бы прекрасно, если бы мы могли просто ввести что-то один раз и потом обращаться к нему... сохранить его где-нибудь. Ну, конечно же, мы можем это сделать, иначе я бы не заговорил об этом!

Чтобы сохранить строку в памяти вашего компьютера, нам нужно дать строке какое-то имя. Программисты часто говорят об этом процессе как о присваивании, а имена они называют переменными. Имя переменной — это просто любая последовательность латинских букв и цифр, но первый символ должен быть буквой в нижнем регистре. Давайте снова испробуем нашу последнюю программу, но на этот раз я дам строке имя `myString` (хотя я с таким же успехом мог назвать её `str` или `myOwnLittleString` или `henryTheEighth`).

```
myString = '...ты можешь сказать это снова...'
puts myString
puts myString

...ты можешь сказать это снова...
...ты можешь сказать это снова...
```

Каждый раз, когда вы пытаетесь что-нибудь сделать с `myString`, программа вместо этого делает это со строкой `'...ты можешь сказать это снова...'`. Вы можете думать о переменной `myString` как об "указателе на" строку `'...ты можешь сказать это снова...'`. А вот немного более интересный пример:

```
name = 'Патриция Розанна Джессика Милдред Оппенгеймер'
puts 'Меня зовут ' + name + ' .'
puts 'Ого! ' + name + ' - это правда длинное имя!'

Меня зовут Патриция Розанна Джессика Милдред Оппенгеймер.
Ого! Патриция Розанна Джессика Милдред Оппенгеймер - это правда длинное имя!
```

Кроме того: так же просто, как мы можем *присвоить* объект переменной, мы можем *переприсвоить* другой объект этой же переменной. (Вот почему мы называем их переменными: потому что то, на что они указывают, может изменяться.)

```
composer = 'Моцарт'
puts composer + ' был "сенсацией" в своё время.'
composer = 'Бетховен'
puts 'Но мне лично больше нравится ' + composer + '.'
```

Моцарт был "сенсацией" в своё время.
Но мне лично больше нравится Бетховен.

Конечно, переменные могут указывать на объект любого типа, а не только на строки:

```
var = 'ещё одна ' + 'строка'  
puts var  
var = 5 * (1+2)  
puts var
```

```
ещё одна строка  
15
```

Фактически переменные могут указывать почти на всё... кроме других переменных. А что же произойдёт, если мы попробуем?

```
var1 = 8  
var2 = var1  
puts var1  
puts var2  
puts ''  
var1 = 'восемь'  
puts var1  
puts var2
```

```
8  
8  
восемь  
8
```

Сначала, когда мы попытались сделать, чтобы переменная **var2** указывала на **var1**, она в действительности стала вместо этого указывать на **8** (на то же число, на которое указывала переменная **var1**). Затем мы сделали так, чтобы **var1** стала указывать на **'восемь'**, но поскольку **var2** никогда не указывала на **var1**, она продолжает указывать на **8**.

Ну а теперь, когда у нас есть переменные, числа и строки, давайте научимся, собирать их все вместе!