InnoDom

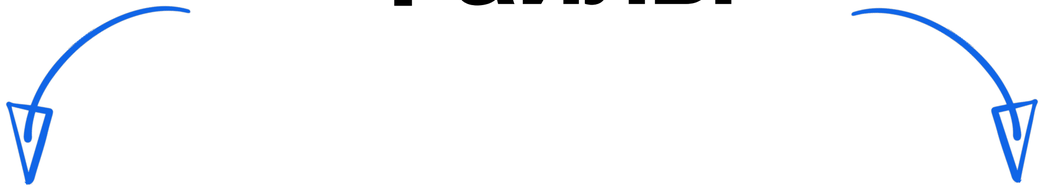
# Файлы

# Файлы

это набор данных, хранящий последовательность из *битов*.

У файла есть всегда есть **название** и **расширение**. Расширение файла - это несколько букв в конце файла через *точку*, оно определяет принадлежность файла к какому-либо приложению.

# Файлы



## Текстовые

тип файла, который хранит символы и данные в том виде, в котором человек их внёс

## Бинарные

тип файла, который хранит биты. В таких файлах данные отображаются в виде 0 и 1

# Операции работы с файлами

- Открытие
- Закрытие
- Запись или чтение

# Текстовые файлы (.txt)

Это простейший формат хранения текста, его преимущество в том, что конвертировать его можно практически в *любой* другой текстовый файл.

Однако читать его неудобно из-за того, что форматирование практически отсутствует.

# Конструкции и функции для работы с файлами

Инструменты Python позволяют использовать 2 способа для работы с файлами:

1. Работа с файлами через функцию `open()` и `close()`

```
f = open('mytxtfile.txt', 'r')


f.close()
```

# Конструкции и функции для работы с файлами

2. Работа с файлами через конструкцию `with`

```
with open('mytxtfile.txt', 'r') as f:
    pass
```

! Ключевое слово `pass` позволяет ничего не делать, в данном случае используется чтобы показать синтаксис конструкции, не работая с файлом. `Pass` может использоваться в циклах, условиях, функциях, везде, где действие с конструкцией происходит в локальном месте. Когда элемент управления интерпретатора видит `pass`, он пропускает код после ключевого слова и идёт дальше.

# Функция open()

Функция `open()` принимает себя 2 обязательных параметра: название файла и режим работы с файлом.

! Если вы пишете название файла и такого файла не существует, то функция создает файл с таким же названием в том же месте, где находится ваш python-скрипт.
Если файл открывается в неправильной кодировке, тогда требуется указать параметр **encoding='utf-8'**.

# Режимы работы с файлами

| Режим | Обозначение |
|---|---|
| 'r' | открытие на чтение (является значением по умолчанию) |
| 'w' | открытие на запись, содержимое файла удаляется, если файла не существует, создается новый |
| 'x' | открытие на запись, если файла не существует, иначе исключение |
| 'a' | открытие на дозапись, информация добавляется в конец файла |
| 'b' | открытие в двоичном режиме |
| 't' | открытие в текстовом режиме (является значением по умолчанию) |
| '+' | открытие на чтение и запись |

Режимы могут быть **объединены**, т.е., например, 'rb' - чтение в двоичном режиме. По умолчанию режим равен 'rt'.

# Методы для работы с файлами

| Название метода | Описание |
|---|---|
| `file.closed` | возвращает True, если файл закрыт, иначе False |
| `file.next()` | возвращает следующую строку файла |
| `file.read(n)` | чтение первых *n* символов файла |
| `file.readline()` | читает одну строчку строки или файла |
| `file.readlines()` | читает и возвращает список всех строк в файле |
| `file.write(str)` | добавляет строку str в файл |
| `file.writelines(sequence)` | добавляет последовательность строк в файл |
| `print(str1,file=file1)` | добавляет строчку с str1 в файл file1 |

# Лайфхаки

По файлу можно пройтись циклом, где i - каждая строчка в файле:

```
with open('mytxtfile.txt', 'r') as f:
    for i in f:
        print(i)
```

Посмотреть, содержится ли строка в файле:

```
with open('mytxtfile.txt', 'r') as f:
    if "a" in f:
        print("a")
```

# Лайфхаки

Можно сразу посмотреть всё содержимое файла:

```
with open('mytxtfile.txt', 'r') as f:
    print(*f)
```

Посмотреть информацию о файле:

```
with open('mytxtfile.txt', 'r') as f:
    print(f)
```