Подключение библиотек Python

При разработке приложений существует необходимость использования одного и того же функционала повторно. Повторное использование кода необходимо не только самому автору кода. Иногда разработанный функционал настолько полезен, что его использование упростило бы работу широкому спектру разработчиков программного обеспечения.

Python позволяет нам упаковывать код в файлы или наборы файлов, называемые модулями. Модуль — это набор пространств имен Python, предназначенных для широкого использования в качестве инструмента.

Выражаясь более конкретно, модули обычно соответствуют файлам программ Python. Каждый файл является модулем, а модули импортируют другие модули, чтобы задействовать определяемые в них имена.

Обычно, чтобы использовать модуль в файле, вам понадобится следующий синтаксис в верхней части этого файла:

>>from module_name import object_name

Однако, есть несколько возможных вариантов работы с библиотеками:

`import`	Позволяет клиенту (импортеру) извлечь модуль как единое целое.
`from`	Позволяет клиентам извлекать отдельные имена из модуля.

Одной из распространенных библиотек, входящей в состав стандартной библиотеки Python, является datetime. datetime помогает вам работать с датами и временем в Python.

Методы класса datetime:

datetime.today() - объект datetime из текущей даты и времени. Работает также, как и datetime.now() со значением tz=None.

datetime.fromtimestamp(timestamp) - дата из стандартного представления времени.

datetime.fromordinal(ordinal) - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.

datetime.now(tz=None) - объект datetime из текущей даты и времени.

datetime.combine(date, time) - объект datetime из комбинации объектов date и time.

datetime.strptime(date_string, format) - преобразует строку в datetime (так же, как и функция strptime из модуля time).

datetime.strftime(format) - см. функцию strftime из модуля time.

datetime.date() - объект даты (с отсечением времени).

datetime.time() - объект времени (с отсечением даты).

datetime.replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]]]) - возвращает новый объект datetime с изменёнными атрибутами.

datetime.timetuple() - возвращает struct_time из datetime.

datetime.toordinal() - количество дней, прошедших с 01.01.1970.

datetime.timestamp() - возвращает время в секундах с начала эпохи.

datetime.weekday() - день недели в виде числа, понедельник - 0, воскресенье - 6.

datetime.isoweekday() - день недели в виде числа, понедельник - 1, воскресенье - 7.

datetime.isocalendar() - кортеж (год в формате ISO, ISO номер недели, ISO день недели).

datetime.isoformat(sep='T') - красивая строка вида "YYYY-MM-DDTHH:MM:SS.mmmmmm" или, если microsecond == 0, "YYYY-MM-DDTHH:MM:SS"

Иногда необходимо выполнять форматирование вывода даты и времени. Для этого необходимо учитывать таблицу форматов:

Символ	**Описание**	**Пример**
%a	День недели, короткий вариант	Wed
%A	Будний день, полный вариант	Wednesday
%w	День недели числом 0-6, 0 — воскресенье	3
%d	День месяца 01-31	31
%b	Название месяца, короткий вариант	Dec
%B	Название месяца, полное название	December
%m	Месяц числом 01-12	12
%y	Год, короткий вариант, без века	18
%Y	Год, полный вариант	2018
%H	Час 00-23	17
%I	Час 00-12	05
%p	AM/PM	PM
%M	Минута 00-59	41
%S	Секунда 00-59	08
%f	Микросекунда 000000-999999	548513
%z	Разница UTC	+0100
%Z	Часовой пояс	CST
%j	День в году 001-366	365
%U	Неделя числом в году, Воскресенье первый день недели, 00-53	52
%W	Неделя числом в году, Понедельник первый день недели, 00-53	52
%c	Локальная версия даты и времени	Mon Dec 31 17:41:00 2018
%x	Локальная версия даты	12/31/18
%X	Локальная версия времени	17:41:00
%%	Символ “%”	%

Задание

1. Импортировать библиотеку **datetime.**
2. Создайте переменную current_time и установите ее равной datetime.now()
3. Вывести в консоль результат

Модуль Random

Модуль random предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

random.seed([X], version=2) - инициализация генератора случайных чисел. Если X не указан, используется системное время.

random.getstate() - внутреннее состояние генератора.

random.setstate(state) - восстанавливает внутреннее состояние генератора. Параметр state должен быть получен функцией getstate().

random.getrandbits(N) - возвращает N случайных бит.

random.randrange(start, stop, step) - возвращает случайно выбранное число из последовательности.

random.randint(A, B) - случайное целое число N, $A \leq N \leq B$.

random.choice(sequence) - случайный элемент непустой последовательности.

random.shuffle(sequence, [rand]) - перемешивает последовательность (изменяется сама последовательность). Поэтому функция не работает для неизменяемых объектов.

random.sample(population, k) - список длиной k из последовательности population.

random.random() - случайное число от 0 до 1.

random.uniform(A, B) - случайное число с плавающей точкой, $A \leq N \leq B$ (или $B \leq N \leq A$).

random.triangular(low, high, mode) - случайное число с плавающей точкой, $low \leq N \leq high$. Mode - распределение.

random.betavariate(alpha, beta) - бета-распределение. alpha>0, beta>0. Возвращает от 0 до 1.

random.expovariate(lambd) - экспоненциальное распределение. lambd равен 1/среднее желаемое. Lambd должен быть отличным от нуля. Возвращаемые значения от 0 до плюс бесконечности, если lambd положительно, и от минус бесконечности до 0, если lambd отрицательный.

random.gammavariate(alpha, beta) - гамма-распределение. Условия на параметры alpha>0 и beta>0.

random.gauss(значение, стандартное отклонение) - распределение Гаусса.

random.lognormvariate(mu, sigma) - логарифм нормального распределения. Если взять натуральный логарифм этого распределения, то вы получите нормальное распределение со средним mu и стандартным отклонением sigma. mu может иметь любое значение, и sigma должна быть больше нуля.

random.normalvariate(mu, sigma) - нормальное распределение. mu - среднее значение, sigma - стандартное отклонение.

Задание

1. Импортировать библиотеку random
2. Создайте переменную random_list и установите ее равной пустому списку
3. Превратите пустой список в понимание списка, которое использует random.randint() для генерации случайного целого числа от 1 до 100 (включительно) для каждого числа в диапазоне (101).

4. Создайте новую переменную randomer_number и установите ее равной random.choice() с random_list в качестве аргумента.

Импортирование модулей и пространство имен

Обратите внимание: когда мы хотим вызвать функцию randint(), мы вызываем random.randint(). Это поведение по умолчанию, когда Python предлагает пространство имен для модуля. Пространство имен изолирует функции, классы и переменные, определенные в модуле, от кода в файле, выполняющем импорт. Тем временем ваше локальное пространство имен — это место, где запускается ваш код.

Python по умолчанию называет пространство имен после импортируемого модуля, но иногда это имя может быть двусмысленным или длинным. Иногда имя модуля также может конфликтовать с объектом, который вы определили в своем локальном пространстве имен.

К счастью, это имя можно изменить с помощью псевдонима с помощью ключевого слова as:

>>import module_name as name_you_pick_for_the_module

Задание

1. Импортируйте pyplot из модуля matplotlib с псевдонимом plt.
2. импортируйте random ниже других операторов импорта. Лучше всего хранить весь импорт в верхней части файла.
3. Создайте переменную number_a и установите ее равной диапазону чисел от 1 до 12 (включительно).
4. Создайте переменную number_b и установите ее равной случайной выборке из двенадцати чисел в диапазоне (1000).
5. Теперь давайте построим эти наборы чисел друг против друга, используя plt. Вызовите plt.plot() с двумя переменными в качестве аргументов.
6. Теперь вызовите plt.show() и запустите свой код!

Вы должны увидеть график отображаемых случайных чисел. Для этого вы использовали два модуля Python (random и matplotlib).

Задание 1

1. Создайте переменную employees и присвойте ей список, который состоит из списка сотрудников:

ФИО	Должность	Дата найма	Оклад
Иванов Иван Иванович	Менеджер	22.10.2013	250 000
Сорокина Екатерина Матвеевна	Аналитик	12.03.2020	75 000

Струков Иван Сергеевич	Старший программист	23.04.2012	150 000
Корнеева Анна Игоревна	Ведущий программист	22.02.2015	120 000
Старчиков Сергей Анатольевич	Младший программист	12.11.2021	50 000
Бутенко Артем Андреевич	Архитектор	12.02.2010	200 000
Савченко Алина Сергеевна	Старший аналитик	13.04.2016	100 000

2. Начальство решило выплачивать премии ко дню программиста в сентябре (13 сентября – день программиста). Премия будет составлять 3% от оклада каждому программисту. Реализуйте функцию, которая будет рассчитывать премию.
3. Начальство решило выплачивать премии к 8 марта всем сотрудницам, а к 23 февраля сотрудникам, равные 2000. Реализуйте функцию, которая будет рассчитывать премию.
4. Было решено провести индексацию зарплат сотрудников. Сотрудникам, которые отработали в компании более 10 лет, индексация будет равна 7% оклада, остальным – 5%.
5. Необходимо составить график отпусков сотрудников. Для этого необходимо в отдельный список занести тех сотрудников, которые отработали в компании более 6 месяцев. Реализовать функцию.

Задание 2

В торговом центре проводится розыгрыш призов. Для получения приза необходимо загадать число. Случайным образом администраторы выбирают второе число в интервале от 1 до 9. Те пользовательские номера, сумма цифр которого делятся нацело на загаданное администрацией число, считаются выигрышными. Компания обратилась к вам, чтобы вы разработали приложение, которое будет осуществлять вывод в консоль выигрышных номеров. Если выигрышных номеров слишком много, отбираются первые 5, то есть как только в консоль выведено 5 игроков, розыгрыш заканчивается.

Задания для самостоятельной работы

Задание 1. Напишите программу, которая с помощью модуля random моделирует броски монеты. Программа принимает на вход количество попыток и выводит результаты бросков: Орел или Решка (каждое на отдельной строке).

Задание 2. Напишите программу, которая с помощью модуля random моделирует броски игрального кубика с 6 гранями. Программа принимает на вход количество попыток и выводит результаты бросков — выпавшее число, которое написано на грани кубика (каждое на отдельной строке).

Задание 3. Напишите программу, которая с помощью модуля random генерирует случайный пароль. Программа принимает на вход длину пароля и выводит случайный пароль, содержащий только символы английского алфавита a..z, A..Z (в нижнем и верхнем регистре).