Программирование на Python

Урок №10

План урока:

1. Зачем нужны функции
2. Что такое функци
3. Функции в Python
4. Задачи
5. Контрольные вопросы

Результат:

```
def summa(x, y):
    summa = x + y
    return summa
x = summa(5, 6)
print(x)
```

Зачем нужны функции:

Как мы могли заметить ранее, многие конструкции в программировании очень часто повторяются. К примеру, перебор всех элементов списка или ввод данных в список. Такие конструкции “раздувают” программу, делают ее малочитабельной и не наглядной. К примеру, мы хотим немного усложнить сортировку. но программа сортировки уже и так довольно большая, а если мы начнем ее усложнять, то просто запутаемся. Это одна из самых главных проблем программирования — написать очень сложную программу так, чтобы она выглядела простой. И один из основных инструментов в арсенале программиста для упрощения кода являются функции.

Что такое функция:

Многие из ежедневно сталкиваются с функциями. К примеру, можно зайти в настройки телефона и там обязательно будет раздел, который содержит слово “Функции”. Мы часто слышим слово многофункциональность по отношению к машинам, которые могут выполнять множество задач. Или у аппарата по продаже шоколадок есть функция выдать товар и функция вернуть сдачу. Но что их всех объединяет? Мы не знаем как они устроены внутри, но точно знаем что будет после выполнения функции. К примеру, мы не знаем как устроена соковыжималка внутри. Но мы точно знаем, что если положить в нее два апельсина, то мы получим апельсиновый сок н выходе. Так же можно рассматривать и функцию в программировании — она принимает в себя некоторый набор объектов и выдает что-то на выходе. На самом деле, мы уже сталкивались с функциями в питоне. И именно в этом их прелесть — мы даже не подозревали что это функции и не представляли, как они устроены внутри, но свободно ими пользовались. К примеру, input() — функция. int() — тоже функция, а int(input()) — функция внутри функции. len() — тоже функция.

Функции в Python:

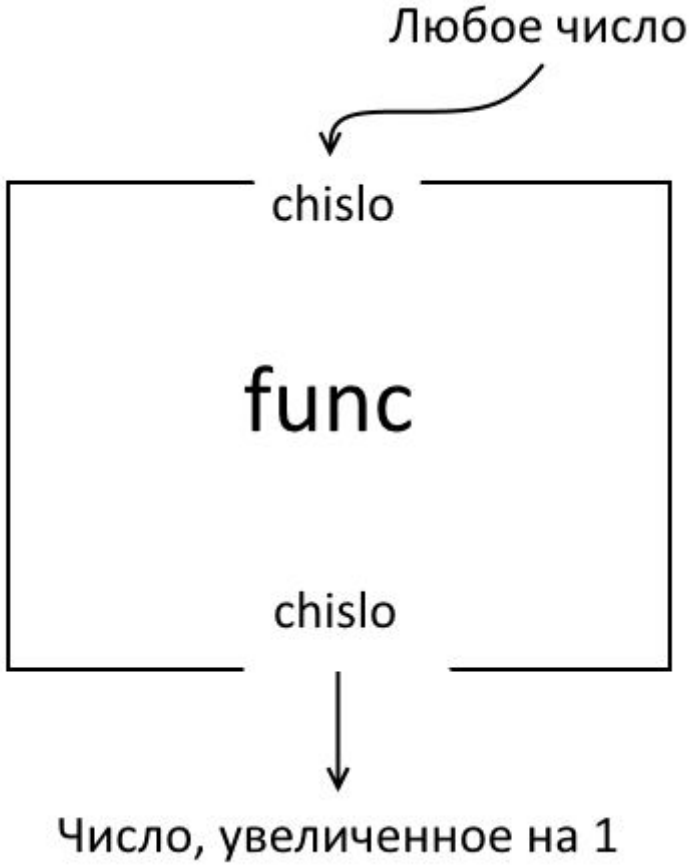
Ключевое слово для функции **def**. Причем, в отличии от всех остальных команд, функции пишутся ДО начала написания самой программы. Попробуем написать простую функцию. Она будет принимать число и увеличивать его на 1.

```
def func(chislo):
    chislo = chislo + 1
    return chislo
```

Если запустить эту программу, то ничего не произойдет. Почему? Создание функции можно сравнить с добавлением новой кнопки в телефоне. Она есть, но пока мы ничего не нажали — ничего не произойдет. Попробуем вызвать нашу функцию.

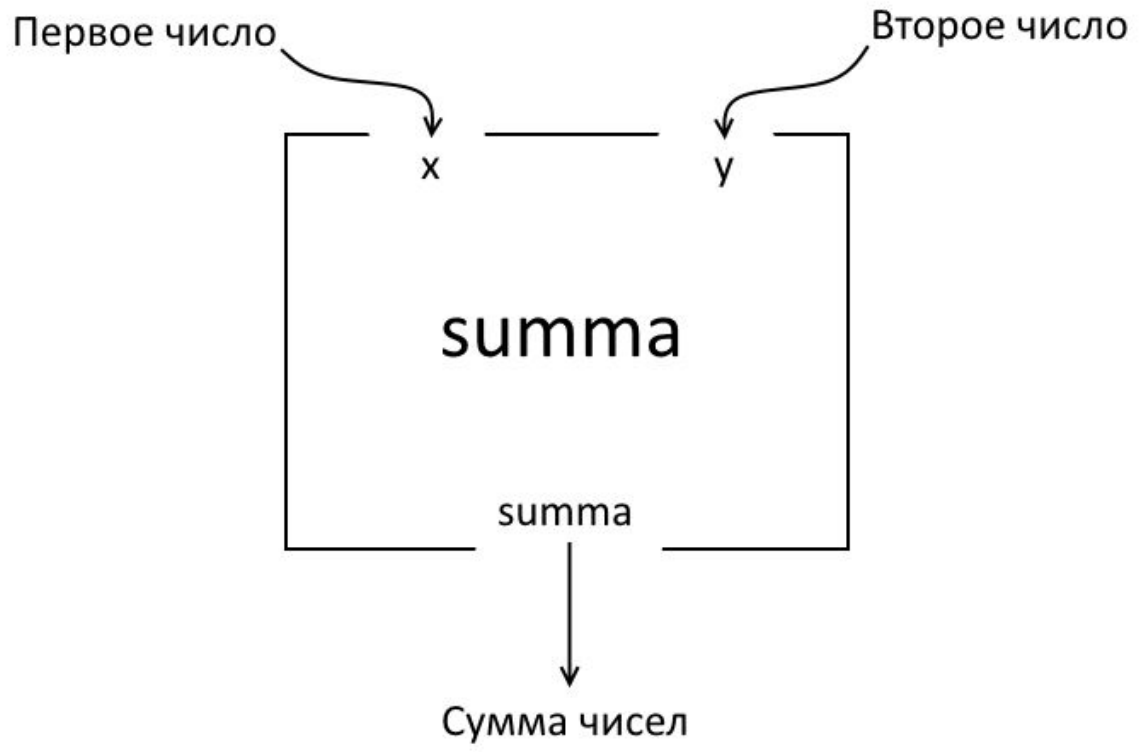
```
def func(chislo):
    chislo = chislo + 1
    return chislo
x = 5
x = func(x)
print(x)
```

Вывелось число 6. Мы можем рассматривать функцию как соковыжималку, в которую сверху подаются переменные-коробочки или мешки-списки. Внутри функции с ними что-то происходит, а потом через условный краник нашей соковыжималки выходят уже другие переменные или списки. Мы будем визуализировать наши функции следующим образом.



Сверху в нашу “соковыжималку” подается какое-то число, переменная chislo принимает это значение. Снизу из краника нашей “соковыжималки” возвращается переменная chislo, которое на 1 больше, чем то число, которые положили в нашу “соковыжималку” сверху. **func** название функции, которые идет сразу после **def**. Те перемены или списки, которые мы хотим положить в нашу

“соковыжималку” указываются в скобках после названия. Они указываются переменными, в которые мы положим значения этих переменных или списков. То, что мы хотим вернуть из нашего краника указывается после ключевого слова return. Оно переводится с английского как “вернуть”. Попробуем написать функцию, которая будет возвращать сумму двух чисел. Для начала визуализируем нашу функцию.



Попробуем написать программу на языке Python.

```
def summa(x, y):
    summa = x + y
    return summa
x = summa(5, 6)
print(x)
```

Как мы можем заметить, переменная x внутри функции и вне ее никак друг от друга не зависят. Можно считать, что функция это совершенно другая программа со своими переменными.

Задачи:

1. Написать функцию вычитания
2. Написать функцию деления

3. Написать функцию деления, которая возвращает строку “На ноль делить нельзя”, если делитель равен нулю.
4. Написать функцию, которая возвращает максимальное из двух чисел
5. Написать функцию, которая возвращает максимальное из трех чисел

Тайминг:

Тема	Время с начала занятия, мин
Зачем нужны функци	15
Что такое функции?	35
Функции в Python	60
Задачи	85
Контрольные вопросы	90

Контрольные вопросы:

1. Что такое функция?
2. Привести несколько примеров функций из реальной жизни
3. Чем создание функции отличается от ее вызова?
4. Что такое передаваемое в функцию значение?
5. Что такое возвращаемое значение?
6. Можно ли использовать функцию практически как число/строку?