



**Работа с файлами и каталогами. Основные операции с путями к файлам. Импорт пакета. Важнейшие стандартные пакеты. Подсистема `pip`. Установка стороннего модуля. Создание собственных модулей.**

Турашова Анна Николаевна

Преподаватель

[anna1turashova@gmail.com](mailto:anna1turashova@gmail.com)

Telegram: @anna1tur



# Проверка домашнего задания



## Стандартный модуль ***datetime***

Содержит 5 классов:

**datetime.datetime** — для представления одновременно даты и времени.

**datetime.date** — для представления только даты. Содержит методы, аналогичные методам datetime для работы с датами.

**datetime.time** — для представления только времени. Содержит методы, аналогичные методам datetime для работы со временем.

**datetime.timedelta** — для представления разницы во времени, используется для проведения арифметических действий над датами и временем

**datetime.tzinfo** — для представления информации о временной зоне (часовой пояс).

Полная документация к модулю доступна на сайте **python** (на английском языке).

<https://docs.python.org/3.8/library/datetime.html>



## Задание 1.

Попробуйте создать объекты класса `datetime.datetime`:

```
import datetime
dt = datetime.datetime(2019, 1, 16)
dt2 = datetime.datetime(2019, 1, 16, minute=11)
dt3 = datetime.datetime.today()
print(dt.replace(year=2013, hour=12))
```

Дополнительно почитайте про этот модуль в интернете.  
Попробуйте также сложить дни:

```
now = datetime.datetime.now()
two_days = datetime.timedelta(2)
in_two_days = now + two_days
```



## Задание 2.

Почитайте различные статьи про ООП.  
Например, эту:

<https://python.ivan-shamaev.ru/classes-python-3-methods-oop-examples/>



# Datetime

# Модуль datetime



```
import datetime as dt
```

```
# тип данных 'дата' (год + месяц + день)
```

```
my_date = dt.date(2019, 11, 5)
```

```
print(my_date)
```

```
print(dt.date.today())
```

```
print(dt.date.today().weekday())
```

```
# тип данных 'время' (час + минута + секунда +  
миллисекунда)
```

```
my_time = dt.time(23, 15, 29)
```

```
print(my_time)
```

# Модуль datetime



# тип данных дата + время

```
my_datetime = dt.datetime(2020, 12, 15, 13, 25,  
9) print(my_datetime)
```

# можно объединить дату и время

```
my_date = dt.date(2019, 11, 5)
```

```
my_time = dt.time(23, 15, 29)
```

```
my_datetime = dt.datetime.combine(my_date,  
my_time)
```

```
print(my_datetime)
```

```
print(my_datetime.year)
```

# и получить текущую дату и время

```
print(dt.datetime.now())
```



# Интервал времени timedelta



# можно указать, сколько недель, дней, часов, минут, секунд содержит интервал

```
delta_time1 = dt.timedelta(seconds=10,  
weeks=2)
```

```
print(delta_time1)
```

```
print(delta_time1.days)
```

```
print(delta_time1.total_seconds())
```

# интервал может быть отрицательным

```
delta_time2 = dt.timedelta(days=-10)
```

```
print(delta_time2)
```

# Операции с датами и временем



# операции для интервалов времени

```
delta_time1 = dt.timedelta(days=4, hours=4)
```

```
delta_time2 = dt.timedelta(days=1, hours=1)
```

```
print(delta_time1 - delta_time2)
```

```
print(delta_time1 * 10)
```

```
print(delta_time1 / delta_time2)
```

# операции для дат

```
date1 = dt.date(2021, 11, 5)
```

```
date2 = dt.date(2020, 05, 1)
```

```
delta_time1 = dt.timedelta(days=4, hours=4)
```

```
print(date1 + delta_time1)
```

```
print(date1 - date2)
```



# Модуль OS, zip, fractional

# Файлы и папки



- Все ОС поддерживают иерархические файловые системы. Файлы размещаются в папках (каталогах).
- Папка — это контейнер, содержащий файлы, причем папка может быть вложена в другую папку, а та — в другую папку и т. д.
- Файловая система представляет собой дерево с вершинами-папками и листьями-файлами.
- В Python есть средства для работы с файловой системой:
  - Операции с файлами и папками: копирование, перемещение, удаление
  - Обход файловой системы
  - Проверка существования файла и его типа

# Модуль os



- `name` – тип ОС (`posix`, `nt`, `java`)
- `system(command)` – выполнение команды ОС
- `getcwd()` – текущая папка
- `chdir(путь)` – сменить текущую папку
- `access(имя файла, флаг)` – проверка доступности файла, флаги `W_OK`, `R_OK`, `F_OK` отвечают за возможность записи, чтения, а также за факт существования файла
- `listdir(путь)` – возвращает список файлов и папок в папке
- `walks(путь)` – рекурсивный перебор папок, начиная с указанной, возвращает кортеж: текущий каталог, список вложенных каталогов, список файлов в текущей папке

```
for currentdir, dirs, files in os.walk('files'):
    print(currentdir, dirs, files)
```



# Операции с файлами



## Модуль os

- `os.mkdir(path)` - создаёт папку. `OSError`, если существует.
- `os.remove(path)` - удаляет файл
- `os.rename(src, dst)` - переименовывает файл или папку из `src` в `dst`
- `os.rmdir(path)` - удаляет пустую папку

## Модуль shutil (shell utilites)

- `shutil.copy(источник, приемник)` – копирование файлов
- `shutil.copytree(источник, приемник)` – копирование папок
- `shutil.move(источник, приемник)` – перенос файлов или папок
- `shutil.rmtree(путь)` – удаление папки со всем содержимым

# Модуль `os.path`



- `exists(имя)` можно проверить, существует ли файл
- `isfile(имя)` и `isdir(имя)` - определение типа: файл или папка
- `abspath(имя)` - вернет абсолютный путь по относительному
- `dirname(имя)` — полное имя папки, в которой находится файл
- `getatime(path)` - время последнего доступа к файлу
- `getmtime(path)` - время последнего изменения файла
- `getctime(path)` - время создания файла
- `getsize(path)` - размер файла в байтах

# Работа с архивами zip



- `shutil.make_archive(base_name, format[, root_dir])` - создаёт архив и возвращает его имя
  - `base_name` это имя файла для создания, включая путь, но не включая расширения (не нужно писать ".zip" и т.д.)
  - `format` - формат архива
  - `root_dir` – папка, которую мы архивируем
- `shutil.unpack_archive(filename[, extract_dir[, format]])` - распаковывает архив
  - `filename` - полный путь к архиву
  - `extract_dir` - куда будет извлекаться
  - `format` - формат архива



# Модуль fractional



Fractional - рациональное число

Можно создать из пары чисел (числитель, знаменатель), из действительного числа и строки

```
from fractions import Fraction
```

```
print(Fraction(3, 9))
```

```
print(Fraction())
```

```
print(Fraction('3/7'))
```

```
print(Fraction(3.1415))
```

```
print(Fraction('1/3') + Fraction('1/7'))
```



# Домашнее задание



# Задание 1.

С помощью модуля `os` создайте папку в директории вашего проекта. Затем добавьте в неё несколько файлов и папок. Попробуйте переименовать файлы, скопировать.

Заархивируйте созданную папку.

В конце удалите всё созданное с помощью модуля `os`.



Входит в ГК Аплана



**АКАДЕМИЯ АЙТИ**

Основана в 1995 г.

Е-learning  
и очное  
обучение

Направления обучения:

Информационные технологии

Информационная безопасность

ИТ-менеджмент и управление проектами

Разработка и тестирование ПО

Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск,  
Хабаровск, Красноярск, Тюмень, Нижний  
Новгород, Краснодар, Волгоград, Ростов-на-Дону



Ежегодные награды  
Microsoft,  
Huawei, Cisco и  
другие

Головной офис  
в Москве

Разработка  
программного  
обеспечения и  
информационных  
систем

Программы по  
импортозамещению

Ресурсы более 400  
высококласных  
экспертов и  
преподавателей

Сеть региональных учебных центров  
по всей России

Крупные заказчики



**100+**

сотрудников



АКАДЕМИЯ АЙТИ



# Спасибо за внимание!

Центральный офис:

Москва, Варшавское шоссе 47, корп. 4, 7 этаж

Тел: +7 (495) 150-96-00

[academy@it.ru](mailto:academy@it.ru)

[academyit.ru](http://academyit.ru)