

Lab 3

Instrukcje warunkowe cd. and, or :

```
temperature = float(input("pomiar temperatury: "))
if temperature < 28 or temperature > 41:
    print("Uwaga! zagrożenie życia!")
else:
    print("temperatura: "+str(temperature))

pressure = "180/110"
symptom = "ból w klatce piersiowej"
if pressure == "180/110" and symptom == "ból w klatce piersiowej":
    print("Natychmiastowe wezwanie pogotowia !")
else:
    print("...")
```

Pętle:

<https://docs.python.org/3/tutorial/datastructures.html>

while:

Nieskończona pętla + warunek:

```
while True:
    dane = input("podaj dane: (koniec - kończy działanie programu) ")
    if dane == "koniec":
        break
    else:
        print(dane)
```

skończona pętla:

```
i = 0
while i < 10:
    print(i)
    i += 1

i = 0
while i < 10:
    if i % 4 != 0:
        break
    print(i)
    i += 1
```

for:

```
for i in range(0, 10):
    print("FOR"+str(i))
```

```
for i in range(0, 10, 2):
    print("2FOR"+str(i))
```

```
for i in range(0, 10, 2):
    if i == 6:
        break
    print("FOR"+str(i))
```

```
for i in range(0, 10, 2):
    if i == 6:
        continue
    print("FOR"+str(i))
```

for oraz lista:

```
moja_lista = ["Warszawa", "Kraków", "Wrocław", "Łódź", "Poznań",  
             "Gdańsk", "Szczecin", "Bydgoszcz", "Lublin", "Białystok"]  
  
for index in range(0, len(moja_lista)):  
    print(moja_lista[index])
```

UWAGA !!! Lista indeksowana jest od 0.

Zmiana elementu:

```
moja_lista[index] = "New York City"
```

dodanie elementu na koniec:

```
moja_lista.append("Berlin")
```

Wstawianie elementu pod zadany indeks i przesunięcie pozostałych elementów:

```
moja_lista.insert(0, "Bangkok")
```

['Bangkok', 'Warszawa', 'Kraków', 'Wrocław', 'Łódź', 'Poznań', 'Gdańsk', 'Szczecin', 'Bydgoszcz', 'Lublin', 'New York City', 'Berlin']

Usunięcie elementu z listy znając indeks:

```
del moja_lista[0]  
['Warszawa', 'Kraków', 'Wrocław', 'Łódź', 'Poznań', 'Gdańsk', 'Szczecin', 'Bydgoszcz', 'Lublin', 'New  
York City', 'Berlin']
```

Usunięcie i zwrócenie elementu:

z „góry”:

```
print(moja_lista.pop())
```

Z indeksem:

```
print(moja_lista.pop(0))
```

Berlin

Warszawa

Usunięcie po wartości (usuwa pierwszy element):

```
moja_lista.remove("Poznań")
```

['Kraków', 'Wrocław', 'Łódź', 'Gdańsk', 'Szczecin', 'Bydgoszcz', 'Lublin', 'New York City']

Dodanie nowej listy do istniejącej:

```
moja_lista.extend(["nowe", "dane"])
```

['Kraków', 'Wrocław', 'Łódź', 'Gdańsk', 'Szczecin', 'Bydgoszcz', 'Lublin', 'New York City', 'nowe', 'dane']

Zwrócenie indeksu pierwszego wystąpienia elementu:

```
print(moja_lista.index("nowe"))
```

8

Dla przedziału:

```
print(moja_lista.index("nowe", 0, 8))
```

Utworzenie listy z przedziału oraz przejście przez elementy listy:

```
liczby = list(range(1, 11, 1))  
for element in liczby:  
    print(element)
```

Usunięcie wszystkich elementów o zadanej wartości z listy:

```
moja_lista.add("nowe")  
tekst="nowe"  
while tekst in moja_lista:  
    moja_lista.remove(tekst)  
print(moja_lista)
```

Funkcje len, max, min, sum zwracają kolejno liczbę elementów, wartość maksymalną, minimalną oraz sumę elementów:

```
print(len(liczby))  
print(max(liczby))  
print(min(liczby))  
print(sum(liczby))
```

Zwrócenie liczby pseudolosowej z przedziału:

```
import random  
random.randint(0, 10)
```

10

konwersja łańcucha znaków na listę przy użyciu separatora:

```
cities = "Warszawa,Kraków,Wrocław,Łódź,Poznań,Gdańsk,Szczecin,Bydgoszcz,Lublin,Białystok"  
cities_list = cities.split(",")
```

Zadania:

1. Napisz program, który pobierze od użytkownika liczby rozdzielone przecinkiem a następnie policzy znajdzie ich max oraz min, bez używania wbudowanych funkcji. (20%)
2. Napisz program, który wyświetli plan wycieczki – wybierając losowo z listy 10 największych miast w Polsce miasta do odwiedzenia. Miast ma być 10, nie mogą się powtarzać. (20%)
3. Napisz grę – papier nożyce kamień. (60%)
 - Gra pozwoli wybrać ilość rund.
 - Pozwoli wybrać tryb gry – komputer / 2 graczy (hot seats)
 - Pozwoli nazwać graczy.
 - Zapamięta wynik z każdej rundy.
 - Na koniec wyświetli listę wyników oraz ostateczny wynik z informacją, który gracz wygrał.

W opcji hot seats należy użyć getpass:

```
import getpass  
choice = getpass.getpass('wybór: ')  
print(choice)
```

Zagraj w trybie hot seats ze znajomymi.

UWAGA !! Aby getpass zadziałała w IDE PyCharm – przy uruchamianiu skryptu należy zaznaczyć emulację terminala w Edit Configurations:

