

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №2.15 по дисциплине:
основы программной инженерии**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ:

1. Открытие файла

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    fileptr = open("file.txt", "r")
    if fileptr:
        print("file is opened successfully")
    fileptr.close()
```

1 x

C:\Users\79616\PycharmProjects\test.py\venv\Scripts\python3.exe
file is opened successfully

Process finished with exit code 0

2. Применение оператора with

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    with open("file.txt", 'r') as f:
        content = f.read()
    print(content)
```

1 x

C:\Users\79616\PycharmProjects\test.py\venv\Scripts\python3.exe
lilililalala

3. Выполнение примера 1

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     with open("file2.txt", "w") as fileptr:
6         fileptr.write(
7             "Python is the modern day language. It makes things so simple.\n"
8             "It is the fastest-growing programming language"
9         )
```

file2.txt – Блокнот

Файл Правка Формат Вид Справка

Python is the modern day language. It makes things so simple.
It is the fastest-growing programming language

4. Выполнение примера 2

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     with open("file2.txt", "a") as fileptr:
6         fileptr.write(
7             "Python has an easy syntax and user-friendly interaction.")
```

file2.txt – Блокнот

Файл Правка Формат Вид Справка

Python is the modern day language. It makes things so simple.
It is the fastest-growing programming languagePython has an easy syntax and user-friendly

5. Выполнение 3 примера

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     with open("file2.txt", "r") as fileptr:
6         content1 = fileptr.readline()
7         content2 = fileptr.readline()
8
9         print(content1)
10        print(content2)
```

if __name__ == '__main__' > with open("file2.txt", "r") as ...

ex3 x

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex3.py
Python is the modern day language. It makes things so simple.

It is the fastest-growing programming language

6. Выполнение 4 примера

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     with open("file2.txt", "r") as fileptr:
6         content = fileptr.readlines()
7         print(content)
```

ex4 x

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex4.py
['Python is the modern day language. It makes things so simple.\n', 'It is the

7. Выполнение примера 5

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     with open("newfile.txt", "x") as fileptr:
6         print(fileptr)
7
8         if fileptr:
9             print("File created successfully")
10
if __name__ == '__main__' > with open("newfile.txt", "x") a... > if fileptr
```

ex5 ×

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex5.py
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully

8. Выполнение 6 примера

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     with open("text.txt", "w", encoding="utf-8") as fileptr:
6         print(
7             "UTF-8 is a variable-width character encoding used for electronic communication.",
8             file=fileptr
9         )
10        print(
11            "UTF-8 is capable of encoding all 1,112,064 valid character code points.",
12            file=fileptr
13        )
14        print(
15            "In Unicode using one to four one-byte (8-bit) code units.",
16            file=fileptr
17    )
```

9. Выполнение 7 примера

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     with open("text.txt", "r", encoding="utf-8") as f:
6         sentences = f.readlines()
7         for sentence in sentences:
8             if "," in sentence:
9                 print(sentence)
```

if __name__ == "__main__" > with open("text.txt", "r", enco... > for sentence in sentences > if "," in senter

ex7 ×

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex7.py
UTF-8 is capable of encoding all 1,112,064 valid character code points.

10. Выполнение 8 примера

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     with open("file2.txt", "r", encoding="utf-8") as fileptr:
6         print("The filepointer is at byte :", fileptr.tell())
7
8         fileptr.seek(10);
9
10        print("After reading, the filepointer is at:", fileptr.tell())
11
```

ex8 ×

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex8.py
The filepointer is at byte : 0
After reading, the filepointer is at: 10

11. Переименование файла (пример 9)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os

if __name__ == "__main__":
    os.rename("file2.txt", "file3.txt")
```

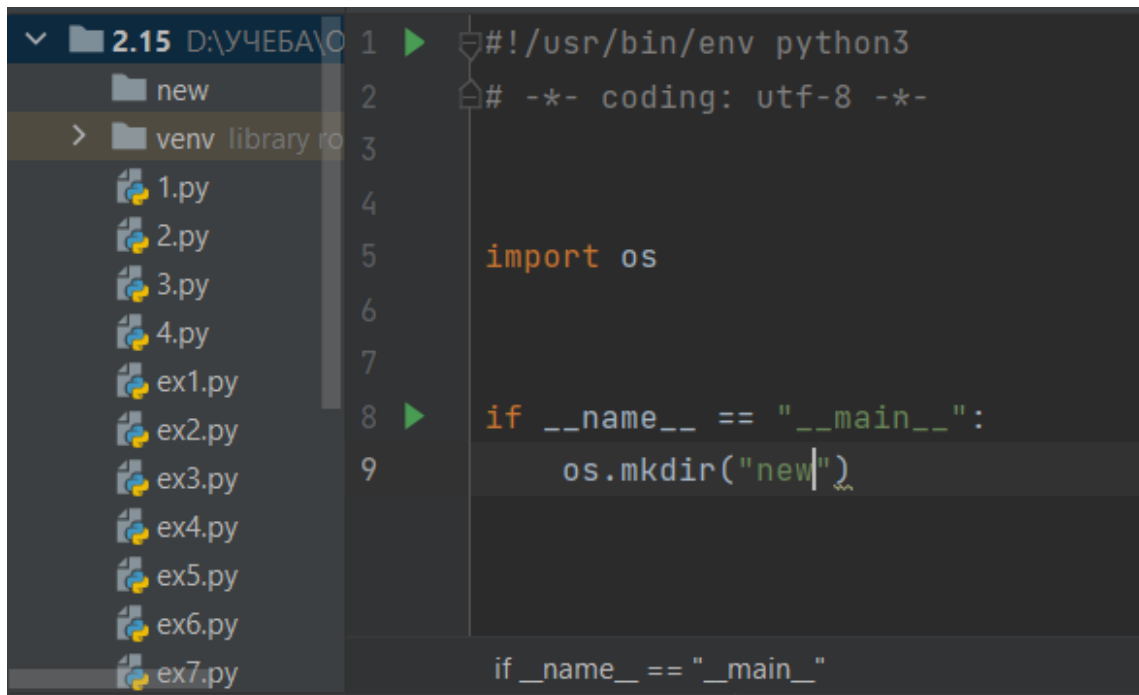
12. Удаление файла (пример 10)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os

if __name__ == "__main__":
    os.remove("newfile.txt")
```

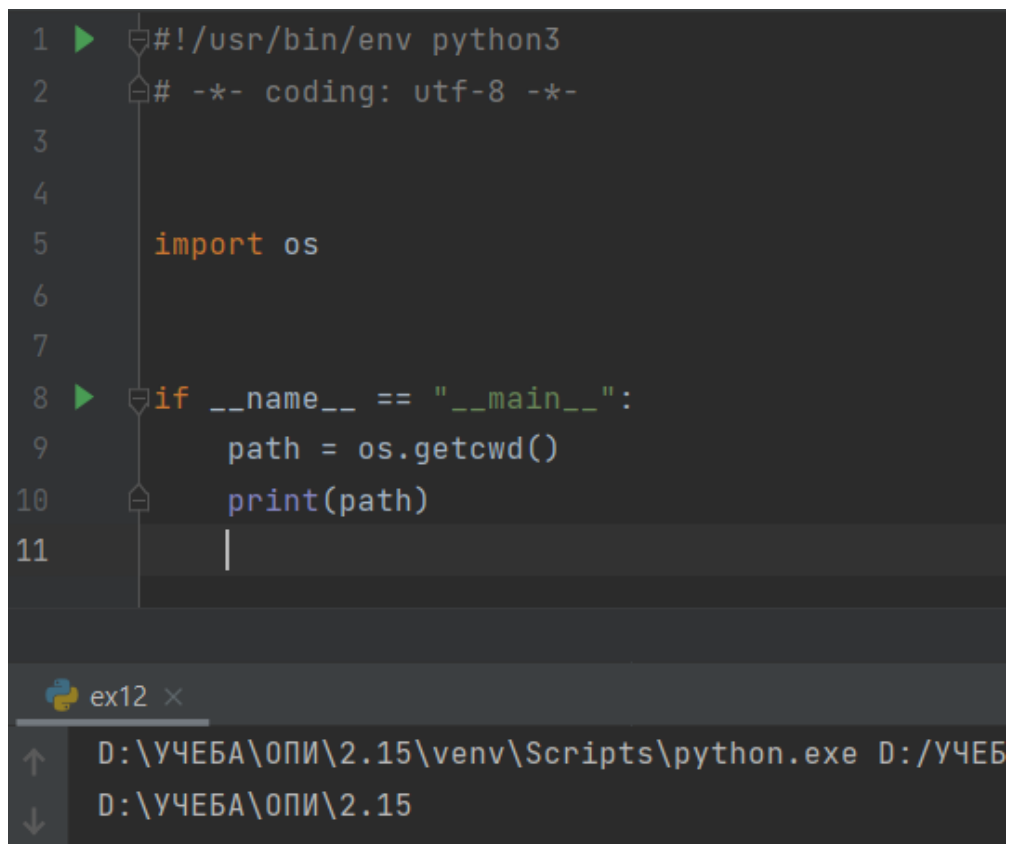
13. Создание нового каталога (пример 11)



The screenshot shows a code editor with a file explorer on the left and a Python script on the right. The file explorer shows a directory structure with a 'new' folder and a 'venv' folder. The Python script is a simple program that creates a new directory named 'new'.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import os
6
7
8 if __name__ == "__main__":
9     os.mkdir("new")
10
11 if __name__ == "__main__":
```

14. Получение текущего рабочего каталога (пример 12)



The screenshot shows a code editor with a Python script that prints the current working directory. Below the editor, a terminal window shows the command and output.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import os
6
7
8 if __name__ == "__main__":
9     path = os.getcwd()
10     print(path)
11
```

Terminal output:

```
ex12 x
D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБ
D:\УЧЕБА\ОПИ\2.15
```


15. Изменение текущего рабочего каталога (пример 13)

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5     import os
6
7
8 ▶ if __name__ == "__main__":
9     os.chdir("C:\\Windows")
10    print(os.getcwd())
11    |
```

ex13 x

↑ D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe
↓ C:\Windows

16. Удаление каталога (пример 14)

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5     import os
6
7
8 ▶ if __name__ == "__main__":
9     os.rmdir("new")
```

17. Доступ к элементам командной строки в языке программирования Python (пример 15)

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import sys
6
7
8 ▶ if __name__ == "__main__":
9     print("Number of arguments:", len(sys.argv), "arguments")
10    print("Argument List:", str(sys.argv))
11
```

ex15 ×

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex15.py
Number of arguments: 1 arguments
Argument List: ['D:/УЧЕБА/ОПИ/2.15/ex15.py']

18. Выполнение примера 16

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import sys
6
7
8 ▶ if __name__ == "__main__":
9     for idx, arg in enumerate(sys.argv):
10        print(f"Argument #{idx} is {arg}")
11    print("No. of arguments passed is ", len(sys.argv))
```

if __name__ == "__main__"

ex16 ×

D:\УЧЕБА\ОПИ\2.15\venv\Scripts\python.exe D:/УЧЕБА/ОПИ/2.15/ex16.py
Argument #0 is D:/УЧЕБА/ОПИ/2.15/ex16.py
No. of arguments passed is 1

19. Пример использования аргументов командной строки (пример 17)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os
import secrets
import string
import sys

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("The password length is not given!", file=sys.stderr)
        sys.exit(1)

    chars = string.ascii_letters + string.punctuation + string.digits
    length_pwd = int(sys.argv[1])

    result = []
    for _ in range(length_pwd):
        idx = secrets.SystemRandom().randrange(len(chars))
        result.append(chars[idx])

    print(f"Secret Password: {''.join(result)}")
```

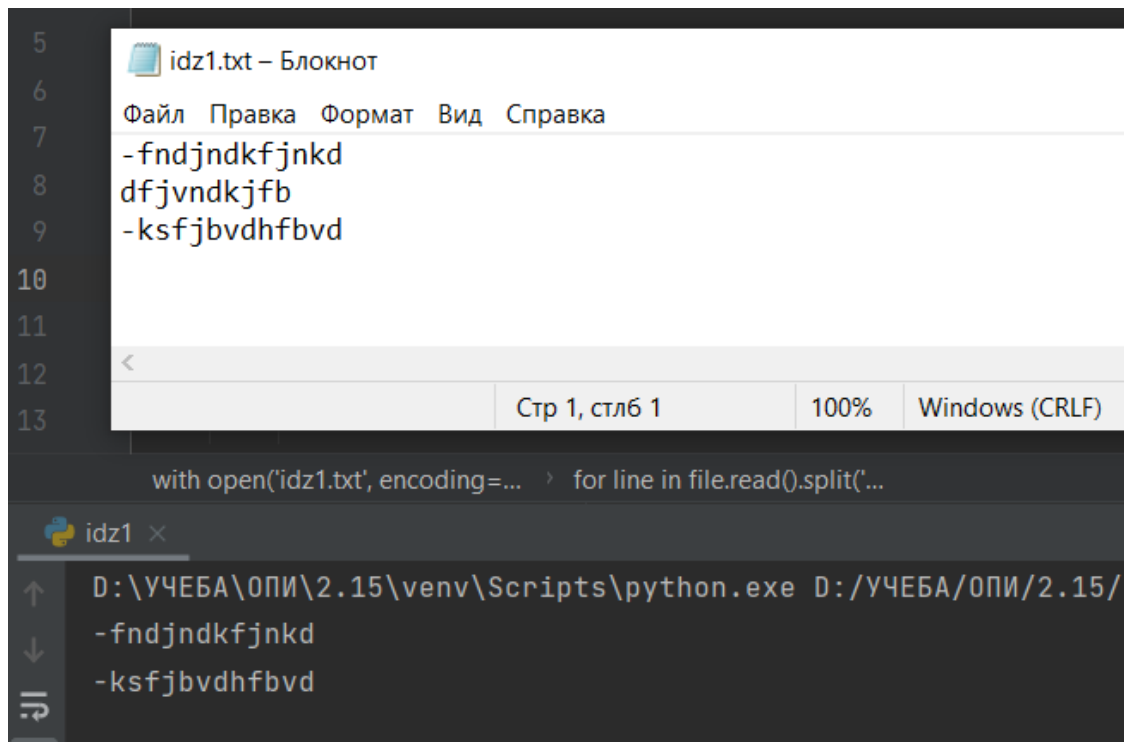
Выполнение индивидуальных заданий.

1. Написать программу, которая считывает текст из файла и выводит на экран только предложения, начинающиеся с тире, перед которым могут находиться только пробельные символы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import os

with open('idz1.txt', encoding='utf-8') as file:
    for line in file.read().split('\n'):
        if line.strip().startswith('-'):
            print(line)
```



2. Истории литературы известен случай написания романа объемом около 50 тыс. слов, в котором ни разу не была употреблена самая популярная в английском алфавите буква Е. Название его – «Gadsby». Напишите программу, которая будет считывать список слов из файла и собирать статистику о том, в каком проценте слов используется каждая буква алфавита. Выведите результат для всех 26 букв английского алфавита и отдельно отметьте букву, которая встречалась в словах наиболее редко. В вашей программе должны игнорироваться знаки препинания и регистр символов.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import re
import string

input_text = open('idz2.txt', 'r').read().lower().split(' ')
alphabet = list(string.ascii_lowercase)
unique_words = []

for word in input_text:
    if word not in unique_words:
        unique_words.append(word)
unique_words = re.sub(r'[\^\\w\\s]', '', str(unique_words)).split()

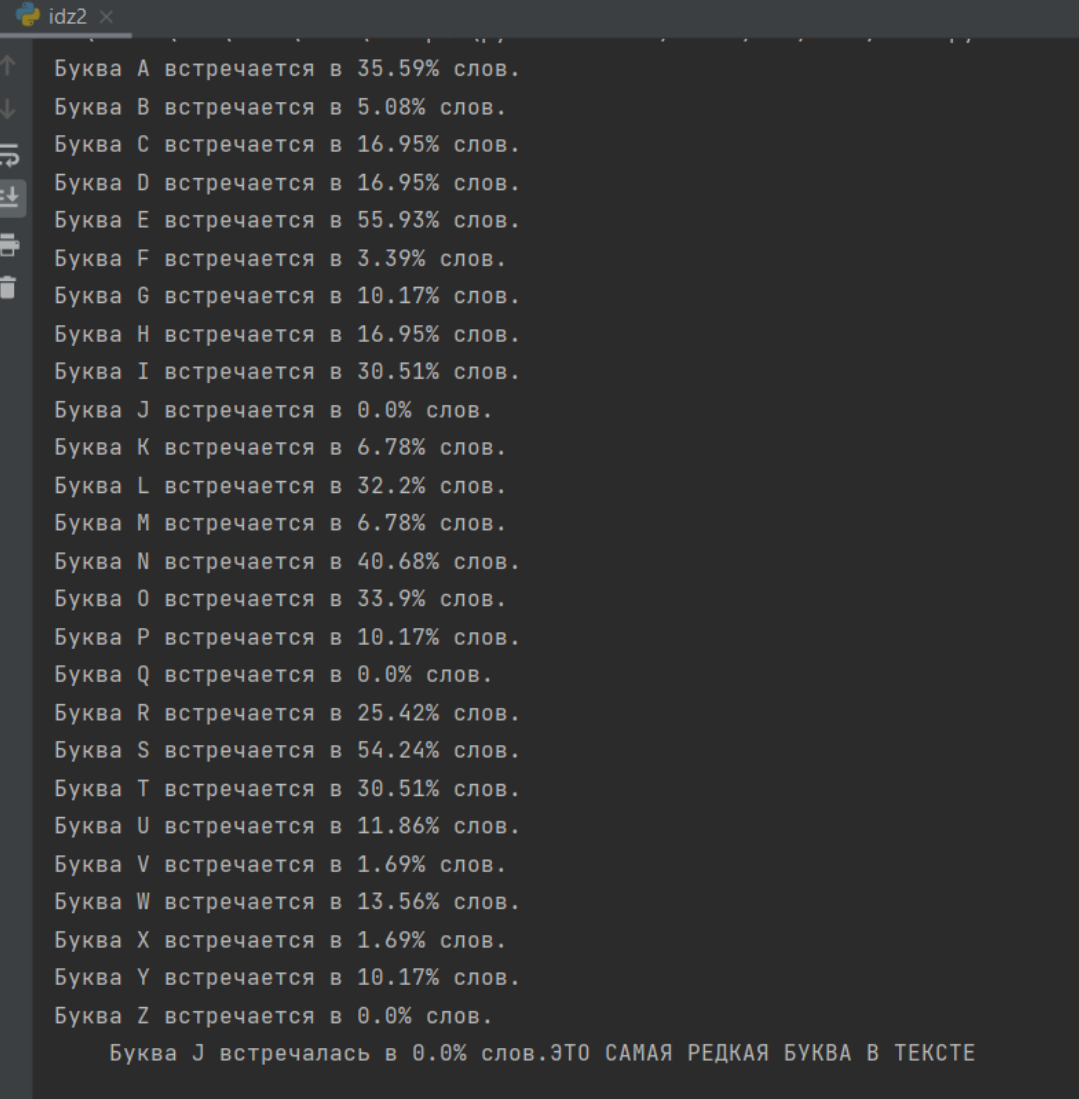
min_value = len(unique_words)
min_char = 0
for char in alphabet:
    counter = 0
    for word in unique_words:
        if char in word:
            counter += 1
    if counter < min_value:
```

```

min_value = counter
min_char = char
print(f'Буква {char.upper()} встречается в {round(counter /
len(unique_words) * 100, 2)}% слов.')

print(f'\tБуква {min_char.upper()} встречалась в {round(min_value /
len(unique_words) * 100, 2)}% слов.'
      f'ЭТО САМАЯ РЕДКАЯ БУКВА В ТЕКСТЕ')

```



```

Буква А встречается в 35.59% слов.
Буква В встречается в 5.08% слов.
Буква С встречается в 16.95% слов.
Буква D встречается в 16.95% слов.
Буква Е встречается в 55.93% слов.
Буква F встречается в 3.39% слов.
Буква G встречается в 10.17% слов.
Буква H встречается в 16.95% слов.
Буква I встречается в 30.51% слов.
Буква J встречается в 0.0% слов.
Буква K встречается в 6.78% слов.
Буква L встречается в 32.2% слов.
Буква M встречается в 6.78% слов.
Буква N встречается в 40.68% слов.
Буква O встречается в 33.9% слов.
Буква P встречается в 10.17% слов.
Буква Q встречается в 0.0% слов.
Буква R встречается в 25.42% слов.
Буква S встречается в 54.24% слов.
Буква T встречается в 30.51% слов.
Буква U встречается в 11.86% слов.
Буква V встречается в 1.69% слов.
Буква W встречается в 13.56% слов.
Буква X встречается в 1.69% слов.
Буква Y встречается в 10.17% слов.
Буква Z встречается в 0.0% слов.

Буква J встречалась в 0.0% слов.ЭТО САМАЯ РЕДКАЯ БУКВА В ТЕКСТЕ

```

Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

С помощью команды: `fileobj = open("file.txt", "r")`

2. Как открыть файл в языке Python только для записи?

С помощью команды: `fileobj = open("file.txt", "w")`

3. Как прочитать данные из файла в языке Python?

`with open("file.txt", 'r') as f:`

`content = f.read(); print(content)`

Построчное чтение содержимого файла в цикле:

```
with open("file2.txt", "r") as fileptr:  
    for i in fileptr:  
        print(i), где i – одна строка файла.
```

Построчное чтение содержимого файла с помощью методов файлового объекта:

```
with open("file2.txt", "r") as fileptr:  
    content1 = fileptr.readline() content2 = fileptr.readline() print(content1)  
    print(content2)
```

Чтение строк с помощью функции `readlines()`:

```
with open("file2.txt", "r") as fileptr:  
    content = fileptr.readlines() print(content)
```

`readlines()` считывает строки в файле до его конца (EOF)

4. Как записать данные в файл в языке Python?

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода `open` с одним из следующих режимов доступа:

'w': он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.

'a': добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует.

5. Как закрыть файл в языке Python?

Используя метод `close()`. Любая незаписанная информация уничтожается после вызова метода `close()` для файлового объекта.

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

Данная конструкция является менеджером контекста. Помимо файлов может использоваться в работе с базами данных.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Есть возможность записать в файл большой объем данных, если он может быть представлен в виде списка строк.

```
with open("examp.le", "w") as f: f.writelines(list_of_strings)
```

8. Какие существуют, помимо рассмотренных, функции модуля os для работы с файловой системой?

Предположим, вы хотите создать не только одну папку, но и несколько вложенных:

```
# вернуться в предыдущую директорию os.chdir("..")
```

```
# сделать несколько вложенных папок os.makedirs("nested1/nested2/nested3")
```

Перемещение файлов:

Функцию `os.replace()` можно использовать для перемещения файлов или каталогов:

```
# заменить (переместить) этот файл в другой каталог os.replace("renamed-text.txt", "folder/renamed-text.txt")
```

Стоит обратить внимание, что это перезапишет путь, поэтому если в папке `folder` уже есть файл с таким же именем (`renamed-text.txt`), он будет перезаписан.

Список файлов и директорий:

```
# распечатать все файлы и папки в текущем каталоге print("Все папки и файлы:", os.listdir())
```

Функция `os.listdir()` возвращает список, который содержит имена файлов в папке. Если в качестве аргумента не указывать ничего, вернется список файлов и папок текущего рабочего каталога:

Все папки и файлы: `['folder', 'handling-files', 'nested1', 'text.txt']`

А что если нужно узнать состав и этих папок тоже? Для этого нужно использовать функцию `os.walk()`:

```
# распечатать все файлы и папки рекурсивно for dirpath, dirnames, filenames in
```

```
os.walk("."): 
```

```
# перебрать каталоги for dirname in dirnames: 
```

`os.walk()` — это генератор дерева каталогов. Он будет перебирать все переданные составляющие. Здесь в качестве аргумента передано значение «.», которое обозначает верхушку дерева.

