Программа Python.
2021-2022 год.

Лекция №2 Группа К4:

Язык Python характеризуют краткие синтаксические конструкции. Python любят за то, что на нем можно писать очень короткий код.

Короткий код – это не означает короткую программу. Это означает, что код на Python можно писать очень компактным. Текст самой программы Python гораздо меньше, чем на других языках программирования.

Также язык Python характеризует большое количество стандартных библиотек - их мы будем изучать немного позднее.

Если коротко, что такое **библиотеки в языке Python:**

Сам язык обычно очень маленький. Он содержит в себе только основные вещи, без которых вообще невозможно программировать.

На этом языке потом пишутся **различные функции**, которые помогают удобно работать с интернетом, с графикой, с локальными сетями и т.д. **Эти функции или набор функций называются библиотеками.** Программист подбирает функции (библиотеки) под конкретные задачи. Библиотеки могут быть стандартные, а могут быть и ваши собственные. Вот в Python очень большое количество стандартных библиотек, которые могут решить практически любую задачу при создании программ. Есть, конечно, и внешние библиотеки, которые созданы программистами, они доступны для скачивания с интернета, и они могут решать задачи, не входящие в стандартные библиотеки.

Что такое программа на Python?

Программа это текст, который содержит последовательность инструкций (команд) и выполняются инструкции (команды) тоже последовательно при запуске программы.

На первом этапе изучения языка Python, мы будем создавать консольные приложения. Это текстовые программы.

Почему? Где же графика или игры, спросите вы?

Потому, что сложность программирования это не картинки и визуальное оформление. Сложность программирования заключается в продумывании логики программы.

Поэтому, чтобы не отвлекаться и не тратить время на прорисовку графики, мы будем работать с текстовой информацией, которая внутри программы.

Что же может быть внутри консольной текстовой программы?

Рассмотрим оператор **print():**

Оператор print() - выводит (печатает) данные из круглых скобок на экран. Скобки оператора print означают, что оператор является функцией языка Python.

Пример:
Есть математическая функция: $f(x) = x^2$
Если передать в функцию f x=2: f(2) = 2*2 = 4
В языках программирования есть функции, которые ничего не вычисляют, а выполняют различные действия (команды).

Print() умеет печатать различные объекты.
К примеру, умеет печатать, простой объект - целое число (int) - сокращение от слова integer (англ.) - целое число (рус.)

Создаем файл в среде Wing IDE - Файл - Новый.

```
print (25)          = 25
```

```
print (2+2)        = 4
print (3-1)        = 2
print (2*2)        = 4
print (2**3)       = 8
print (5//2)       = 2
print (5/2)        = 2.5
print (5%2)        = 1
print (abs(-3))    = 3
```

(Ctrl +, Ctrl - можно менять масштаб текста в редакторе)

Запускаем программу при помощи жука! Не используем для запуска зеленую стрелку.

При первом запуске попросит программа сохранить файл.

Далее сохранение будет автоматическим, перед запуском выполнения программы (при нажатии на жука).

Очень важный момент!

В консоли Wing IDE (вкладка Python Shell) можно выполнять не только команды print(), но и сразу вычислять выражения 2+2!

В программе, где пишется код программы, выражение 2+2 не сработает. При запуске программы (файла) выражение без команды print() не отобразится на экране.

Теперь изучим оператор input():

Оператор input() - вводит данные в программу.

Это функция ждет, пока пользователь введет значение и нажмет enter. Результат этой функции это значение, которое ввел пользователь.

Конечно, чтобы мы могли записать значение полученное функцией input() и использовать его в дальнейшем в программе, необходимо это значение записать в переменную.
name = input(), где знак (=) это оператор присваивания.

Важно!
print(name) или print("name") - это разные понятия!
в первом случае это печать значения переменной, во втором случае это просто печать слова - name

Примечание: У Python есть своя область памяти. В памяти живут объекты. Числа или текст и т.д. Так же есть таблица в памяти, где живут все имена переменных. В этой таблице каждому имени переменной соответствует адрес ячейки памяти, где лежит значение этой переменной. Поэтому, когда выполняется команда name = input(), то имя переменной находится в таблице переменных и уже от имени переменной в таблице идет переход по ссылке на ячейку памяти, где живет значение переменной name. Получается, имена переменных лежат в отдельной таблице, а значения этих переменных лежат в отдельных ячейках памяти, согласно адресу пути, который прописывается в таблице переменных.

input() - запускаем программу. Во вкладке Debug I/O видим мигающий курсор - программа ждет ввода данных от пользователя программы.

Вводим данные и нажимаем enter. Программа выполнилась и закончила свое выполнение – жучек опять активен.

Перепишем немного код, чтобы отобразить введенные данные пользователем программы:

```
m = input()
print(m)
```

Теперь можно собрать все наши полученные знания и создать программу, которая узнает ваше имя и приветствует вас по имени:

```
print('Здравствуйте, как Вас зовут?')
name = input()
print('Добрый день,', name)
```

Лекция №3 (К1)

По умолчанию данные вводимые пользователем в функцию input() программа определяет как текст!

Пример:

```
m = input('Введите целое число от 1 до 10: ')
print(m + m)
```

Результат: 1010, если мы ввели число 10

И какие цифры мы не вводили бы, мы не добьемся от программы, чтобы она их арифметически сложила между собой!

Потому, что текст складывается вот так:

```
print('A' + 'B' + 'C') - ABC
```

Как научится вводить цифры в input() и производить математические операции с цифрами?

Для перевода символа цифры (строки) в число можно использовать встроенную функцию **int()**. Функция принимает любую строку и переводит ее в цифру (переводит из строкового типа в числовой тип данные)

перепишем нашу программу:

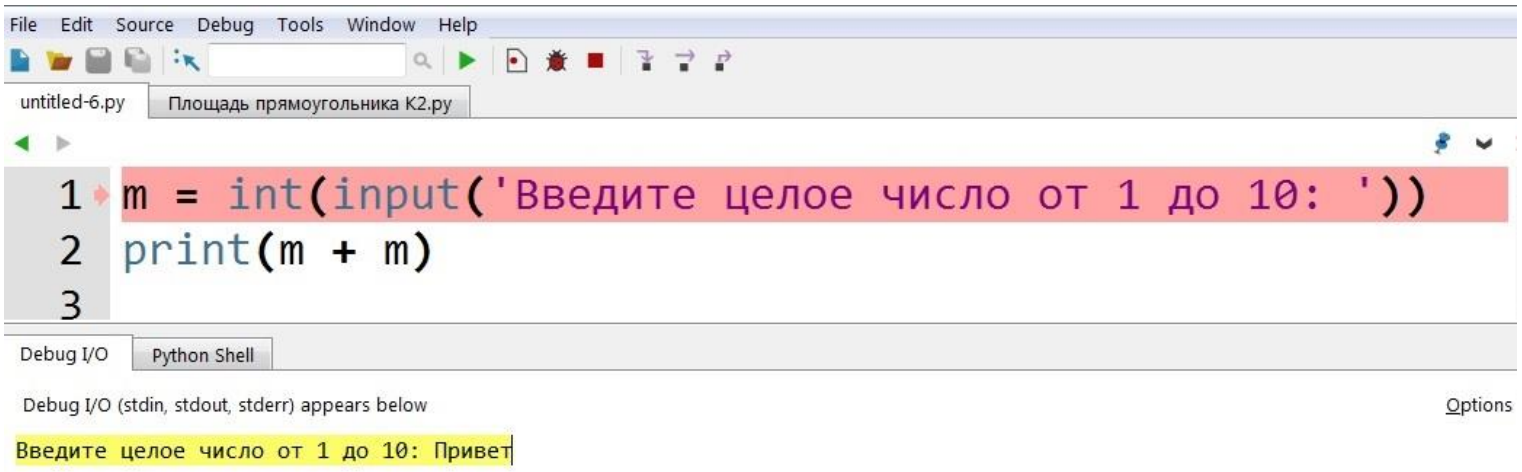
```
m = int(input('Введите целое число от 1 до 10: '))
print(m + m)
```

Результат: 20, если мы ввели число 10

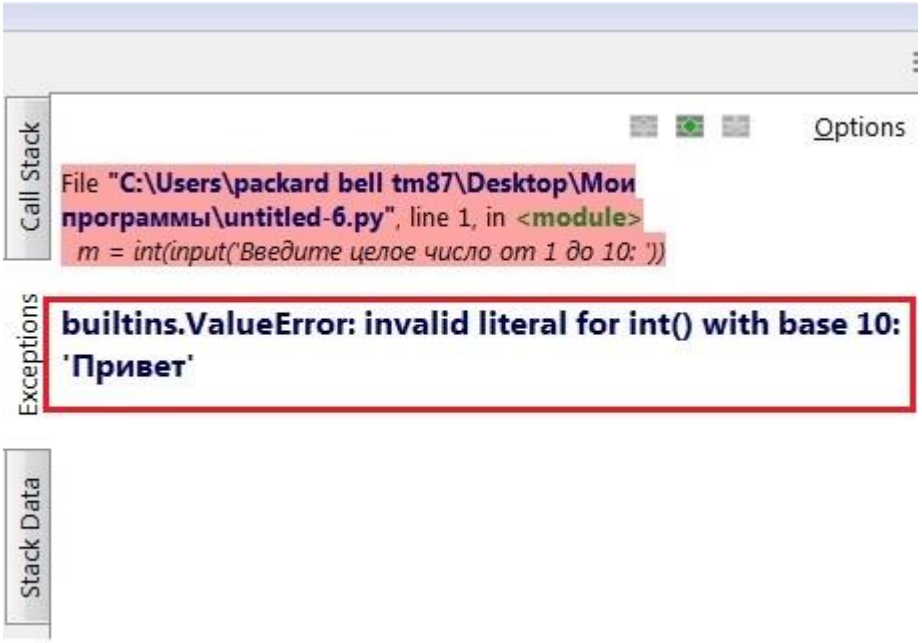
Отлично, теперь числа складываются арифметически.

Если ввести в нашей программе не цифру, а текст! То получим ошибку:

Введите целое число от 1 до 10: Привет



Получаем ошибку в программе:

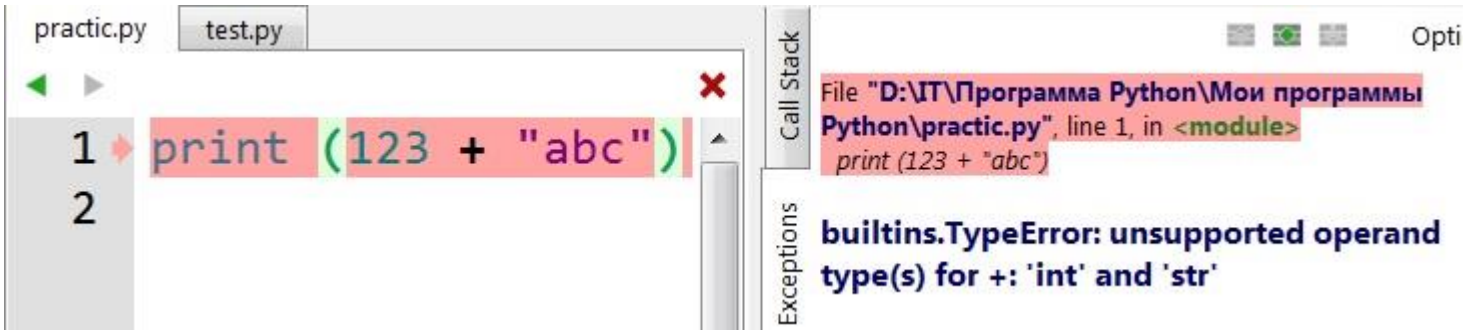


builtins – означает встроенная ошибка

Если попробовать сложить текст и цифру?

Получаем ошибку! Можно только объединить текст. Не цифру с текстом.

Аналогично ошибка будет если сложить первой цифру с текстом



Правило:
Складываем текст с текстом, либо цифру с цифрой.

Python автоматически не умеет приводить при объединении текста с числом, число в текст. Python выдаст ошибку.

Далее: Что будет если написать такую программу?

```
m = input() + input()
print (m)
```

Получит одно значение input(), потом на другой строчке получит значение второго input() и затем сложит эти значения и покажет значение m на экране.

Пример:

```
m = "Привет!"
n = "Меня зовут"
c = "Вася!"
print(m, n, c,)
```

результат: Привет! Меня зовут Вася!

Так же можно вывести переменные в строку без пробелов:

```
m = 'Привет!'
n = 'Меня зовут'
c = 'Вася!'

print(m+n+c)
```

результат: Привет!Меня зовутВася!