



**UNIWERSYTET
WSB **MERITO**
WARSZAWA**

Python developer

Agenda

1. Czego używamy?
2. Print & Input
3. Typy zmiennych
4. Instrukcje warunkowe
5. Pętle
6. Praktyka

Czego używamy?

- Cudzysłowy (tzw. ciapki) zamiennie, pojedyncze bądź podwójne, aby wyświetlić tekst,
- Nawiasy () [] {}
- Wcięcia / tabulacje aby wskazać, która część kodu powinna się wykonać
- Znak # komentarz, program nie czyta takiej linii

Print & Input

Print – jak wyświetlić napis?
Input – jak wprowadzić dane?

```
print('Pierwszy program')
```

```
a = 5
```

```
b = 7
```

```
napis = 'Python'
```

```
print(a)
```

```
print(a+b)
```

```
print('suma a i b wynosi: ',a+b , ' a ty lubisz jezyk , napis')
```

```
imie = input('Podaj imie ')
```

```
print('Witaj' , imie)
```

Typy zmiennych

```
a = 5
b = 5.5
c = 'mama'
d = ['mama', 'kot', 5, 5.5]
```

```
print('Zmienna a jest typu ', type(a))
print('Zmienna b jest typu ', type(b))
print('Zmienna c jest typu ', type(c))
print('Zmienna d jest typu ', type(d))

print('Pierwszy element listy to ', d[0])
```

```
#int - liczba całkowita
#float - liczba zmiennoprzecinkowa
#string - napis
#list - lista elementów
```

```
#wypisanie typu zmiennej 'a'
```

Typy zmiennych

```
x = input('Ile masz lat? '))  
y = int(input('jeszcze raz, ile masz lat? '))
```

```
print(type(x))  
print(type(y))
```

```
#if x > 18:  
#    print('Jestes pelnoletni')
```

#błąd, x nie jest liczbą

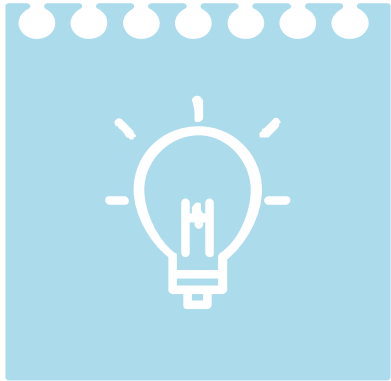
```
if int(x) > 18:  
    print('Jestes pelnoletni')
```

```
if y > 18:  
    print('Jestes pelnoletni')
```

Instrukcja warunkowa „if”

```
print('Witamy w programie')           #wypisanie napisu na ekranie
x = (input('Ile masz lat? '))         #wypisanie napisu, oczekiwanie na wartość
x = int(x)                            #zapisanie wartości jako liczba całkowita
if x < 18:                             #sprawdzenie warunki
    print('Nie masz 18 lat. Nie masz dostępu')
elif x == 18,                          #sprawdzenie drugiego warunku
    print('Masz dostep, ale zalecamy ostroznosc')
else:                                  #akcja, gdy warunki nie zostały spełnione
    print('Witamy')
```

Instrukcja warunkowa „if”



01

Napisz program, który spyta użytkownika o zarobki brutto oraz o liczbę dzieci, a następnie obliczy, ile użytkownik dostanie pieniędzy netto.

- a) podatek wynosi 20%, a na każde dziecko przysługuje 800 zł
- b) podatek wynosi 10%, gdy zarabiamy powyżej 3000 zł i 20%, gdy zarabiamy powyżej 5000 zł. Program 800+ przysługuje tylko na drugie i trzecie dziecko

02

Napisz program, który zapyta o wymiary mieszkania oraz o typ mieszkania i wypisze miesięczny koszt ogrzania mieszkania.

- a) Mieszkanie z cegły - ogrzanie 1m^3 to 30gr dziennie
- b) Mieszkanie z płyty - ogrzanie 1m^3 to 50gr dziennie

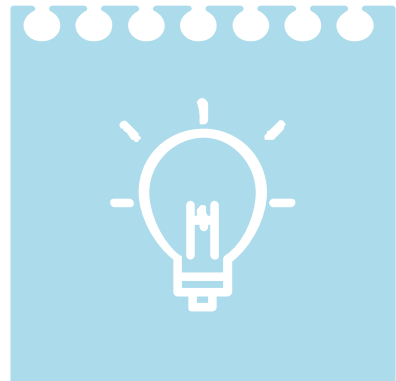
Pętla for

```
print('Witamy w programie')
x = int(input('Ile razy chcesz wykonać pętlę? '))

for i in range (x):
    print('To jest ', x, ' wykonanie')
    print('Lecimy dalej')
print('Koniec pętli')
```

#wykonanie akcji „x” razy
#wszystkie linie z wcięciem
#wykonają się w petli

Pętla for



01

Dana jest lista liczb całkowitych

```
Lista = [1, 5, 3, 24, 15, 6, 8, 12, 31]
```

1. Napisz program, który wyświetli po kolei wszystkie liczby razem z ich miejscem w liście
2. Napisz program, który wyświetli tylko liczby parzyste LUB liczby większe od 10

02

Dana jest lista nazw użytkowników

```
uzytkownicy = ['kamil', 'marek', 'mandarynka98',  
               'nowy1', 'pixi']
```

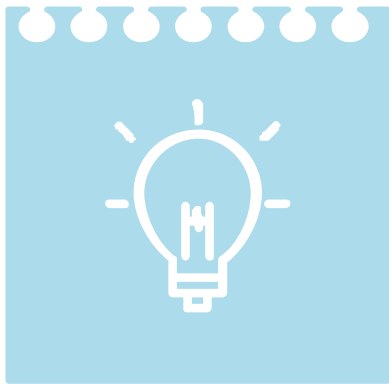
Napisz program, który zapyta o nazwę użytkownika oraz poinformuje, czy nazwa użytkownika jest wolna, a jeśli nie, to poinformuje, na którym miejscu w liście jest zapisany dany użytkownik.

Pętla while True

```
while True:
    password = input('Wpisz hasło: ')
    if password == '123abc':
        print('Hasło przyjęte')
        break
    else:
        print('Złe hasło, jeszcze raz')
```

pętla będzie się wykonywać,
#aż do polecenia „break”

Pętla while True



01

Gra "zgadnij jaką liczbę mam na myśli"?

- Program losuje liczbę z przedziału $<1, 100>$
- Program pyta użytkownika o liczbę i mówi, czy podana liczba jest większa, mniejsza bądź równa wylosowanej.

Zadanie

Co jest bardziej prawdopodobne:

- rzut 1 kostką, gdzie sukces to 4, 5, 6
- rzut 3 kostkami, gdzie sukces to przynajmniej 1 szóstka?

Jak policzyć, co z tych dwóch opcji ma większą szansę na sukces?

- *Można analitycznie... ?*
- *A można napisać program, który rzuci kostkami 1000 razy, a następnie przedstawi wyniki oraz porówna wyniki?*

Zadanie

1. Sprawdźmy, czy dane słowo jest anagramem.
Anagram to: axa, kajak, owo.
2. Napisz program, który zamienia wszystkie wykrzykniki na kropki.
3. Sprawdźmy, czy dane hasło jest silne. Jeśli nie – dlaczego?

Rodzaje Zmiennych (Typy Danych)

Podstawowe typy danych:

int – liczby całkowite, np. 5, -10, 42

float – liczby zmiennoprzecinkowe, np. 3.14, 0.5, -12.3

str – łańcuchy znaków (stringi), np. "hello", 'Python'

bool – wartości logiczne, tylko dwie możliwe: True i False

Typy złożone:

list – lista (kolekcja), np. [1, 2, 3], może zawierać różne typy elementów

tuple – krotka (niezmienialna lista), np. (1, 2, 3)

dict – słownik (mapowanie klucz-wartość), np. {'klucz': 'wartość', 'imie': 'Anna'}

set – zbiór, np. {1, 2, 3}, przechowuje unikalne elementy

Specjalne typy danych:

NoneType – specjalna wartość None, która oznacza brak wartości lub pusty wskaźnik

Operacje na Zmiennych

Operacje arytmetyczne:

Dodawanie: +

Odejmowanie: -

Mnożenie: *

Dzielenie: / (wynik zawsze jest typu float)

Dzielenie całkowite: // (daje wynik całkowity)

Reszta z dzielenia: %

Potęgowanie: **

Operacje na stringach

Konkatenacja: + (łączenie stringów, np. "hello" + " world")

Powielanie: * (np. "ha" * 3 da "hahaha")

Indeksowanie i wycinanie: np. "Python"[0] zwróci 'P', a "Python"[0:3] zwróci 'Pyt'

Operacje porównania

Równość: $==$

Nierówność: $!=$

Większy niż: $>$

Mniejszy niż: $<$

Większy lub równy: $>=$

Mniejszy lub równy: $<=$

Operacje logiczne

- **i (AND): and** (zwraca True, jeśli oba wyrażenia są prawdziwe)
- **lub (OR): or** (zwraca True, jeśli przynajmniej jedno wyrażenie jest prawdziwe)
- **nie (NOT): not** (odwraca wartość logiczną wyrażenia)

Słowa kluczowe

and, or, not – operatory logiczne

if, elif, else – instrukcje warunkowe

for, while – pętle

break, continue – instrukcje sterujące pętlami

def – definiowanie funkcji

return – zwraca wartość z funkcji

class – definiowanie klasy

import – importowanie modułów

try, except, finally – obsługa wyjątków

with – używane przy zarządzaniu zasobami, np.
otwieranie plików

lambda – tworzenie funkcji anonimowych

True, False, None – specjalne wartości logiczne i
brak wartości

Operacje na listach

append() – dodaje element na końcu listy

remove() – usuwa pierwszy napotkany element o podanej wartości

pop() – usuwa element z podanej pozycji (domyślnie ostatni)

sort() – sortuje listę

reverse() – odwraca kolejność elementów