

1) Krótki feedback na temat ostatnich zajęć

W którą stronę zmienić prędkość tłumaczenia?

- A. Przyspieszyć zajęcia
- B. Zostawić tak, jak było
- C. Zwolnić zajęcia
- D. Nie uczestniczyłem w ostatnich zajęciach (2 tygodnie temu, 27 października)

**nie zwracajcie uwagi na "poprawną odpowiedź" w tym pytaniu.*

2) Czy udało ci się rozwiązać zadanie #2 na ostatnich zajęciach?

Zadanie 2

(5 - 7 min)

1. Poprosić użytkownika o podanie kilku nazw przedmiotów.
2. Stworzyć listę złożoną z takiej samej ilości losowych liczb.
3. Używając znanych poleceń, wypisać oceny obok każdego przedmiotu (jak na przykładzie poniżej).
4. Stworzyć plik `oceny.txt` w którym wypisać oceny do każdego przedmiotu.

Oczekiwany widok pliku `oceny.txt` :

```
Analiza Matematyczna - 5
Język Angielski - 4
Business Intelligence - 3
Matematyka Wyższa dla Inżynierów - 4
Sztuczna Inteligencja - 5
```

- A. Tak, skończyłem go (robiłem samodzielnie lub z pomocą kolegów))
- B. Tak, skończyłem go (przepisałem z tablicy)
- C. Zdążyłem zrobić tylko część
- D. Nawet nie zacząłem go rozwiązywać
- E. Nie było mnie na ostatnim zajęciu

**nie zwracajcie uwagi na "poprawną odpowiedź" w tym pytaniu.*

3) W jaki sposób tworzymy pustą listę w języku Python?

- A. `pusta_lista = Vector{Float64}()`
- B. `pusta_lista = []`
- C. `var pusta_lista = List[Double]()`
- D. `pusta_lista <- list()`

4) Pobierzemy od użytkownika 3 przedmioty. W tym celu najpierw tworzymy pustą listę, a dalej prosimy użytkownika o podanie każdego przedmiotu:

```
przedmioty = []

# Pobieramy pierwszą nazwę przedmiotu:
nazwa_pierwszego_przedmiotu = ?_1_?("Podaj pierwszy przedmiot: ")
przedmioty.?_2_?(nazwa_pierwszego_przedmiotu)

# Możemy trochę skrócić kod, nie tworząc osobnej zmiennej dla nazwy
# przedmiotu, a natychmiast przekazując ten przedmiot do listy.
# A więc dodajemy drugi przedmiot:
przedmioty.?_2_?(?_1_?("Podaj drugi przedmiot: "))

# I ostatni trzeci przedmiot:
przedmioty.?_2_?(?_1_?("Podaj trzeci przedmiot: "))
```

Wpisz nazwę funkcji, którą warto użyć zamiast ?_1_?: _____ .

Jakiej metody użyjemy w miejscu ?_2_?, żeby dodać nową nazwę przedmiotu do naszej listy?

- A. `extend`
- B. `add`
- C. `append`

5) W ostatnim zadaniu utworzyliśmy następujący kod (jeżeli pominąć komentarze):

```
przedmioty = []
nazwa_pierwszego_przedmiotu = ?_1_?("Podaj pierwszy przedmiot: ")
przedmioty.?_2_?(nazwa_pierwszego_przedmiotu)
przedmioty.?_2_?(?_1_?("Podaj drugi przedmiot: "))
przedmioty.?_2_?(?_1_?("Podaj trzeci przedmiot: "))
```

Jak widzimy, bardzo duża ilość kodu się powtarza w każdej linijce. Nie jest to duży problem dla trzech przedmiotów, ale gdybyśmy chcieli dodać 10, to byłoby to już trochę irytujące. Ten problem można rozwiązać za pomocą pętli:

```
przedmioty = []
for i in ?__?:
    przedmioty.append(input("Podaj przedmiot: "))
```

Które z poniższych poleceń najlepiej użyć zamiast znaku ?__?:

- A. range(3)
- B. enumerate(["Przedm 1", "Przedm 2", "Przedm 3"])
- C. zip(["Przedm "] * 3, ["1", "2", "3"])

Możemy jeszcze dodać numerację przedmiotów "pierwszy", "drugi", "trzeci"... W tym celu wystarczy stworzyć listę ["pierwszy", "drugi", "trzeci"] i korzystać z niej, zamiast funkcji range(3):

```
przedmioty = []
liczebniki = ["pierwszy", "drugi", "trzeci"]
for ?_1_? in liczebniki:
    przedmioty.append(input(?_2_?"Podaj {liczebnik} przedmiot: "))
```

Co trzeba wpisać zamiast ?_1_?, żeby skrypt wykonał się poprawnie? _____
Jaki typ ciągu znaków trzeba użyć, żeby powyższy skrypt wykonał się sensownie?

- A. r-string: r"Podaj {liczebnik} przedmiot: "
- B. f-string: f"Podaj {liczebnik} przedmiot: "
- C. u-string: u"Podaj {liczebnik} przedmiot: "
- D. b-string: b"Podaj {liczebnik} przedmiot: "

6) Dodajmy cztery linijki do poprzedniego skryptu. W nich stworzymy losowe oceny dla każdego przedmiotu. Przy każdej nowej linijce w komentarzu wskazane jest przeznaczenie tej linijki.

```
?_1_? random          # Importowanie paczki `random`

oceny = []             # Utworzenie pustej listy dla ocen
przedmioty = []
liczebniki = ["pierwszy", "drugi", "trzeci"]
for liczebnik in liczebniki:
    przedmioty.append(input(f"Podaj {liczebnik} przedmiot") )

    ocena = ?_2_?      # Tworzenie przypadkowej liczby z zakresu (2, 5)
    ?_3_?              # Dodanie tej liczby `ocena` do listy `oceny`
```

?_1_? - _____

?_2_? - _____

?_3_? - _____

7) Dygresja (nie ma obowiązku przepisywać ten kod)

Jaki wynik zwróci poniższy kod?

```
jakis_tekst = "ABC"
jakis_tekst = jakis_tekst + "XYZ"
print(jakis_tekst)
```

- A. ABC
- B. XYZ
- C. ABCXYZ
- D. jakis_tekst

A taki kod?

```
inny_tekst = "ABC"
inny_tekst += "ZYX"
print(inny_tekst)
```

- A. ABC
- B. ZYX
- C. ABCXYZ
- D. ABCZYX

8) Dygresja (niema obowiązku przepisywać ten kod)

Jaką funkcję należy wpisać w pole `?___?`, aby wynik skryptu odpowiadał wynikowi podanemu poniżej?

```
>>> for a, b in ?___?(
...     ["WMiFS", "WEiI", "WBMiL"],
...     ["Trendy Pink", "Hippie Blue", "Old Brick"]
... ):
...     print(f"Kolor logo wydziału {a} nazywa się '{b}'")
...
Kolor logo wydziału WMiFS nazywa się 'Trendy Pink'.
Kolor logo wydziału WEiI nazywa się 'Hippie Blue'.
Kolor logo wydziału WBMiL nazywa się 'Old Brick'.
```

- A. range
- B. enumerate
- C. zip
- D. list

Nazwy kolorów pobrano ze strony <https://colors.artyclick.com/color-name-finder/>.

9) Wracając do naszego zadania, przeanalizujmy poniższy kod:

```
>>> wyniki = ""
>>> wyniki += f"{przedmioty[0]} - {oceny[0]}\n"
>>> wyniki += f"{przedmioty[1]} - {oceny[1]}\n"
>>> wyniki += f"{przedmioty[2]} - {oceny[2]}\n"
>>> print(wyniki)
Analiza Matematyczna - 3
Język Angielski - 4
Wychowanie Fizyczne - 5
```

Gdybyśmy chcieli przepisać powyższy kod w postaci pętli, to tym razem potrzebujemy dla każdej iteracji dwie wartości: ocenę i przedmiot.

```
wyniki = ""
for przedmiot, ocena in ?__?:
    wyniki += f"{przedmiot} - {ocena}\n"
print(wyniki)
```

Wpisz polecenie, którego chcesz użyć zamiast ?__?:

10) Aktualnie nasz kod wygląda następująco:

```
import random

przedmioty = []
oceny = []
liczebniki = ["pierwszy", "drugi", "trzeci"]
for liczebnik in liczebniki:
    przedmioty.append(input(f"Podaj {liczebnik} przedmiot:"))
    oceny.append(random.randint(2, 5))

wyniki = ""
for przedmiot, ocena in zip(przedmioty, oceny):
    wyniki += f"{przedmiot} - {ocena}\n"
print(wyniki)
```

Powyższy skrypt już można uruchomić i on zadziała! Została tylko jedna część: zapisać wartość zmiennej `wyniki` do pliku.

Ale przed tym, jak pracować z plikami, potrzebujemy uruchomić małą część kodu z poprzedniego zajęcia:

```
import os
from pathlib import Path

PROJECT_DIR = Path(r"Ścieżka/do/mojego/folder/")
OUTPUT_DIR = PROJECT_DIR.joinpath("results")
os.makedirs(OUTPUT_DIR, exist_ok=True)
```

11) Poniżej jest podany przykładowy skrypt zapisywania danych do pliku:

```
with open(OUTPUT_DIR.joinpath("moj_plik.txt"),
          mode="w",
          encoding="UTF-8") as moj_test:
    moj_test.write("Jakiś tekst.\nNowa linia tekstu.")
```

Na podstawie powyższego fragmentu, uzupełnij wszystkie pola `?_?_?` w poniższym skrypcie.

```
with ?_1_?(OUTPUT_DIR.?_2_?("oceny.txt"),
    mode="?_3_?",
    ?_4_?="UTF-8") as ?_5_?:
    plik_oceny.?_6_?(wyniki)
```

?_1_? - _____
?_2_? - _____
?_3_? - _____
?_4_? - _____
?_5_? - _____
?_6_? - _____

12) Poniżej znajduje się szablon dowolnej funkcji w Pythonie.

```
def some_function(parameter1, parameter2, parameter3="I'm
```

```

optional!"):
    # ...
    # Any code here
    # ...

    # Just to show, how parameters work:
    print("parameter1 = ", parameter1)
    print("parameter2 = ", parameter2)
    print("parameter3 = ", parameter3)

    # We can return some value from function with return keyword:
    return "I was returned!"

```

Korzystając z powyższego szablonu, uzupełnij wszystkie pola ?__? w poniższym skrypcie.

```

>>> ?_1_? ?_2_?(a, ?_3_?):
...     return a + b
...
>>> dodaj(2)
3
>>> dodaj(3)
4
>>> dodaj(3, 5)
8

```

?_1_? - _____
 ?_2_? - _____
 ?_3_? - _____

13) Zadanie 3

1. Napisać funkcję postaci `func(x, y, c=0)` w Pythonie, która odpowiada poniższej funkcji matematycznej:

$$f(x, y) = \frac{\sin\left(\sqrt{x^2 + y^2 + c^2}\right)}{\sqrt{x^2 + y^2 + c^2}}$$

2. Żeby sprawdzić poprawność napisania funkcji podaj wynik następnego kodu:

`round(func(1, 1, 1), 3)` _____

3. Używając tej funkcji narysować wykres za pomocą jednego z podanych poniżej sposobów.

Część wspólna dla obu sposobów:

```
import numpy as np

c = 1e-5
limits = 25

v_func = np.vectorize(func)
xs = np.arange(-limits, limits, 0.25)
ys = np.arange(-limits, limits, 0.25)
xs, ys = np.meshgrid(xs, ys)
zs = v_func(xs, ys, c)
```

Pierwszy sposób: implementacja w Matplotlib

```
import matplotlib.pyplot as plt
from matplotlib import cm

fig, ax = plt.subplots(subplot_kw={"projection": "3d"}, figsize=(16, 9))

surf = ax.plot_surface(xs, ys, zs, cmap=cm.coolwarm,
                      linewidth=0, antialiased=True)
```

Drugi sposób: implementacja w Plotly

```
import plotly.graph_objects as go

fig = go.Figure(data=[go.Surface(x=xs, y=ys, z=zs)])

fig.update_layout(width=1000, height=600, scene_aspectmode='manual',
                  scene_aspectratio=dict(x=4, y=4, z=1))
```

14) W celu ekonomii czasu, po prostu przekopiujemy poniższe linijki:

```
import os
import subprocess
from pathlib import Path

PROJECT_DIR = Path(r"Ścieżka/do/mojego/folder/")
OUTPUT_DIR = PROJECT_DIR.joinpath("results")
os.makedirs(OUTPUT_DIR, exist_ok=True)
RESOURCES_DIR = PROJECT_DIR.joinpath("resources")
SPR_DIRS = (
    RESOURCES_DIR.joinpath("sprawozdanie-1"),
    RESOURCES_DIR.joinpath("sprawozdanie-2"),
    RESOURCES_DIR.joinpath("sprawozdanie-3")
)
```

Sprawdźmy poprawność wykonania powyższego kroku, poprzez wykonanie następnego polecenia:

```
len(os.listdir(SPR_DIRS[2]))
```

Proszę podać otrzymaną liczbę: _____

15) Szablon kodu LaTeX:

```
\documentclass{article}

% Page margins
\usepackage{geometry}
\newgeometry{inner=2.54cm, outer=2.54cm, top=2.54cm,bottom=2.54cm,
bindingoffset=0cm}

% Language support (date, contents, etc.)
\usepackage[polish]{babel}
\usepackage{polski}
\usepackage[utf8]{inputenc}

% For images (\includegraphics)
\usepackage{graphicx}

\begin{document}
\begin{figure}
  \centering
  \includegraphics[width=0.8\textwidth]{ścieżka/do/zdjęcia}
  \caption{%(figure_caption)s}
\end{figure}

\end{document}
```

Wersja 1

Podaj swój wynik polecenia: `len(content)` _____

**ten wynik będzie nieco inny dla każdego*

16) Jaki jest wynik wykonania następnego polecenia?

`sum(map(len, command[:4]))`

17) Wersja 2

Podaj swój wynik polecenia: `len(content)`

**ten wynik będzie nieco inny dla każdego*

18) Wersja 3

Podaj swój wynik polecenia: `len(content)`

**ten wynik będzie nieco inny dla każdego*

19) Czy forma dzisiejszych zajęć jest lepsza, od formy poprzednich zajęć?

- A. Tak, jest lepiej
- B. Jest tak samo
- C. Nie, jest gorzej

Jeżeli masz jakieś propozycji lub uwagi, to możesz zostawić ich w poniższym polu:

Key:

1. D
2. E
3. B
4. input / C
5. A / B
6. import / random.randint(2, 5) / oceny.append(ocena)
7. C / D
8. C
9. zip(przedmioty, oceny)
- 10.
11. open / joinpath / w / encoding / plik_oceny / write
12. def / dodaj / b=1
13. 0.57
14. 36
- 15.
16. 48
- 17.
- 18.
19. B