Zaawansowane programowanie - lab 1
Kurs po Pythonie I:

Interpreter

Interpreter można uruchomić poprzez konsolę:

```
python --version
```

Pliki z rozszerzeniem “.py” uruchamiamy w następujący sposób:

```
python nazwa_pliku.py
```

Zmienne

Tekstowe	str
Numeryczne	int, float, complex
Sekwencyjne	list, tuple, range
Mapujące	dict
Zbiory	set, frozenset
Boolowskie	bool
Binarne	bytes, bytearray, memoryview
None Type	NoneType

Przypisywanie wartości:

```
zmienna_str = "Wartość"
print(zmienna_str, type(zmienna_str))

zmienna_liczbowa = 1
print(zmienna_liczbowa, type(zmienna_liczbowa))
```

Rzutowanie typów:

```
zmienna liczbowa = 1
zmienna_str = str(zmienna_liczbowa)
zmienna_float = float(zmienna liczbowa)

print(zmienna_liczbowa, type(zmienna_liczbowa))
print(zmienna_str , type(zmienna_str))
print(zmienna_float , type(zmienna_float))
```

Instrukcje warunkowe

Równy	==
Nierówny	!=
Mniejszy	<
Mniejszy bądź równy	<=
Większy	>
Większy bądź równy	>=

Instrukcje warunkowe `if`:

```
większe_od_zera = 1

if większe_od_zera  > 0:
      print(f"{większe_od_zera} jest większe od zera")
```

Instrukcje warunkowe `if` + `elif` + `else`:

```
większe_od_pięć = 6

if większe_od_pięć < 5:
      print(f"{większe_od_pięć} jest mniejsze od 5")
elif większe_od_pięć == 5:
      print(f"{większe_od_pięć} jest równe 5")
else:
      print(f"{większe_od_pięć} jest większe od 5")
```

Złożone instrukcje warunkowe `and` oraz `or` :

```
liczba = 5.5

if liczba > 5 and liczba <6 or liczba > -5 and liczba < -6:
      print(f"Pierwiastek z liczby {liczba} podniesionej do kwadratu
znajduje się pomiędzy 5 i 6")
```

Pętle

pętla `while` :

```
liczba = 6

while liczba < 10:
      print(liczba)
      liczba +=1
```

Pętla `while` + `else` :

```
liczba = 9

while liczba < 10:
      print(liczba)
      liczba +=1
else:
      print("Koniec pętli")
```

Pętla `for` + `range` :

```
for i in range(10):
      print(i)
```

Pętla for (foreach):

```
kolekcja = ['element_1', 'element_2', 'element_3']

for element in kolekcja:
      print(element)
```

Przerwanie pętli for lub while poprzez break :

```
for i in range(10):
      if i == 5:
            break
      print(i)
```

Pominięcie iteracji pętli for lub while poprzez continue :

```
for i in range(10):
      if i == 5:
            continue
      print(i)
```

Matematyka

Operatory matematyczne

Dodawanie	+
Odejmowanie	-
Mnożenie	*
Dzielenie	/
Dzielenie całkowite	//
Reszta z dzielenia	%
Potęgowanie	**

```
a, b = 5, 6

print(a % b)
```

Listy i listy składane:

Dodawanie elementów do listy z wykorzystaniem pętli `for` :

```
tekst = "Tekst zawierajacy bardzo duzo slow"
lista_slow = tekst.split()

dlugosc_slow = []

for slowo in lista_slow:
      dlugosc_slow.append(len(slowo))

print(tekst)
print(dlugosc_slow)
```

Dodawanie elementów bezpośrednio do listy składanej:

```
tekst = "Tekst zawierajacy bardzo duzo slow"
lista_slow = tekst.split()

dlugosc_slow = [len(slowo) for slowo in lista_slow]

print(tekst)
print(dlugosc_slow)
```

Funkcja

```
def nazwa_funkcji(parametr1, parametr2):
      return parametr1 + parametr2

print(nazwa_funkcji(2, 3))
```

Krótkie 95 zadań z Pythona na stronie W3Schools: ZADANIA

Zadania do realizacji:

Uwaga proszę każde zadanie umieścić w osobnych plikach o nazwie:
`zadanie_X_Imie_Nazwisko.py`
następnie spakować pliki do archiwum: `lab_1_Imie_Nazwisko.zip`

Zadanie 1. Wyszukiwanie najdłuższego imienia
Przykładowe dane: `["Anna", "Krzysztof", "Marek", "Ewa", "Tomasz", "Aleksandra", "Piotr", "Magdalena"]`
Napisz funkcję, która przyjmie w parametrze listę 8 imion, a następnie zwróci imię o największej liczbie znaków.

Zadanie 2. Filtrowanie liczb parzystych i ich kwadraty
Napisz funkcję, która przyjmuje listę 15 liczb (użyj `range` do stworzenia tej listy), a następnie zwróci nową listę, która będzie zawierać kwadraty jedynie liczb parzystych.

Zadanie 3. Odwracanie słów
Przykładowe dane: `["programowanie", "Python", "zadanie", "student", "klasa"]`
Napisz funkcję, która przyjmie listę 5 słów, a następnie zwróci nową listę, w której każde słowo będzie zapisane od tyłu.

Zadanie 4. Sumowanie elementów na pozycjach nieparzystych
Napisz funkcję, która przyjmuje listę 12 liczb i zwróci sumę elementów znajdujących się na pozycjach nieparzystych.

Zadanie 5. Filtrowanie imion na literę “P”
Przykładowe dane: `["Emilia", "Krzysztof", "Ela", "Marek", "Edward", "Ewa", "Zbigniew", "Anna", "Eryk", "Ola"]`
Napisz funkcję, która przyjmie w parametrze listę 10 imion i zwróci listę, zawierającą tylko imiona zaczynające się na literę “E” (bez względu na wielkość litery, czyli “E” lub “e”).