

Zadanie 1

Stwórz skrypt, który zliczy ilość linii we wskazanym pliku. Informacje na temat czytanego pliku oraz ilość linii zapisze do drugiego pliku output.txt. Obsłuż wyjątek w przypadku braku pliku we wskazanej ścieżce.

```
In [ ]: def main():
        plik_we = r"ścieżka do pliku"
        plik_wy = r"ścieżka do pliku"
        try:
            with open(plik_we) as we:
                wiersze = we.readlines()
                liczba_wierszy = len(wiersze)
                print(f"Liczba linii w pliku to {liczba_wierszy}")
            with open(plik_wy, "w") as wy:
                wy.writelines(['plik: ',plik_we, ' linie: ', str(liczba_wiersz
y)])
        except FileNotFoundError:
            print("Nie znaleziono pliku")

main()
```

Zadanie 2

Stwórz skrypt, który kasuje plik ze wskazanego folderu. Przygotuj dwie wersje używając:

- try except
- if else

```
In [ ]: import os
        myfile= r"ścieżka do pliku"

        ## If file exists, delete it ##
        if os.path.isfile(myfile):
            os.remove(myfile)
        else:
            ## Show an error ##
            print("Error: %s file not found" % myfile)
```

```
In [ ]: import os
        myfile= r"ścieżka do pliku"
        try:
            os.remove(myfile)
        except FileNotFoundError as e:
            print(e.strerror)
```

Zadanie 3

Stwórz skrypt, który kasuje folder ze wskazanej ścieżki. Przygotuj wersje skryptu używając:

- try except

```
In [1]: import shutil

mydir= r"sciezka do pliku"
try:
    shutil.rmtree(mydir)
except OSError as e:
    print ("Error: %s - %s." % (e.filename, e.strerror))
```

Error: sciezka do pliku - System nie może odnaleźć określonej ścieżki.

Zadanie 4

Kasowanie **starych** plików. Stwórz skrypt, który skasuje pliki ze wskazanej ścieżki starsze o jeden dzień od dziś. Obsłuż wyjątki, jeśli się ich spodziewasz.

```
In [2]: import os
import datetime

def main():
    try:
        katalog = r"sciezka do pliku"
        pliki = os.listdir(katalog)
        print(pliki)
        for plik in pliki:
            sciezka_bezwzgledna = os.path.join(katalog, plik)
            (mode, ino, dev, nlink, uid, gid, size, atime, mtime, ctime) =
os.stat(sciezka_bezwzgledna)
            data_modyfikacji = datetime.datetime.fromtimestamp(mtime)
            roznica = datetime.datetime.now() - data_modyfikacji
            print(f"{plik}, size = {size}, rozmiar B: {size}, mtime = {dat
a_modyfikacji}, roznica={roznica}, czy plik starszy niż 1 dzień: {roznica.
days > 0}") #roznica.seconds
            if roznica.days > 0:
                os.remove(sciezka_bezwzgledna)
        except FileNotFoundError as e:
            print(e.strerror)

main()
```

System nie może odnaleźć określonej ścieżki

Zadanie 5

"Zabawa z plikiem" Stwórz skrypt, który odczyta zawartość pliku, jeśli pliku nie ma, to wyrzuci wyjątek.

```
In [ ]: def main():
    path = r"sciezka do pliku"
    try:
        f = open(path)
        print(f.readlines())
    except OSError as e:
        print(e)

main()
```

Zadanie 6

Stwórz skrypt, który obsłuży błędnie podaną liczbę typu float. Liczba może zostać podana przez metodę `input()` lub jako zmienna.

```
In [ ]: napis_wczytany = "2.,5" #2.5
try:
    liczba = float(napis_wczytany)
    print(f"Liczba jest {liczba}")
except ValueError as e:
    print("Och nie, nie udało się sparsować liczby! Szczegóły poniżej:")
    print(e)
```

Zadanie 7

"Silnia z minusem" Stwórz skrypt, który wyrzuci wyjątek `ValueError` przy błędnie podanej liczbie ujemnej do metody `silnia()`

```
In [ ]: # rozw. 1

def silnia(n):
    if n < 0:
        raise ValueError("silnia niezdefiniowana dla liczb ujemnych")
    wynik = 1
    for i in range(1, n+1):
        wynik *= i
    return wynik

# wywołanie metody z liczbą ujemną
silnia(-1)
```

```
In [ ]: # obsługa błędu

try:
    print(f"Silnia z -5 to {silnia(-5)}")
except ValueError as e:
    print("Och nie, coś poszło nie tak! Szczegóły poniżej:")
    print(e)
```

Zadanie 8

"Zgadnij liczbę" Stwórz skrypt, który wyrzuci wyjątek, gdy liczba będzie za mała lub za duża. Zdefiniuj odpowiednie wyjątki: `ValueTooSmallError` oraz `ValueTooLargeError` dziedziczące po nadzrędnym zdefiniowanym wyjątku `Error`, który dziedziczy po `Exception`. Stwórz odpowiednią `def __init__` w klasie wyjątku z polem `message` opisującym problem.

```

In [ ]: # główny wyjątek
class Error(Exception):
    """Base class for other exceptions"""
    pass

class ValueTooSmallError(Error):
    """Raised when the input value is too small"""
    def __init__(self, message="This value is too small, try again!"):
        self.message = message
        super().__init__(self.message)

class ValueTooLargeError(Error):
    """Raised when the input value is too large"""

    def __init__(self, message="This value is too large, try again!"):
        self.message = message
        super().__init__(self.message)

# you need to guess this number
number = 10

# user guesses a number until he/she gets it right
while True:
    try:
        i_num = int(input("Enter a number: "))
        if i_num < number:
            raise ValueTooSmallError
        elif i_num > number:
            raise ValueTooLargeError
        break
    except ValueTooSmallError as e:
        print(e)
    except ValueTooLargeError as e:
        print(e)

print("Congratulations! You guessed it correctly.")

```

Zadanie 9

"Długie zdanie". Stwórz skrypt, który wyrzuci wyjątek, gdy użytkownik na wejściu poda zdanie mniejsze niż 20 znaków. Zdefiniuj odpowiednio wyjątki samodzielnie. Stwórz odpowiednią definicję `__init__` w klasie wyjątku z polem `message` opisującym problem.

```

In [ ]: class InputTooShort(Exception):
    def __init__(self, message="To zdanie jest za krótkie!"):
        self.message = message
        super().__init__(self.message)

user_input = input("Wpisz jakieś fajne zdanie: \n")

if len(user_input) < 20:
    raise InputTooShort
else:
    print(user_input)

```

Zadanie 10

"Ile mam lat?" Stwórz skrypt, który wyrzuci wyjątek, gdy użytkownik przekroczy 25 lat. Na wejściu pobieraj rok urodzenia. Stwórz odpowiednią `def __init__` w klasie w wyjątku z polem `message` opisującym problem. W zadaniu użyj modułu: `datetime`

```
In [ ]: import datetime

class YouAreTooOld(Exception):

    def __init__(self, message="jesteś już ciut za stary :)"):
        self.message = message
        super().__init__(self.message)

current_year = datetime.datetime.now().year
user_year = int(input("Podaj rok urodzenia: "))
if current_year - user_year > 25:
    raise YouAreTooOld
else:
    print("Wiek poniżej 25 lat.")
```

Zadanie 11

"Ile ma stron?" Wykorzystaj moduł `PyPDF2` do odczytu plików PDF. Stwórz skrypt, który odczyta plik PDF i wyświetli ilość stron. Aby zainstalować moduł wykonaj: `! pip install PyPDF2`

```
In [ ]: # ! pip install PyPDF2
import PyPDF2

pdfFile = open(r'sciezka do pliku pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFile)
print(pdfReader.getNumPages())
pageObj = pdfReader.getPage(1)
print(pageObj.extractText())
pdfFile.close()
```

Zadanie 12

"Podziel PDF" Wykorzystaj bibliotekę `PyPDF2` do odczytu plików PDF. Stwórz skrypt, który umożliwi podział pliku PDF. Metodę nazwij: `pdf_split(pdf, startPage, endPage)`

```
In [ ]: def pdf_split(pdf, startPage, endPage):
    pdfFile = open(pdf, 'rb')
    pdfReader = PyPDF2.PdfFileReader(pdfFile)
    pdfWriter = PyPDF2.PdfFileWriter()
    outputpdf = pdf.split('.pdf')[0] + '_part' + '.pdf'
    for i in range(startPage, endPage):
        pdfWriter.addPage(pdfReader.getPage(i))
        with open(outputpdf, "wb") as f: # with do pracy ze strumieniami p
lików, zastępuje close
            pdfWriter.write(f)

    pdfFile.close()

pdf_split(r'sciezka do pliku pdf', 100, 101)
```

Zadanie 13

"Przenieś plik" Stwórz skrypt, który przeniesie plik z jednego folderu do drugiego. Zaimportuj dwie biblioteki `os` oraz `shutil`. Wykonaj tą samą czynność w wykorzystując dwa sposoby:

- przeniesienie poprzez zmianę ścieżki do pliku
- klasyczne przenoszenie.

```
In [ ]: import os
import shutil

# Move a file by renaming it's path
os.rename(r'sciezka/1/dom.pdf', r'sciezka/2/dom.pdf')

# Move a file from the directory 1 to 2
shutil.move(r'sciezka', r'nowa sciezka')
```

Zadanie 14

"Grupowa zmiana nazw plików" Stwórz skrypt, który umożliwi grupową zmianę nazw plików w folderze. Zaimportuj moduł `os`.

```
In [ ]: import os

def main(prefix, sufix, path_to_files):
    try:
        for i, f in enumerate(os.listdir(path_to_files)): #enumerate, aby
dodac indeks
            before = path_to_files + '/' + f
            after = path_to_files + '/' + prefix + str(i + 1) + sufix
            os.rename(before, after)
            print("renamed: " + f + " to: " + prefix + str(i + 1) + sufix)
    except FileNotFoundError as e:
        print(e.strerror)

main("textowy", ".txt", r'X:/example')
```

Objaśnienie metody `open()`

w : Otwiera plik do zapisu i tworzy nowy plik, jeśli jeszcze nie istnieje. W przypadku, gdy plik istnieje, zastępuje go.

w+ : Otwiera plik do zapisu i tworzy nowy plik, jeśli jeszcze nie istnieje. W przypadku, gdy plik istnieje, zastępuje go.

r : Otwiera plik tylko do odczytu.

rb : Otwiera plik do odczytu w formacie binarnym.

wb : Otwiera plik do zapisu w formacie binarnym.

wb+ : Oba powyższe

a : Otwiera plik i dodaje nowe treści na koniec pliku (appending)

+ : Ogólnie rzecz biorąc, ten znak jest używany obok r, w lub a i oznacza zarówno pisanie, jak i czytanie.

Jeśli nie określono trybu pliku, to domyślnie zostanie przyjęty **r**.

Objaśnienie rodzajów wyjątków:

`ArithmeticError` :

Klasa bazowa wyjątków wbudowanych wywoływanych w przypadku wystąpienia różnych błędów arytmetycznych:

`OverflowError` , `ZeroDivisionError` , `FloatingPointError`

`LookupError` Klasa bazowa wyjątków wywoływanych w przypadku użycia niewłaściwego klucza lub indeksu dla uzyskania dostępu do elementów sekwencji: `IndexError` , `KeyError`

`EOFError` Wywołany w przypadku, gdy jedna z funkcji wbudowanych `input()` lub `raw_input()` osiągnie koniec pliku (`EOF`)

`FloatingPointError` Wywołany w przypadku błędu operacji zmiennoprzecinkowej.

`IOError` Wywołany w przypadku wystąpienia błędu operacji wejścia/wyjścia instrukcji `print()` , funkcji wbudowanej `open()` czy też metod obiektów plikowych

`ImportError` Wywołany w przypadku wystąpienia błędu instrukcji `import`

`IndexError` Wywołany w przypadku użycia indeksu spoza zasięgu

`KeyError` Wywołany w przypadku nie wystąpienia klucza

`MemoryError` Wywołany w przypadku wyczerpania się pamięci operacji

`NameError` Wywołany w przypadku nieodnalezienia lokalnej lub globalnej nazwy

`OverflowError` Wywołany w przypadku wystąpienia wyniku operacji arytmetycznej, który jest zbyt duży aby mógł być reprezentowany.

`SyntaxError` Wywołany w sytuacji, gdy parser napotka błąd składniowy.

`TypeError` Wywołany w sytuacji wykonania operacji wbudowanej na obiekcie niewłaściwego typu.

`ValueError` Wywołany w przypadku wywołania operacji z argumentem właściwego typu, lecz o nieprawidłowej wartości

`ZeroDivisionError` Wywołany w przypadku, gdy drugi argument operacji dzielenia lub reszty z dzielenia (`modulo`) jest równy zero