**Типы переменных**

[Java Syntax Pro](https://javarush.ru/quests/QUEST_JAVA_SYNTAX_PRO)

[Уровень 1](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX_PRO&level=1), Лекция 4



**1. Объявление переменных**

Давайте еще раз разберем, как создавать переменные.

Чтобы создать переменную, нужно написать такую команду: тип имя;.

Примеры:

|  |  |
| --- | --- |
| **Команда** | **Пояснение** |
| String s; | Создается переменная s типа String. Такая переменная может хранить текст. |
| int x; | Создается переменная x типа int. Такая переменная может хранить в себе целые числа. |
| int a, b, c;  int d; | Создаются переменные a, b, c, d типа int. Такие переменные могут хранить целые числа. |

Важно!

Нельзя создать две переменные с одинаковыми именами в одном методе. А вот в разных методах — можно. Это как коробки, стоящие в разных домах.

Еще есть ограничения на имя переменной. С одной стороны, оно может быть любым, с другой – оно не может содержать пробелов, символов +, - и т.д. Лучше всего в имени переменной использовать только латинские буквы и цифры.

Обратите внимание, что в **языке Java** играет роль, какие буквы вы пишите — **большие** или **маленькие**. int a — это не то же самое, что и Int a.

Кстати, в Java можно одновременно создавать переменную и присваивать ей значение. Это экономит время и место:

|  |  |
| --- | --- |
| **Компактный код** | **Длинный код, эквивалентный коду слева** |
| int a = 5; | int a;  a = 5; |
| int b = 6; | int b;  b = 6; |
| int c = 7; | int c;  c = 7; |
| int d = c + 1; | int d;  d = c + 1; |
| String s = "I'm Amigo"; | String s;  s = "I'm Amigo"; |

Так гораздо компактнее и понятнее.

Ну а раз с созданием переменных мы уже разобрались, давайте немного познакомимся с двумя самыми используемыми типами в языке Java. Это типы int (целые числа) и String (текст/строки).

**2. Тип int**

В переменной типа int можно хранить целые числа. Также над типом int можно выполнять различные операции: сложение, вычитание, умножение, деление и другие. Примеры:

|  |  |
| --- | --- |
| **Код** | **Пояснение** |
| int x = 1;  int y = x\*2;  int z = 5\*y\*y + 2\*y + 3; | x будет равен 1 y будет равен 2 z будет равен 20+4+3, будет равен 27 |
| int a = 5;  int b = 1;  int c = (a-b) \* (a+b); | a будет равно 5 b будет равно 1 c будет равно 4\*6, будет равно 24 |
| int a = 64;  int b = a/8;  int c = b/4;  int d = c\*3; | a будет равно 64 b будет равно 8 c будет равно 2 d будет равно 6 |

**3. Тип String**

Тип String позволяет хранить текстовые строки.

Чтобы задать какую-то текстовую строку, в Java надо написать текст строки, а с обеих сторон поставить двойные кавычки. Пример:

|  |  |
| --- | --- |
| **Код** | **Пояснение** |
| String s = "Amigo"; | s будет содержать текст Amigo |
| String s = "123"; | s будет содержать текст 123 |
| String s = "Bond 007"; | s будет содержать текст Bond 007 |

Выглядит несложно, правда? Ну если так, тогда вот еще один интересный факт.

Строки в Java можно склеивать с помощью знака плюс — +. Пример:

|  |  |
| --- | --- |
| **Код** | **Пояснение** |
| String s = "Amigo" + " the best"; | s будет содержать Amigo the best |
| String s = ""; | s будет содержать пустую строку — вообще без символов. |
| int x = 333;  String s = "Amigo" + x; | s будет содержать Amigo333 |

Обратите внимание на последний пример: мы сложили строку и число. Тут тоже все просто: число будет преобразовано в строку, затем две строки будут склеены. При сложении строк и чисел, всегда получается строка.

**4. Вывод переменной на экран**

Вроде все так очевидно и элементарно. Ну тогда, может, вы сразу догадаетесь, какой командой можно вывести переменную на экран?

На самом деле все просто. Чтобы вывести что-то на экран, мы используем команду System.out.println(), куда в качестве параметра передаем то, что мы выводим.

|  |  |
| --- | --- |
| **Код** | **Вывод на экран** |
| System.out.println("Amigo"); | Amigo |
| System.out.println("Ami" + "go"); | Amigo |
| String s = "Amigo";  System.out.println(s); | Amigo |
| String s = "Am";  System.out.println(s + "igo"); | Amigo |

Надеюсь, теперь стало немного понятнее. А правильно вы все поняли или нет, мы сейчас проверим. Практика — критерий истины: только с помощью практики можно проверить, хорошо ли вы во всем разобрались.

