



# ЛЕКЦИЯ №2

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ

Преподаватель: Борисов Денис Дмитриевич

# НЕБОЛЬШОЕ ПРЕДИСЛОВИЕ.

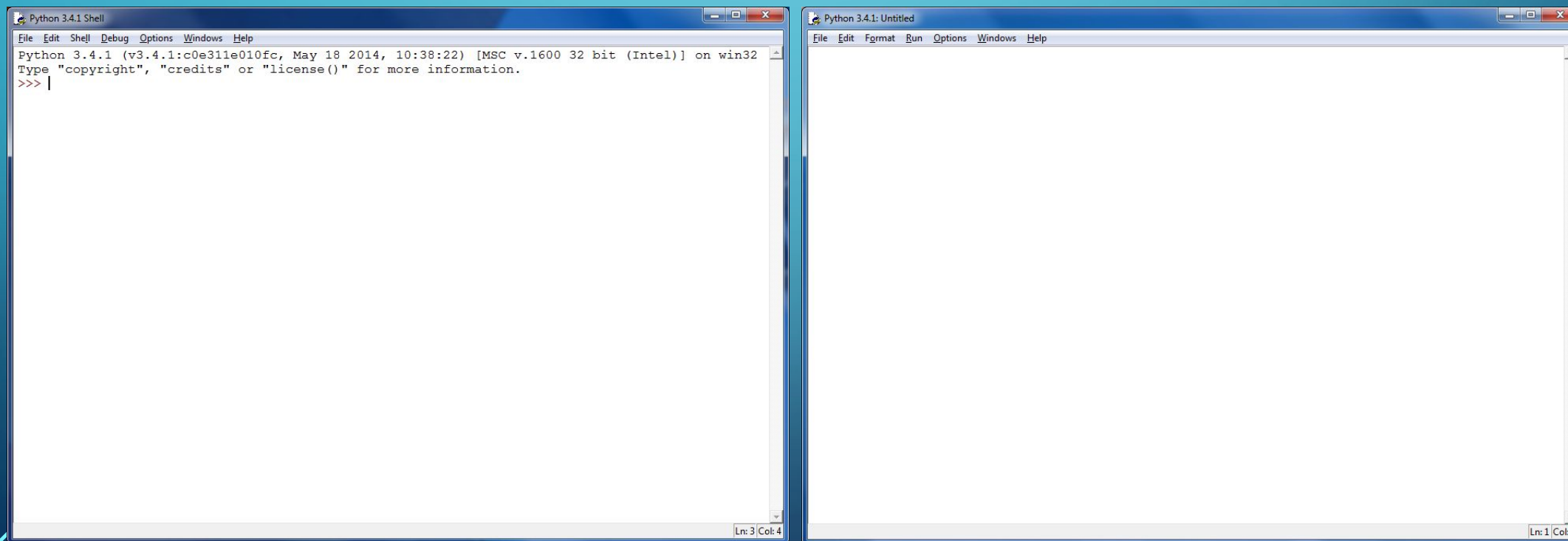
- Перед началом хотелось бы кое-что рассказать про сам язык Python.
- Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Разработчики языка Python придерживаются определённой философии программирования, называемой «The Zen of Python» («Дзен Питона»). Её текст выдаётся интерпретатором Python по команде `import this` (работает один раз за сессию). Автором этой философии считается Тим Петерс (Tim Peters).
- Текст философии:
  - Красивое лучше, чем уродливое.
  - Явное лучше, чем неявное.
  - Простое лучше, чем сложное.
  - Сложное лучше, чем запутанное.
  - . . . .

# 1. РАБОТА В ИНТЕРАКТИВНОМ РЕЖИМЕ

В Python написание программ возможно в двух режимах:

В ИНТЕРАКТИВНОМ РЕЖИМЕ

В СКРИПТОВОМ РЕЖИМЕ



Интерактивный режим языка Python позволяет писать программы построчно. Это очень удобно когда нужно быстро проверить работу какого-нибудь короткого кусочка кода. (Введем такое обозначение: `>>>` <какая-то команда>, для того, чтобы я каждый раз не вставлял скриншоты). Например в прошлый раз мы писали программу, которая выводила сообщение. Эту же самую программу можно сделать в IP.

Попробуйте написать в IP следующее:

```
>>> print('Hello new day!')
```

Вы должны увидеть следующий вывод:

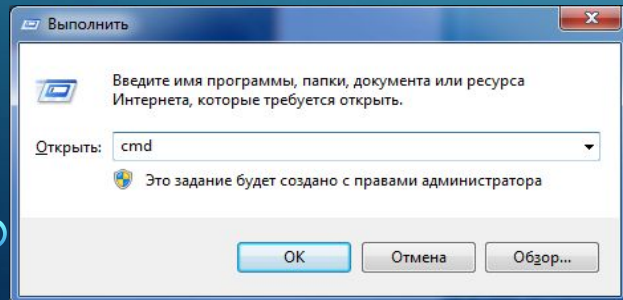
```
Hello new day!
```

В будущих лекциях вы узнаете, что в интерактивном режиме можно запускать не только строки кода, но и целые циклы, функции, классы и т.д.

Поэкспериментируйте с выводом. Поменяйте сообщение в кавычках на то, что вы бы хотели вывести.

Начать работать в IP можно либо, запустив IDLE, либо с помощью командной строки. Делается это так.

Нажмите Win + R и введите cmd. Запустится командная строка. В командной строке наберите `python`. Теперь вы можете пользоваться IP.

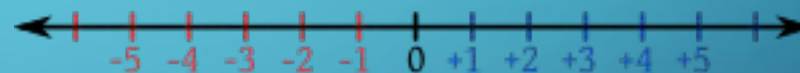


## 2. ВСТРОЕННЫЕ ТИПЫ ДАННЫХ ЯЗЫКА PYTHON. И ОПЕРАЦИИ НАД НИМИ. ЧИСЛА.

Все с чем мы будем иметь дело в Python является объектом. Типы объектов могут быть либо встроенными либо описанными с помощью классов. Я бы хотел рассказать о встроенных типах. Основные встроенные типы данных Python:

$\mathbb{Z}$

- Числа
- Строки
- Списки
- Кортежи
- Множества
- Словари
- Файлы



Как видно из заголовка, в этой лекции мы будем изучать только числа.

**Целые числа** (`int`, от английского слова *integer*, означающее «целое число»). Как вы помните из математики множество целых чисел обозначается символом  $\mathbb{Z}$  и получается из множества натуральных чисел 1, 2, 3, 4, ...,  $+\infty$  добавлением нуля и отрицательных чисел.

Над целыми числами в Python можно совершать следующие операции (таблица справа).

Обратите внимание, что нумерация начинается не с 1, а с нуля. **В программировании принято начинать нумерацию с нуля**, чтобы в дальнейшем не совершать ошибок. С этого момента я буду всегда начинать любую нумерацию с нуля (принятому в программировании обычаю начинать нумерацию с нуля =).

Операции с 0 по 8 являются операциями из школьной математики. Ничего сложного нет. Операции 0, 1, 2, 3, 4, 5, 6, 7, 8 нужно, достаточно в IP с каждой из них поэкспериментировать.

```
>>> 4 + 6
```

```
10
```

```
>>> -4 + 6
```

```
2
```

```
>>> 17 - 20
```

```
-3
```

```
>>> 13 * 8
```

```
104
```

```
>>> 100 / 2
```

```
50.0
```

Обратите внимание на то, что при делении получается число не целое, а вещественное. Даже, когда числах мы поговорим позже.

```
>>> 2 ** 10
```

```
1024
```

Вообще можете попробовать возвести 1024 в 1024 степень! Посмотрите, что у вас получится.

```
>>> abs(-989)
```

```
989
```

Операция // отличается от операции / тем, что она возвращает целую часть числа, откидывая дробную. К примеру,

№	Операция	Описание
0	$x + y$	Сложение
1	$x - y$	Вычитание
2	$x * y$	Умножение
3	$x / y$	Деление
4	$x // y$	Получение целой части от деления
5	$x \% y$	Остаток от деления
6	$-x$	Смена знака числа
7	$abs(x)$	Модуль числа
8	$x ** y$	Возведение в степень
9	$x   y$	Побитовое ИЛИ (OR)
10	$x \& y$	Побитовое И (AND)
11	$x \wedge y$	Побитовое исключающее или (XOR)
12	$x \ll y$	Битовый сдвиг влево
13	$x \gg y$	Битовый сдвиг вправо
14	$\sim x$	Инверсия битов

## Вещественные числа или числа с плавающей точкой (`float`, float – поплавок).

В математике вещественные числа обозначаются символом

Наглядно понятие вещественного числа можно представить себе при помощи числовой прямой. Если на прямой выбрать направление, начальную точку и единицу длины для измерения отрезков, то каждому вещественному числу можно поставить в соответствие определённую точку на этой прямой, и обратно, каждая точка будет представлять некоторое, и притом только одно, вещественное число. Например числа  $0.8$ ,  $1$ ,  $8.9999$ ,  $-1$ ,  $-23.09$  все это вещественные числа.



Числа типа `float` поддерживают все те же операции, что и числа типа `int`, кроме 9 – 14 (т.е. побитовых операций). Попробуйте произвести доступные операции над различными числами, как мы делали это с целыми числами. И попробуйте вызвать функцию `type(228.0)`. Результат оставьте в комментариях к посту с лекцией. Про комплексные числа в рамках данного курса возможно будут рассмотрены, но только в качестве дополнительного материала. С соответствующими задачами повышенной сложности. А теперь перейдем к изучению переменных – одному из самых основных понятий программирования.

# 3. ПЕРЕМЕННЫЕ

Когда нам нужно обработать какую-то информацию, возникает вопрос как ее хранить? Для хранения различных данных в языках программирования есть *переменные*. Переменная представляет собой зарезервированное место в оперативной памяти для временного хранения данных. Переменная характеризуется *именем*, *типом* и *значением*.

Синтаксис объявления переменных в языке Python показан на рисунке.

Имя переменной должно начинаться с латинской буквы в любом регистре или символа подчеркивания «\_», после чего в имени можно использовать буквы, цифры и `_`. Значение – это те данные, которые мы бы хотели хранить. На месте два два восемь может быть любой объект Python. Python чувствителен к регистру, это значит, что она отличает, к примеру, переменные `p` и `P`! Особенностью языка Python является, то что он динамически типизирован. Т.е. на месте два два восемь может стоять совершенно любой объект, а питон сам определит его тип и выделит под этот объект необходимый объем памяти.





- Упражнение №1:

- Попробуйте в IP создать переменные которые показаны ниже.
- Затем выведите тип переменной, а затем и значение переменной.
- Напоминаю, что для того, чтобы узнать тип объекта можно использовать функцию `type()`, а для вывода функцию `print()` Это упражнение нужно делать в скриптовом режиме. Для проверки отправьте ссылку на код в [pastebin](#).

```
s = 'программирование – это круто'
t = (1, 2, 3, 4)
l = [555, 666, 777]
d = {'first_name' : 'Михет', 'last_name' : 'Бозик'}
f = 5.5
i = 6
```

В общем случае у функции `print()` такой синтаксис: `print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)`

`*objects` означает, что функция принимает неограниченное числа аргументов.

`sep=' '` означает, что объекты, которые мы выводим будут разделяться строкой ' ', на ее месте может быть абсолютно любая строка

`end='\n'` означает какой строкой нужно окончить вывод, на месте '\n' также может быть любая строка

`file=sys.stdout` это поток вывода, об этом поговорим позже

`flush` я не совсем понимаю, что такое `flush` =)

## • Упражнение №2:

Давайте с вами создадим программу, которая находит корень любого линейного уравнения. Как вы помните из математики общий вид линейного уравнения от одной переменной выглядит вот так:

$$y = kx + b$$

Соответственно корень уравнения можно найти так  $x = \frac{y-b}{k}$  (\*)

Предполагается, что  $k$ ,  $b$  и  $y$  вводятся пользователем. Сейчас я объясню как нужно считывать данные от пользователя. Для этого существует функция `input()`. Попробуйте в IP сделать следующее:

```
>>> x = input()
```

Здесь у Вас появится ожидание ввода. Введите что угодно:)

Потом то, что вы ввели выведется с новой строки в кавычках. Т.е. `input()` считывает данные пользователя и возвращает строку. Это важно запомнить, так как многие начинающие программисты на этом часто совершают ошибку. Чуть ниже я объясню, что за ошибку можно совершить.

Еще функция `input()` может иметь аргумент – строку.

```
>>> x = input('Enter message: ')
```

Теперь у вас появится самое настоящее приглашение ко вводу. Это нужно потому, что User может быть не знаком с алгоритмом и реализацией Вашей

программы. Вернее там обычно и бывает. Поэтому нужно ему сообщить, что мы вообще хотим от него получить.

Алгоритм нашей программы:

1. Считать  $y$
2. Считать  $b$
3. Считать  $k$

*порядок считывания не важен*

4. Подставив значения переменных в формулу \* получить результат, сохранив его в переменную.
5. Вывести результат.

Делаем программу в скриптовом режиме.

```
y = input('Enter y :')
b = input('Enter b :')
k = input('Enter k :')
res = (y-b)/k
print('Result :', res)
```

Вот и вся программа!) всего 5 строчек! На самом деле количество строк можно еще уменьшить, при этом даже не теряя читабельность кода, но нам сейчас это ни к чему. Отправьте код вашей программы в комментарии к посту с лекцией.

# ДОМАШНЕЕ ЗАДАНИЕ

- Жду от вас две ссылки:
  - 1 на упражнение №1
  - 2 на упражнение №2

Желаю удачи! =)