

## Практическое задание.

### Теоретическая часть.

Чтобы создать файл на Python, вы можете использовать встроенную функцию `open()`. Вот простой пример:

```
file = open("example.txt", "w")
file.write("Hello, world!")
file.close()
```

В данном примере, функция `open` работает с файлом `example.txt` и выбран режим открытия/создания этого файла.

Вместо режима `"w"` для открытия и записи может быть выбран и другой:

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

Вот несколько примеров использования различных режимов:

- Чтение файла:

```
file = open("example.txt", "r")
content = file.read()
file.close()
print(content)
```

- Запись в файл:

```
file = open("example.txt", "w")
file.write("Hello, world!")
file.close()
```

- Добавление текста в конец файла:

```
file = open("example.txt", "a")
file.write("This is a new line.")
file.close()
```

- Чтение и запись в файл:

```
file = open("example.txt", "r+")
content = file.read()
file.write("This is a new line.")
file.close()
print(content)
```

- Двоичный режим чтения файла:

```
file = open("example.jpg", "rb")
content = file.read()
file.close()
```

- Создание нового файла, если он не существует:

```
file = open("example.txt", "x")
file.write("Hello, world!")
file.close()
```

- Открытие файла в текстовом режиме на чтение и запись:

```
file = open("example.txt", "r+")
content = file.read()
file.write("This is a new line.")
file.close()
print(content)
```

Убедитесь, что вы закрываете файл с помощью метода `close()` после работы с ним, чтобы освободить ресурсы.

В этом примере мы открываем файл "example.txt" для записи с помощью режима "w". Затем мы записываем строку "Hello, world!" в файл и закрываем его. Теперь у вас будет файл "example.txt" с текстом "Hello, world!".

Если вы хотите создать файл в определенной директории, укажите путь к файлу вместе с его именем. Например, если вы хотите создать файл "example.txt" в папке "documents", то код будет выглядеть так:

```
file = open("documents/example.txt", "w")
file.write("Hello, world!")
file.close()
```

Не забудьте закрыть файл с помощью метода `close()` после записи данных.

Python предоставляет конструкцию `with` для работы с файлами. Эта конструкция автоматически закрывает файл после окончания работы с ним, что позволяет избежать ошибок и утечки ресурсов. Вот пример:

```
with open("example.txt", "w") as file:  
    file.write("Hello, world!")
```

В этом примере мы открываем файл "example.txt" для записи с помощью режима "w". Затем мы записываем строку "Hello, world!" в файл. После окончания блока with, файл автоматически закрывается, даже если происходит ошибка внутри блока.

Вы также можете использовать конструкцию with для чтения файла:

```
with open("example.txt", "r") as file:  
    content = file.read()  
    print(content)
```

В этом примере мы открываем файл "example.txt" для чтения с помощью режима "r". Затем мы считываем содержимое файла в переменную content и выводим его на экран. После окончания блока with, файл автоматически закрывается.

Конструкция with обеспечивает более безопасное и эффективное использование файлов в Python.

## **Практическая часть.**

### **Задание 1.**

В своей папке создайте текстовый файл с названием «new file.txt» и запишите в него произвольный текст.

### **Задание 2.**

Создайте два текстовых файла: input.txt и output.txt.

В файле input.txt напишите несколько строк текста.

Напишите программу на Python, которая будет считывать содержимое файла input.txt и записывать его в файл output.txt. В конце скопированного сообщения в файле output, добавьте слово <copy>.

Например, если в файле input.txt есть строка "random message", то в файле output.txt будет запись random message <copy>

Для выполнения задания по копированию содержимого файла в другой файл, вы можете использовать следующие шаги:

1. Создайте два текстовых файла: input.txt и output.txt.

2. В файле input.txt напишите несколько строк текста.
3. Напишите программу на Python, которая открывает файл input.txt для чтения и файл output.txt для записи.
4. Считайте содержимое файла input.txt с помощью метода read().
5. Запишите считанное содержимое в файл output.txt с помощью метода write().
6. Закройте оба файла с помощью метода close().

Вот пример программы:

```
with open("input.txt", "r") as file1:
    content = file1.read()

with open("output.txt", "w") as file2:
    file2.write(content + " <copy>")
```

После выполнения этой программы, содержимое файла input.txt будет скопировано в файл output.txt с добавлением слова "copy" в конце.

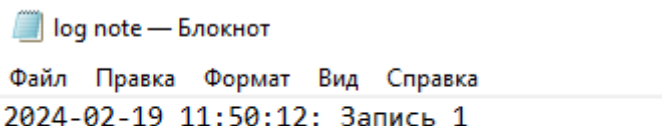
### Задание 3.

Создать функцию, принимающую в себя аргумент строки. Функция создаст файл с именем "log note.txt", а аргумент будет передавать в него новый текст и дата/время его создания.

Пример входных данных 1:

```
add_entry("Запись 1")
```

Пример выходных данных 1:



log note — Блокнот

Файл Правка Формат Вид Справка

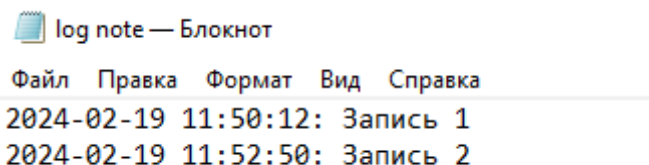
2024-02-19 11:50:12: Запись 1

По мере вызова функции, будут добавляться новые записи:

Пример входных данных 2:

```
add_entry("Запись 2")
```

Пример выходных данных 2:



log note — Блокнот

Файл Правка Формат Вид Справка

2024-02-19 11:50:12: Запись 1

2024-02-19 11:52:50: Запись 2

Возможный алгоритм выполнения:

Для выполнения задания, понадобится модуль datetime, официальная документация: <https://docs.python.org/3/library/datetime.html>

1. Создайте функцию, которая будет принимать в качестве аргумента текстовую строку.
2. Внутри функции, создайте переменную, которая будет хранить текущую дату и время, отформатированные в строку.
3. Откройте файл для записи с использованием конструкции with open().
4. Запишите в файл строку, состоящую из временной метки и текста.
5. Проверьте работу функции, вызвав ее с аргументом "Запись..."

#### Задание 4.

В файле data представлено n количество людей и их придуманные номера телефонов, в формате: n:ФИО:номер

Поместите все записи файла в переменную, и выведите количество человек с именем John.

Возможный алгоритм выполнения:

1. Откройте файл для чтения с использованием конструкции with open().
2. Прочитайте содержимое файла и сохраните его в переменную s.
3. Создайте счетчик count и установите его равным 0.
4. Используйте split('\n') для разделения содержимого файла на строки, затем пройдите цикл по этим строкам.
5. В каждой строке проверьте, содержит ли она 'John'. Если да, увеличьте счетчик на 1.
6. После окончания цикла, выведите значение счетчика, которое будет представлять количество строк, содержащих 'John'.

#### Задание 5.

Раскодируйте сообщение из файла encrypt\_mess.txt, используя следующий код: зашифрованный символ:символ

'б': 'а', 'в': 'б', 'г': 'в', 'д': 'г', 'е': 'д', 'ё': 'е', 'ж': 'ё', 'з': 'ж', 'и': 'з', 'й': 'и', 'к': 'й', 'л': 'к', 'м': 'л',  
'н': 'м', 'о': 'н', 'п': 'о', 'р': 'п', 'с': 'р', 'т': 'с', 'у': 'т', 'ф': 'у', 'х': 'ф', 'ц': 'х', 'ч': 'ц', 'ш': 'ч',  
'щ': 'ш', 'ъ': 'щ', 'ы': 'ъ', 'ь': 'ы', 'э': 'ь', 'ю': 'э', 'я': 'ю', 'а': 'я', 'О': '1', 'Д': '2', 'Т': '3', 'Ч': '4',  
'П': '5', 'Ш': '6', 'С': '7', 'В': '8', 'Е': '9', 'Я': '0', '/': '!', '\_': '!', ':': '!', ' ': '!', '^': '!', '\*': '\n'

1. Создайте словарь, который будет использоваться для шифрования сообщения. Каждый ключ в этом словаре - это символ, который нужно

заменить, а соответствующее значение - символ, на который нужно заменить.

2. Откройте файл с сообщением для чтения. Сохраните содержимое файла в переменную.
3. Пройдитесь по каждому символу в сообщении в цикле. Для каждого символа, проверьте, существует ли он в словаре.
4. Если символ существует в словаре, замените его на соответствующее значение из словаря и напечатайте его. Используйте `end=""` в функции `print()` для того, чтобы избежать перехода на новую строку после каждого символа.
5. После завершения цикла, у вас должно получиться зашифрованное сообщение.

### **Задание 6.**

Создайте программу, которая принимает текстовое сообщение от пользователя. Затем программа спрашивает у пользователя, хочет ли он сохранить это сообщение. Если пользователь отвечает 'yes', программа просит пользователя ввести имя файла и сохраняет введенное сообщение в файл. Если пользователь отвечает что-то другое, программа выводит 'no save' и не сохраняет сообщение.

### **Критерии выставления оценки.**

Оценка 5 - сделано 6 задний.

Оценка 4 – сделано 5 заданий.

Оценка 3 – сделано 4 задания.