

# Zbiór zadań 1

Po każdym z zadań należy wywołać utworzony element przez ich zwizualizowanie funkcją "print()" oraz "print(type())" aby sprawdzić czy wszystko jest zrobione poprawnie.

## Zadanie 1

Stwórz listę która składa się z 15 elementów nazwij ją "l1". Lista powinna zawierać wyłącznie wartości tekstowe.

## Zadanie 2

Stwórz listę która składa się z 5 elementów nazwij ją "l2". Lista powinna zawierać wyłącznie wartości numeryczne (floaty- np. 5.0).

## Zadanie 3

Stwórz listę która składa się z 20 elementów nazwij ją "l3". Lista powinna składać się z:

- 6 wartości numerycznych (intigery-np. 1)
- 4 wartości numerycznych (floaty-np. 2.0)
- 5 wartości booleanowych (Boolean - np. True)
- 5 wartości tekstowych

## Zadanie 4

Sprawdź typy danych poniższych elementów i zapisz je kolejno pod zmiennymi a,b,c,d

- "test"
- 1.0
- 9090
- False

Do tego zadania potrzebna będzie funkcja "type()"

Po zapisaniu typów danych zwizualizuj je jeden pod drugim w jednej komórce. Do tego zadania potrzebna będzie funkcja "print()"

## Zadanie 5

Stwórz dwa słowniki słownik:

- jeden słownik składa się z 5 nazw zwierząt - nazwij go "zwierzeta"
- drugi słownik składa się z 5 wartości wagowych odpowiadającym słownikowi "zwierzeta" i nazwij go "wagi" (wagi mogą być dalekie od prawdy)

Zwizualizuj plisty przy pomocy funkcji "print()" oraz sprawdź ich typy przy pomocy funkcji "type"

## Zadanie 6

Zreplikuj listę "l1" 5 razy (przemnóż ją 5 razy) i nadpisz ją pod nową nazwą "l\_replikacja"

## Zadanie 7

Zamień listę "l\_replikacja" na set. Do tego będzie potrzebna funkcja "set()" i zapisz ją pod zmienną "l\_replikacja\_na\_set" Sprawdź typ.

## Zadanie 8

Stwórz tupla który składa się z 15 elementów liczbowych i nazwij go "t1". W tuplu powinny być dowolnych danych w różnych typach z czego przynajmniej 3 muszą się powtarzać.

## Zadanie 9

Zamień tupla "t1" na set i zapisz go pod nazwą "tuple\_na\_set". Do tego będzie potrzebna funkcja "set()"

## Zadanie 10

Stwórz instrukcję logiczną/ warunkową która (nie funkcje):

- będzie zawierać 5 progów upustu dla cen (10%,20%,30%,33%,35% - wpisujemy je jako stringi)
- upusty mają być przyznawane schodkowo wg. ilości (100,300,500,600,700 szt - np. 10% dla ilości => 100 i ilości < 300)
- w przypadku wyjścia za skalę ilości upustów powinien pojawić się dodatkowo komunikat "Osiągnięto maksymalny pułap upustów" wraz z maksymalnym upustem
- przed funkcja zadeklaruj pod zmienna "ilosc" ilość sztuk (typ float lub intiger)
- sprawdź czy wszystkie progi wywołują się poprawnie
- upust powinien pojawiać za pomocą funkcji "print()"
- w przypadku braku upustu powinien pojawić się komunikat "Brak upustu"

Do wykonania zadania będą potrzebne:

- operator "and"
- składnia instrukcji warunkowej składającej się z "if", "elif" i "else"
- odpowiednia intencja - czyli wcięcia i odstępy
- funkcja "print()"
- dla dodatkowego komunikatu musimy wykorzystać zapis "Jakiś tekst {}".format(x)

## Zadanie 11

Stwórz funkcję kalkulacyjną która będzie składać się z 3 elementów:

- cena
- ilość
- upust

Funkcja powinna nazywać się "kalkulator\_upustów" a jej działanie powinno być wynikiem mnożenia wszystkich elementów przez siebie. Sprawdź działanie funkcji na losowych wartościach numerycznych.

Do wykonania zadania będzie potrzebny:

- zapis funkcji składający się z "def" i "return"
- odpowiednia intencja - czyli wcięcia i odstępy

## Zadanie 12

Stwórz funkcję o nazwie "przypisanie\_zwierząt\_do\_wag" która będzie pobierać dane z 2 słowników "zwierzęta" i "wagi". Funkcja ma wyświetlać zwierze i jego wagę w zależności od numeru zwierzęcia które zostanie wpisane (1-5). Funkcja powinna przyjmować tylko jeden argument. Funkcja powinna przyjmować tylko jeden argument. Sprawdź działanie funkcji na wartościach z przedziału 1-5.

Do wykonania zadania będzie potrzebny:

- zapis funkcji składający się z "def" i "return"
- odpowiednia intencja - czyli wcięcia i odstępy
- funkcja "get()"

## Zadanie 13

Wykonaj pętlę "while" 51 razy, pętla powinna wyświetlać wymienione wartości w zależności od kolejności wykonania:

- "poniżej 10"
- "poniżej 20"
- "poniżej 30"
- "poniżej 40"
- "poniżej 50"
- "powyżej 50"

Do wykonania zadania będzie potrzebne:

- zmienna która przyjmie wartość 0 np. a=0
- polecenie pętli "while"
- pamiętamy o dodaniu parametru wyjścia np. < lub > 10000, aby funkcja nie zapętlila się bez końca
- pamiętamy aby wewnątrz pętli iterować parametr wykonania np. a= jakieś działanie
- funkcja "print()"
- operator logiczny "and"
- funkcje warunkowe
- składnia instrukcji warunkowej składającej się z "if", "elif" i "else"
- odpowiednia intencja - czyli wcięcia i odstępy

## Zadanie 14

Stwórz funkcję o nazwie "kalkulator" która będzie składać się z 3 argumentów:

- wartość1
- działanie
- wartość2

Argument "działanie" może przyjmować wyłącznie wartości tekstowe:

- dodawanie
- odejmowanie
- mnożenie
- dzielenie

Po wpisaniu odpowiedniego działania funkcja powinna dodawać, odejmować, mnożyć i dzielić "wartość1" oraz "wartość2". We wszystkich działaniach w funkcji "wartość1" powinna być pierwsza. Przykład kalkulator(2,"dodawanie",2) = 2 + 2 = 4. Sprawdź działanie funkcji na wartościach różnych od 0.

Do wykonania zadania będzie potrzebne:

- składnia instrukcji warunkowej składającej się z "if", "elif" i "else"
- odpowiednia intencja - czyli wcięcia i odstępy
- zapis funkcji składający się z "def" i "return"

## Zadanie 15

Do funkcji "kalkulator" dodaj obsługę dzielenia przez 0 i nazwij ją "kalkulator\_zabezpieczony". W przypadku wystąpienia 0 w "wartość1" i "wartość2" funkcja powinna pokazać komunikat "Nie można wykonać dzielenia przez 0".

Do wykonania zadania będzie potrzebne:

- skopiowanie funkcji "kalkulator"
- składnia instrukcji warunkowej składającej się z "if", "elif" i "else"
- odpowiednia intencja - czyli wcięcia i odstępy
- zapis funkcji składający się z "def" i "return"
- funkcja "print()"

## Zadanie 16

Zabezpiecz funkcję "kalkulator\_zabezpieczony" przed niespodziewanymi argumentami lub błędami. W przypadku błędnych wartości np. tekstu w wartościach lub błędnego typu działania funkcja powinna zwrócić komunikat "Coś jest nie tak". Przetestuj funkcję wprowadzając wartości tekstowe zamiast liczbowych i działania które nie są przewidziane.

Do wykonania zadania będzie potrzebne:

- skopiowanie funkcji "kalkulator\_zabezpieczony"
- składnia instrukcji warunkowej składającej się z "if", "elif" i "else"
- odpowiednia intencja - czyli wcięcia i odstępy
- zapis funkcji składający się z "def" i "return"
- funkcja "print()"
- zapis podstawowej obsługi błędów "try" i "except"

In [ ]: