

Do wykonania zadań należy pobrać treść z pliku dołączonego do zadań.

poniżej kod startowy do zadań oraz potrzebne importy

```
In [ ]: import re

path_to_file = r"{twoja sciezka do plku}\sample_text.txt"
sample_text = open(path_to_file, 'r', encoding="utf-8").read()
print(f"{sample_text}")
```

Zadanie 1

*Wyszukaj w tekście słowa **który** oraz ich odmian. Policz ilość wystąpień.*

```
In [ ]: # rozwiązanie:

pattern = re.compile(r'który\w*')
matches = pattern.finditer(sample_text)
for i, match in enumerate(matches):
    print(f"{i + 1}: {match}")
```

Zadanie 2

*Wyszukaj w tekście wszystkie litery **Aa**. Policz ilość wystąpień.*

```
In [ ]: pattern = re.compile(r'[Aa]')
matches = pattern.finditer(sample_text)
a_counter = 0
for i, match in enumerate(matches):
    a_counter += 1

print(f"Litera 'A' oraz 'a' występuje w tekście: {a_counter} razy")
```

Zadanie 3

Test jest sformatowany jak w książce, a chcemy mieć każde zdanie od nowej linii, trzeba:

- wyczyścić tekst ze wszystkich znaków nowej linii i tabulatorów
- następie każde zdanie przenieść od nowej linii
- usunąć znaki spacji na początku i na końcu zdania, jeśli wystąpią
- całą logikę umieścić w `def replace_all(text)` możesz umieścić dodatkowe argumenty w metodzie, jeśli chcesz.

Wynik transformacji tekstu wydrukuj na ekran.

```
In [ ]: def replace_all(text):
        dic = {"\n":r"\n", "\t":r"\t", "\r": r"\r"}
        for i, j in dic.items():
            text = text.replace(i, j)
        pattern = re.compile(r"[^.!?]*[.!?]")
        matches = pattern.finditer(text)
        output = ""
        for match in matches:
            output += match.group().strip() + "\n"
        return output

output = replace_all(sample_text)
print(output)
```

Zadanie 4

Zamień w tekście co drugą literę na dużą. PoWiNiEn PoWsTaC TaKi TeKsT

Wykorzystaj przekonwertowany tekst z zadania 3

Wydrukuj na ekran wynik zadania.

```
In [ ]: output = replace_all(sample_text)

i = 0
for letter in output:
    if i % 2 == 0:
        output = output[:i] + letter.upper() + output[i + 1:]
    i += 1
print(output)
```

Zadanie 5

Stwórz maski dla liczb w tekście z maila otrzymanego od kolegi z pracy.

Poniższe zdanie zawiera pin do karty. Przekształć zdanie tak, aby w miejscu pinu widniał napis: (pin) .

e-mail: Cześć Patryk, to jest pin do karty naszego klienta: 4589. Jeśli możesz, to zmień mu pin na taki: 0043 w bazie danych. Dzięki!

```
In [ ]: text = "Cześć Patryk, to jest pin do karty naszego klienta: 4589. Jeśli mo
zesz, to zmień mu pin na taki: 0043 w bazie danych. Dzięki!"

def hide_pin(text):
    pattern = re.compile(r"\d{4}")
    matches = pattern.finditer(text)
    for match in matches:
        text = text[:match.span()[0]] + "(pin)" + text[match.span()[1] +
1:]
    return text

output = hide_pin(text)
print(output)
```

Zadanie 6

Twoim zadaniem jest stworzyć skrypt, który sprawdzi poprawność zapisu dat.

Musisz wykonać regexy dla takich przypadków:

```
dates = [  
"2020-10-12",  
"2020.10.12",  
"2020/10/12",  
"12-10-2020",  
"12.10.2020",  
"12/10/2020",  
"12_10_2020",  
"2020_10_12"  
]
```

Metodę nazwij `check_date(strdate)` powinna zwracać `true/false`. Spróbuj stworzyć kilka prostrzych regexów i umieścić je w liście. Takie podejście jest czytelniejsze i pozwala na łatwą rozbudowę metody w przyszłości. Metoda powinna zwrócić `False` w dwóch ostatnich przypadkach

```
In [ ]: dates = [  
"2020-10-12",  
"2020.10.12",  
"2020/10/12",  
"12-10-2020",  
"12.10.2020",  
"12/10/2020",  
"12_10_2020",  
"2020_10_12"  
]  
  
def check_date(strdate):  
    dic = {  
        r"\d{4}[-]\d{2}[-]\d{2}",  
        r"\d{4}[\.]\d{2}[\.]\d{2}",  
        r"\d{4}[\/]\d{2}[\/]\d{2}",  
        r"\d{2}[-]\d{2}[-]\d{4}",  
        r"\d{2}[\.]\d{2}[\.]\d{4}",  
        r"\d{2}[\/]\d{2}[\/]\d{4}",  
    }  
  
    for i, j in enumerate(dic):  
        pattern = re.compile(j)  
        if pattern.findall(strdate):  
            return True  
    return False  
    # print(f"{i + 1}: pattern: {j}: {match}")  
  
for i in range(0, len(dates)):  
    print(f"{i + 1}. date: {dates[i]}: {check_date(dates[i])}")
```