



Python. Подготовка к собеседованию

## Урок 3

# Стандартная библиотека Python

Базовые функции. Работа с списками. Кортежи, словари, встроенные функции. Стандартные модули Python.

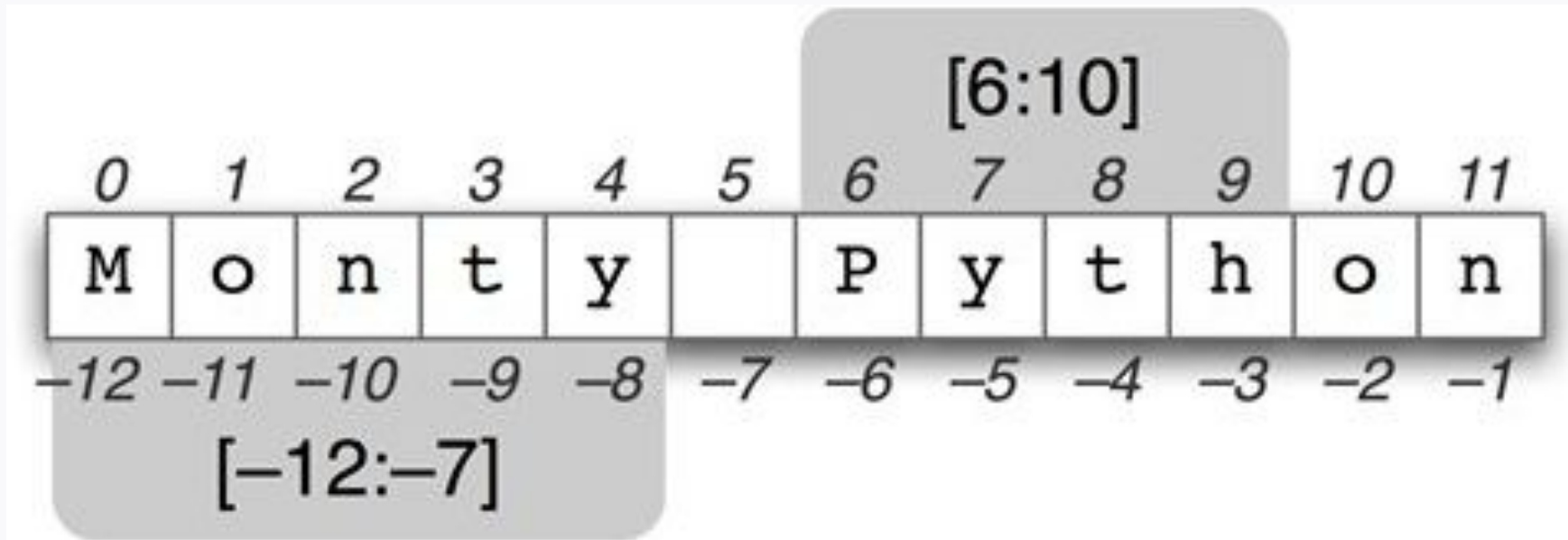
# Цели урока

Повторить словари. Разобрать концепцию ООП.

- Базовые функции;
- Работа с списками;
- Кортежи, словари, встроенные функции;
- Стандартные модули Python.



# Базовые операции со строками



# Методы строк



`'строка'.upper()`

`'строка'.title()`

`'строка'.find('подстрока')`

`'строка'.rfind('подстрока')`

`'строка'.find('подстрока', 'индекс')`



# Форматирование строк

## Strings in Python

‘подстрока\_1’ + ‘переменная’ + ‘подстрока\_2’

‘подстрока\_1, %s’ % (переменная)

‘подстрока\_1, {} {}'.format(переменная\_1, переменная\_2)

‘подстрока\_1, {1} {0}’.format(переменная\_1, переменная\_2)

‘подстрока\_1, {переменная}’.format(переменная=“значение”)

‘подстрока\_1, {0}’.format(“значение”)



# Списки

1.	_____
2.	_____
3.	_____

```
b = list()  a=[1,2,3]
```



# Методы списков

`list.append('элемент_списка')`

`list.pop([i])`

`list.insert(i, 'элемент_списка')`

`list.remove('элемент_списка')`





# Кортежи

переменная = tuple()

переменная = ()

переменная = ('элемент',)

переменная = 'элемент',





# Словари

py\_dict = { 1: 'Apple', 2: 'OnePlus' }

The diagram illustrates the structure of a Python dictionary. It shows the code `py_dict = { 1: 'Apple', 2: 'OnePlus' }`. Above the first item, an arrow labeled 'key' points to the number `1`, and an arrow labeled 'value' points to the string `'Apple'`. Similarly, above the second item, an arrow labeled 'key' points to the number `2`, and an arrow labeled 'value' points to the string `'OnePlus'`. Below the first item, a bracket labeled 'Item 1' spans both the key and value. Below the second item, a bracket labeled 'Item 2' spans both the key and value.



# Методы словарей

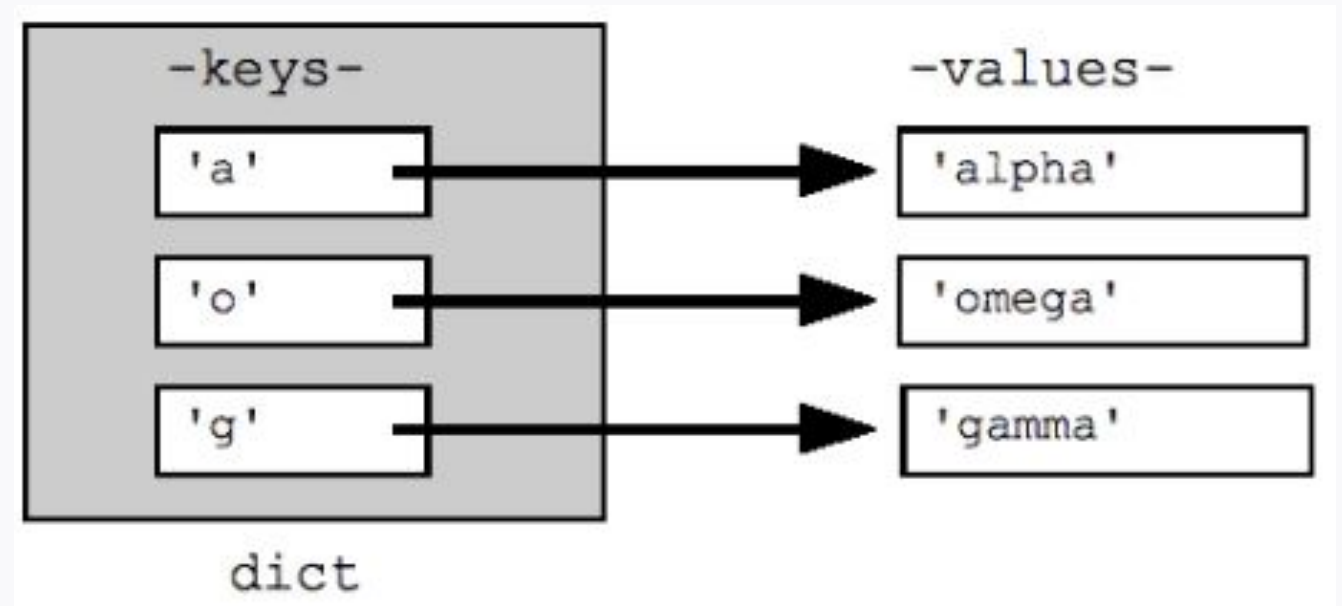
переменная = {}

переменная = {'имя\_ключа': значение, ...}

переменная.keys()

переменная.items()

переменная.values()



# Встроенные функции

print()

input()

len()

int()

range()

sum()

zip()

filter()



# Работа с файлами

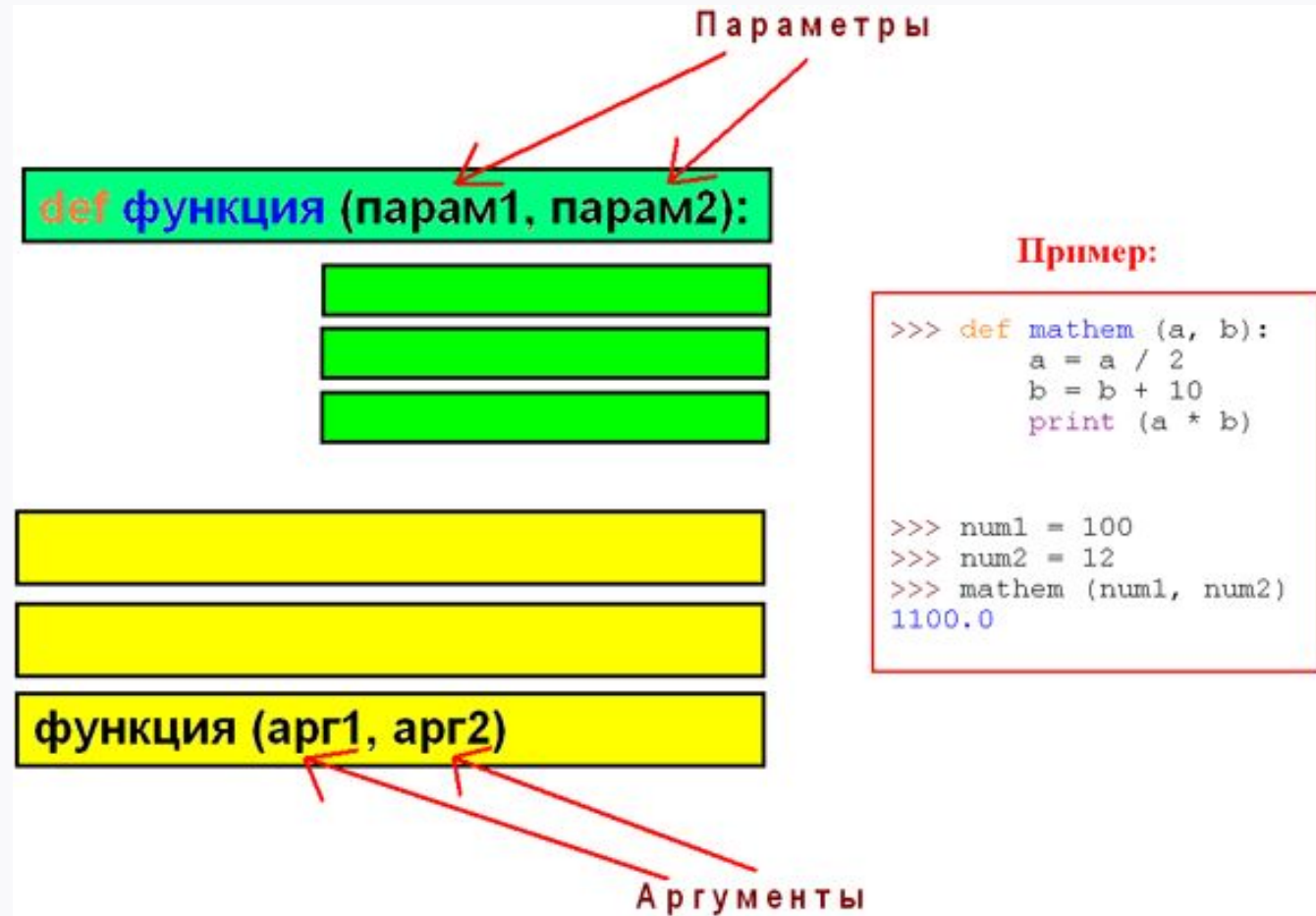
объект\_файла = open(путь\_к\_файлу, 'режим', encoding='кодировка')

with open(путь\_к\_файлу, 'режим', encoding='кодировка') as объект\_файла

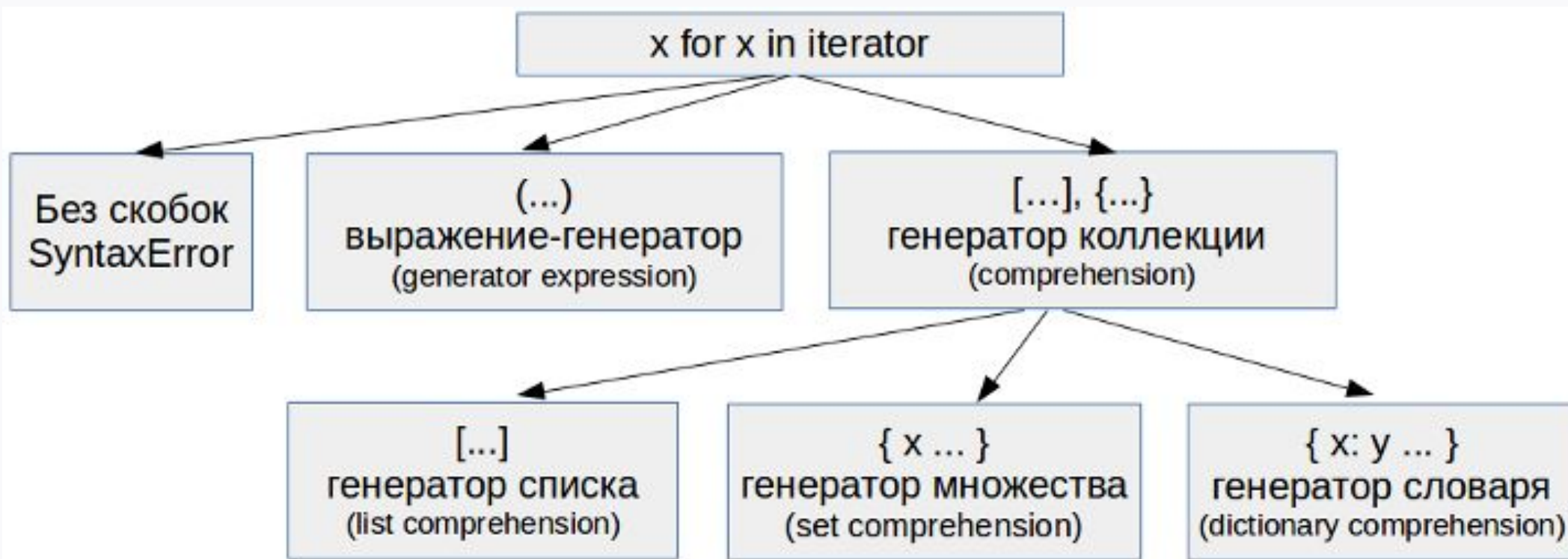




# Передача аргументов по ссылке/значению



# Генераторы списков и словарей



# Регулярные выражения

`re.match(регулярное_выражение, строка)`

`re.search()`

`re.findall()`

`re.split()`

`re.sub()`

`re.compile()`





# Обработка исключений



# Модуль OS



`os.getcwd()`

`os.rename('старое_имя', 'новое_имя')`

`os.replace('старый_файл', 'новый_файл')`      `os.remove('путь')`

`os.path`

`os.mkdir('путь')`



# Модуль SYS

The Python Standard Library

## the Sys module



`sys.argv`

`sys.path`

`sys.modules`

`sys.executable`

`sys.platform`



# Запуск скрипта с параметрами

```
python my_script.py param1 param2 param3
```



# Практическое задание



# Практическое задание (часть 1)

1. Написать программу, которая будет содержать функцию для получения имени файла из полного пути до него. При вызове функции в качестве аргумента должно передаваться имя файла с расширением. В функции необходимо реализовать поиск полного пути по имени файла, а затем «выделение» из этого пути имени файла (без расширения).
2. Написать программу, которая запрашивает у пользователя ввод числа. На введенное число она отвечает сообщением, целое оно или дробное. Если дробное — необходимо далее выполнить сравнение чисел до и после запятой. Если они совпадают, программа должна возвращать значение **True**, иначе **False**.
3. Создать два списка с различным количеством элементов. В первом должны быть записаны ключи, во втором — значения. Необходимо написать функцию, создающую из данных ключей и значений словарь. Если ключу не хватает значения, в словаре для него должно сохраняться значение **None**. Значения, которым не хватило ключей, необходимо отбросить.



# Практическое задание (часть 2)

4. Написать программу, в которой реализовать две функции. В первой должен создаваться простой текстовый файл. Если файл с таким именем уже существует, выводим соответствующее сообщение. Необходимо открыть файл и подготовить два списка: с текстовой и числовой информацией. Для создания списков использовать генераторы. Применить к спискам функцию **zip()**. Результат выполнения этой функции должен быть обработан и записан в файл таким образом, чтобы каждая строка файла содержала текстовое и числовое значение. Вызвать вторую функцию. В нее должна передаваться ссылка на созданный файл. Во второй функции необходимо реализовать открытие файла и простой построчный вывод содержимого. Вся программа должна запускаться по вызову первой функции.





# Практическое задание (часть 3)

5. Усовершенствовать первую функцию из предыдущего примера. Необходимо во втором списке часть строковых значений заменить на значения типа **example345** (строка+число). Далее — усовершенствовать вторую функцию из предыдущего примера (функцию извлечения данных). Дополнительно реализовать поиск определенных подстрок в файле по следующим условиям: вывод первого вхождения, вывод всех вхождений. Реализовать замену всех найденных подстрок на новое значение и вывод всех подстрок, состоящих из букв и цифр и имеющих пробелы только в начале и конце — например, **example345**.



# Дополнительные материалы

1. <https://pythonworld.ru/moduli/modul-os.html>.
2. <https://pythonworld.ru/osnovy/with-as-menedzhery-konteksta.html>.
3. <https://metanit.com/python/tutorial/5.3.php>.
4. <https://o7planning.org/ru/11417/inheritance-and-polymorphism-in-python>.
5. <https://metanit.com/python/tutorial/7.4.php>.

