



Программирование на Python

Презентация занятия

Изучение возможностей и синтаксиса Python: Переменные и выражения.

2 занятие



инжинириум®

МГТУ им. Н.Э. Баумана

2019

Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

Переменные — это удобный способ хранения данных (чисел, текста, списков с числами или символами и так далее) в программе, и они пригодятся нам для решения самых разных задач.

int - используется для целых чисел

```
fred_age = 13
```

```
print(type(fred_age), fred_age) #Выдаст тип переменной
```

```
<class 'int'> 13
```

Функция type() из этого примера возвращает тип переменной

```
john_age = 25
```

```
fred_age = john_age
```

```
print(fred_age, john_age) # Что выдаст программа?
```

Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

float – используется для дробных чисел. Целая и дробная части отделяются точкой

```
num = 150.2
```

```
print(type(num), num)
```

```
<class 'float'> 150.2
```

Из-за представления чисел в компьютере вещественные числа неточны, и это может привести к ошибкам:

```
print(0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1)
```

```
0.9999999999999999
```

Встроенная функция int() предназначена для преобразования чисел с плавающей точкой и строк в целые числа.

```
num = 150.7
```

```
print(type(num), num, "---->" int(num))
```

```
<class 'float'> 150.7 ----> 150 # Потеряли дробную часть
```



Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

В оболочке Python можно умножать, складывать, вычитать и делить числа, делить на цело, считать остаток от деления, а также возводить число в степень

```
print(5 + 10)
```

15

```
print(2 ** 4) # две звёздочки означают возведение в степень
```

16

```
print(37 / 3) # один слэш — это деление с ответом дробью
```

12.333333333333334

```
print(37 // 3) # два слэша считают частное от деления нацело
```

12

```
print(37 % 3) # процент считает остаток от деления нацело
```

1

```
print(((5 + 30) * 20) / 10) # Что выведет программа?
```



Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

Строка – базовый тип представляющий из себя **неизменяемую последовательность** символов

```
example_string = "Курс по Python"
```

```
print(example_string)
```

Курс по Python

```
print(type(example_string))
```

<class 'str'>

Как объединить несколько строк в одну?

```
example_string + 'в Инжинириуме'
```

Курс по Python в Инжинириуме

Важно помнить, что строки - неизменяемый тип.

```
num_inpt = input() # Введем число 55
```

```
print(type(num_inpt), num_inpt, "---->" int(num_inpt) + 2)
```

<class 'str'> 55 ----> 57

```
print(num_inpt + 2) # Ошибка
```



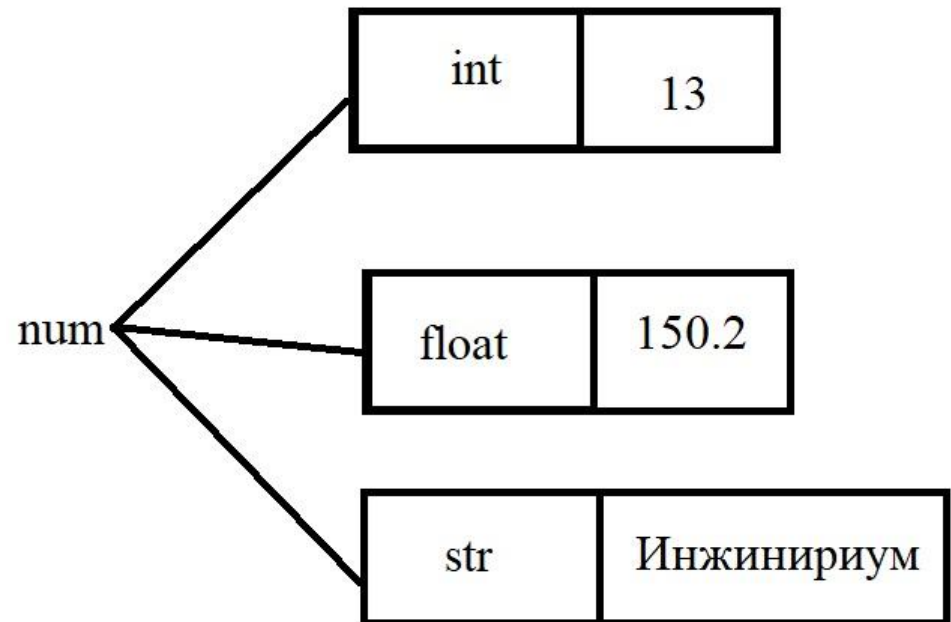
Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

Когда мы присваиваем какое-то значение переменной, то это значение сохраняется в отдельной ячейки, которая состоит из двух значений. Это тип данных и само значение переменной, а переменная - ссылка на эту ячейку данных

```
>>> num = 13
```

```
>>> num = 150.2
```

```
>>> num = "Инжинириум"
```



Тема: Изучение возможностей и синтаксиса Python: Переменные и выражения.

Логический тип представлен двумя постоянными значениями **False** и **True**. Значения используются для представления истинности и являются результатом использования операторов сравнения и логических операций

Значение **True**

```
my_bool_true = True  
my_bool_true = bool(10)  
my_bool_true = bool('some')
```

Значение **False**

```
my_bool_false = False  
my_bool_false = bool(0)  
my_bool_false = bool("")  
my_bool_false = bool()
```

Тема: Изучение возможностей и синтаксиса Python:

Операторы сравнения

Меньше — условие верно, если первый операнд меньше второго: **<**

Больше — условие верно, если первый операнд больше второго: **>**

Меньше или равно: **<=**

Больше или равно : **>=**

Равенство. Условие верно, если два операнда равны: **==**

Неравенство. Условие верно, если два операнда неравны: **!=**

Операторы сравнения в Питоне можно объединять в цепочки (в отличие от большинства других языков программирования, где для этого нужно использовать логические связки), например, `a == b == c` или `1 <= x <= 10`.

Возвращает **True** или **False**



Тема: Изучение возможностей и синтаксиса Python:

Операторы сравнения

В Питоне существуют стандартные логические операторы: логическое И, логическое ИЛИ, логическое отрицание.

Логическое И является бинарным оператором и имеет вид **and**.

Оператор “логическое ИЛИ” имеет вид **or**.

Логическое НЕ (отрицание) является унарным (то есть с одним операндом) оператором и имеет вид **not**, за которым следует единственный операнд.

```
a = 10
```

```
b = 5
```

```
if a % 10 == 0 or b % 10 == 0:
```

```
    print('YES')
```

```
else:
```

```
    print('NO')
```

```
YES
```