

Ćwiczenia do wykonania lokalnie:

Te ćwiczenia wystarczy wykonać lokalnie w programie IDLE, lub w swym wybranym środowisku IDE. Nie trzeba ich wrzucać na repozytorium na GitHub.

Listy (list)

- 1). Wywołaj polecenie `bool([])`
- 2). Stwórz zmienną `lista` i przypisz do niej wartość `[1, True, 2.0, 'String', []]`
- 3). Wywołaj polecenie `lista.append('element')`
- 4). Wywołaj polecenie `len(lista)`
- 5). Wywołaj polecenie `lista.index('element')`
- 6). Wywołaj polecenie `lista.insert(2, 'Wstawiony element')`
- 7). Wywołaj polecenie `lista[2]`
- 8). Wywołaj polecenie `lista[1:3]`
- 9). Wywołaj polecenie `lista[0:-1: 2]`
- 10). Wywołaj polecenie `lista[::-1]`
- 11). Stwórz zmienną `lista_liczb` i przypisz do niej wartość `[13,9, 27, 14, 15]`
- 12). Wywołaj polecenia `max(lista_liczb)`, `min(lista_liczb)`, `sum(lista_liczb)`
- 13). Wywołaj kolejno polecenia: `print(lista_liczb)`, `sorted(lista_liczb)`, `print(lista_liczb)`
- 14). Wywołaj kolejno polecenia: `print(lista_liczb)`, `lista_liczb.sort()`, `print(lista_liczb)`
- 15). Stwórz zmienną `lista_stringow` i przypisz do niej wartość `['c', 'd', 'b', 'e', 'a']` i powtórz polecenia z punktów 12-14 na utworzonej liście.
- 16). Wywołaj kolejno polecenia `print(lista_stringow)`, `lista_stringow[0]='Zmodyfikowany element'`, `print(lista_stringow)`
- 17). Wywołaj kolejno polecenia `print(lista_stringow)`, `del lista_stringow[0]`, `print(lista_stringow)`
- 18). Wywołaj polecenie `for string in lista_stringow: print(string)`
- 19). Wywołaj polecenie `for i in range(len(lista_stringow)): print(lista_stringow[i])`

Krotki (tuple)

- 1). Wywołaj polecenie `bool(())`
- 2). Stwórz zmienną `krotka` i przypisz do niej wartość `(1, True, 2.0, 'String', [])`
- 3). Spróbuj wywołać polecenie `krotka.append('element')` i zapoznaj się z błędem
- 4). Wywołaj polecenie `len(krotka)`
- 5). Wywołaj polecenie `krotka.index('String')`
- 6). Spróbuj wywołać polecenie `krotka.insert(2, 'Wstawiony element')` i zapoznaj się z błędem
- 7). Wywołaj polecenie `krotka[2]`
- 8). Wywołaj polecenie `krotka[1:3]`
- 9). Wywołaj polecenie `krotka[0:-1: 2]`
- 10). Wywołaj polecenie `krotka[::-1]`
- 11). Stwórz zmienną `krotka_liczb` i przypisz do niej wartość `(13,9, 27, 14, 15)`
- 12). Wywołaj polecenia `max(krotka_liczb)`, `min(krotka_liczb)`, `sum(krotka_liczb)`
- 13). Wywołaj kolejno polecenia: `print(krotka_liczb)`, `sorted(krotka_liczb)`, `print(krotka_liczb)`
- 14). Spróbuj wywołać polecenie `krotka_liczb.sort()` i zapoznaj się z błędem

- 15). Stwórz zmienną `krotka_stringow` i przypisz do niej wartość ('c', 'd', 'b', 'e', 'a') i powtórz polecenia z punktów 12-14 na utworzonej liście.
- 16). Spróbuj wywołać polecenie `krotka_stringow[0] = 'Zmodyfikowany element'` i zapoznaj się z błędem
- 17). Spróbuj wywołać polecenie `del krotka_stringow[0]` i zapoznaj się z błędem
- 18). Wywołaj polecenie `for string in krotka_stringow: print(string)`
- 19). Wywołaj polecenie `for i in range(len(krotka_stringow)): print(krotka_stringow[i])`

Słowniki (dict)

- 1). Wywołaj polecenie `bool({})`
- 2). Stwórz zmienną `dane_logowania` i przypisz do niej wartość {'Admin': '1234', 'User1': '12345'}
- 3). Wywołaj polecenia `dane_logowania['Admin']` oraz `dane_logowania.get('Admin')`
- 4). Wywołaj polecenia `dane_logowania['User2']` oraz `dane_logowania.get('User2')` i zapoznaj się z błędem w pierwszym wypadku.
- 5). Wywołaj polecenie `help(dict.get)` i zapoznaj się z komunikatem
- 6). Wywołaj polecenie `dane_logowania.get('User2', -1)` oraz `dane_logowania.get('User2', 'wartosc')`
- 7). Wywołaj polecenie `dane_logowania['User2'] = 'przykladowe_haslo'`
- 8). Wywołaj polecenia `print(dane_logowania.keys())`, `print(dane_logowania.values())`, `print(dane_logowania.items())`
- 9). Wywołaj polecenie `for item in dane_logowania.items(): print(item)`
- 10). Wywołaj polecenie `for key, value in dane_logowania.items(): print(key, value)`
- 8). Wywołaj polecenie `print(dane_logowania)`, `dane_logowania.pop('User2')`, `print(dane_logowania)`
- 9). Wywołaj polecenie `print(dane_logowania)`, `del dane_logowania('Admin')`, `print(dane_logowania)`
- 10). Wywołaj polecenie `if dane_logowania.get('User1'): print('Znaleziono uzytkownika')`
- 11). Wywołaj polecenie `if dane_logowania.get('User3'):`
 `print('Znaleziono uzytkownika')`
else:
 `print('Nie znaleziono uzytkownika')`
- 12). Stwórz zmienną `dane_uzytkownikow` i przypisz do niej wartość {'User1': {'name': 'Gall', 'surname': 'Anonim'}}
- 13). Wywołaj polecenie `dane_uzytkownikow.get('User1').get('name')`
- 14). Wywołaj polecenie `dane_uzytkownikow['User1']['surname']`

Zbiory (set)

- 1). Stwórz zmienne `lista1` oraz `lista2` oraz przypisz do nich wartości kolejno [1,2,2,2,3] oraz [2,3,4,4]
- 2). Stwórz zmienne `set1` oraz `set2` oraz przypisz do nich wartości kolejno `set(lista1)` oraz `set(lista2)`
- 3). Wywołaj polecenia `set1 | set2` oraz `set1.union(set2)`
- 4). Wywołaj polecenia `set1 & set2` oraz `set1.intersection(set2)`
- 5). Wywołaj polecenia `set1 - set2` oraz `set2 - set1`

- 6). Wywołaj polecenie `len(set1)`
- 7). Wywołaj polecenia `print(set1)`, `set1.add(4)`, `print(set1)`
- 8). Powtórz polecenia z punktu 7. Czy set się zmienił?
- 9). Wywołaj polecenia `print(set1)`, `set1.pop(4)`, `print(set1)`
- 10). Wywołaj polecenia `print(set1)`, `set1.remove(4)`, `print(set1)`
- 11). Wywołaj polecenia `print(set1)`, `set1.update(set2)`, `print(set1)`
- 12). Wywołaj polecenie `print(set2)`, `set2.clear()`, `print(set2)`

Skrypty które należy wrzucić na GitHub

Te skrypty muszą znaleźć się w repozytoriach najpóźniej do północy w następnym poniedziałek.

W zadaniach 1 oraz 2, zacznij od zdefiniowania zmiennej `lista = [1,2,3,4,5]`.

- 1). Napisz skrypt który odwróci kolejność elementów w tej liście i zapisze wynik tych operacji do zmiennej `odwrocona_lista`.

Oczekiwany wynik: `odwrocona_lista = [5,4,3,2,1]`

- 2). Napisz skrypt który znajdzie dwie największe liczby w liście i zapisze je pod postacią nowej listy w zmiennej `lista_max`.

Oczekiwany wynik: `lista_max = [4,5]`

- 3). Napisz skrypt który przy pomocy pętli `for` stworzy listę liczb od 3 do 13 włącznie, a także listę liczb od 0 do 10 włącznie przy pomocy pętli `while`. Zapisz wyniki w zmiennych o nazwie `lista1` oraz `lista2`.

Oczekiwany wynik: `lista1 = [3,4,5,6,7,8,9,10,11,12,13]`, `lista2 = [0,1,2,3,4,5,6,7,8,9,10]`

W zadaniu 4 zacznij od zdefiniowania zmiennej `dane_logowania = {'Admin': '1234'}`

- 4). Napisz skrypt który będzie przyjmować od użytkownika informację na temat nazwy użytkownika oraz hasła, a następnie porówna te wartości z danymi znajdującymi się w słowniku `dane_logowania`. Jeśli dane są poprawne, wydrukuj tekst 'Logowanie poprawne', jeśli nie, użytkownik powinien wprowadzić dane ponownie (Wykorzystaj pętle `while` oraz instrukcje warunkowe)

W zadaniu 5, zacznij od zdefiniowania dwóch zmiennych `lista1 = [1,2,3,4,5]`, `lista2 = [1,2,3,4,5]`, `lista3 = [1,2,2,3,3,4,5]`

- 5). Napisz skrypt który przy pomocy struktury danych `set`, zbada czy `lista1`, `lista2` oraz `lista3` zawierają te same elementy, oraz czy któraś z list zawiera duplikaty.

Poprawna odpowiedź: Wszystkie listy zawierają te same elementy, `lista3` zawiera duplikaty.