Библиотека

Библиотека в программировании – сборник подпрограмм или объектов, используемых для разработки программного обеспечения или ПО.

(Википедия https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))

Репозиторий PyPI

PyPI – центральный репозиторий (хранилище) модулей для языка программирования Python. Он как PlayMarket для Android, AppStore для iPhone или CPAN для Perl.

Доступ к PyPI осуществляется по ссылке https://pypi.org/.

Встроенные модули

Python включает в себя комплект стандартной библиотеки, уже достаточно для многих вещей.

Раздел документации по стандартной библиотеке расположен по ссылке https://docs.python.org/3/library/

Описание модулей стандартной библиотеки есть в Википедии https://ru.wikipedia.org/wiki/Стандартная_библиотека_Python

Модули в Python

Модули в Python устроены по иерархическому принципу, как каталоги в файловой системе. Один модуль может быть вложен в другой, причем вложенность не ограничена (хотя на практике редко бывает больше 4). Чтобы пользоваться функциями, объектами и классами из модуля, весь этот модуль или его часть нужно подключить к программе – **импортировать**.

За импорт в Python отвечает директива **import.**

Давайте посмотрим на примерах, как это происходит.

```
From math import pi # Возьмем число пи из библиотеки math
```

Ключевое слово as

Модуль, переменную, класс или функцию можно при импорте назвать своим именем – для этого служит ключевое слово as, например:

```
>>>from math import pi as число_пи
```

```
>>>число_пи
```

3.141592653589793

Более того, поскольку в программе на языке Python в именах допустимы буквенные символы любых алфавитов, можно использовать даже греческие буквы (впрочем, это неудобно, если у вас кириллическо-латинская клавиатура).

```
>>>from math import pi as π

>>>print(π)
```

3.141592653589793

Важно!

Мы можем импортировать всю библиотеку, но тогда для доступа к ее содержимому нужно снова использовать точку:

```
>>>import  math

>>>print(math.pi)
```

3.141592653589793

Особенности импорта модулей

Значения после директивы import можно писать через запятую:

```
>>>from math import sin, cos, tan
```

Значок * означает, что из библиотеки нужно импортировать все, что доступно.

```
>>>from math import*
```

Для получения перечня методов библиотеки существует оператор **dir.**

```
>>>import math

>>>dir(math)
```

['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos',

'acosh','asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign',

'cos', 'cosh','degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',

'factorial', 'floor','fmod', 'frexp', 'fsum', 'gamma', 'hypot', 'isfinite',

'isinf', 'isnan', 'ldexp','lgamma', 'log', 'log10', 'log1p', 'log2',

'modf', 'pi', 'pow', 'radians', 'sin','sinh', 'sqrt', 'tan', 'tanh', 'trunc']

Для получения помощи по использованию метода нужно воспользоваться командой **help.**

```
>>>help (math.sin)
```

Help on built-in function sin in module math:

sin(...)

sin(x)

Return the sine of x (measured in radians).

Модуль random

Этот модуль предназначен для работы с псевдослучайными последовательностями. Такие последовательности важны в математическом моделировании, в криптографии и в различных играх.

Давайте посмотрим структуру модуля.

```
>>> import random
```

Для получения одного псевдослучайного целого числа можно воспользоваться одной из двух функций: randrange или randint. Функция randrange возвращает случайное число из диапазона. Как и в обычном range, мы можем указать начало, конец и шаг диапазона. Функция randint работает похожим образом, но у нее границы диапазона – обязательные параметры, нельзя указать шаг, и верхняя граница включена в диапазон генерации.

```
from random import randrange, randint


# возвращаем случайное целое из диапазона
print(randrange(100))
print(randrange(40, 100, 5))
print(randint(10, 20))
```

Функция choice

Одна из самых популярных – функция choice. С ее помощью можно выбрать один вариант из нескольких альтернатив, заданных в списке, кортеже, строке или любом другом индексируемом типе.

choice нельзя применять для неупорядоченных коллекций – например, множеств и словарей, а также функция не работает с пустыми коллекциями.

Например, вот так можно моделировать подкидывание монетки:

```
>>> from random import choice
>>> choice((1, 2))
2
>>> choice(["орел", "решка"])
'орел'
>>> choice("ab")
'a'
```

А так – имитировать несколько бросков игральных кубиков:

```
from random import choice


dashes = [1, 2, 3, 4, 5, 6]
for i in range(1, 10):
    print(choice(dashes), choice(dashes))
```

```
2 5
6 5
6 1
1 2
5 6
6 1
4 2
4 2
2 3
```

Если нам нужно вернуть не один, а несколько элементов, на помощь придут функции choices и sample. choices возвращает заданное именованным

параметром k количество элементов с возможными повторами (коллекция должна быть непустой), sample – без повторов, но выборка должна быть меньше или равна длине коллекции, иначе тоже будет ошибка.

```
from random import choices, sample


my_list = ['Yes', 'No', 'May be']
# выбираем k элементов коллекции с повторениями
print(choices(my_list, k=5))
# выбираем k элементов без повторений
print(sample(range(10), 6))
```

['No', 'Yes', 'Yes', 'Yes', 'No']

[8, 9, 1, 6, 7, 5]

А функция random возвращает случайное вещественное число от 0 до 1 (не включительно):

```
from random import random as rnd


print(rnd(), rnd(), rnd()) # вещественное число [0, 1)
```

0.7807663953103449 0.1503300563891775 0.6068329639725171

Установка пакетов

Для установки пакетов в Python служит специальная утилита командной строки pip, которая является еще и модулем.

Чтобы установить пакет, нужно выполнить команду

pip install <Имя модуля>.

Если нужно обновить модуль, то выполняем команду

pip install – U <Имя модуля>.

Вам может понадобиться вызвать – h, чтобы получить полный список всего, что pip может сделать.

Команда pip выполняется в командной строке. Чтобы туда попасть нужно сначала открыть меню «Выполнить». Это можно сделать двумя

способами. Наиболее быстрый – это нажать на клавиатуре сочетание клавиш Win + R или запустить его из меню Пуск правой кнопкой мыши.

Затем в командной строке окна «Выполнить» набираем **cmd** и выполняем. Должно появиться окно черного цвета с командной строкой. Команда **pip** выполняется из каталога, содержащего Python.

Найти в проводнике папку, где находится Python. В командной строке перейти к этой папке с помощью команды **cd**.

Далее представлены ссылки на релизы для установки библиотек и информационные ресурсы по работе с ними:

https://www.lfd.uci.edu/~gohlke/pythonlibs/ – список библиотек с указанием ссылок на последние релизы для установки

https://pythonworld.ru/uploads/pythonworldru.pdf – учебник по Python

Numpy

https://pythonworld.ru/numpy/1.html - руководство по работе

https://numpy.org/ - основная информация по **Numpy, в том числе по ссылке –**

https://numpy.org/devdocs/user/quickstart.html – руководство по использованию

Pandas

https://pandas.pydata.org/ – основная информация, в том числе – руководство по использованию по ссылке:

https://pandas.pydata.org/docs/getting_started/index.html

руководство по установке по ссылке:

https://pandas.pydata.org/docs/getting_started/install.html#installing-pandas

Matplotlib

https://matplotlib.org/ – основная информация, в том числе

руководство пользователя по ссылке:

https://matplotlib.org/users/index.html