Komendy i krótkie opisy — lab10

`print(type(1))` : Zwraca typ obiektu (tu: `int`).

`print(type(1.0))` : Zwraca typ obiektu (tu: `float`).

`print(type('1.0'))` : Zwraca typ obiektu (tu: `str`).

`print(type(['1.0']))` : Zwraca typ obiektu (tu: `list`).

`first_list = [1,2,3,4,5]` : Tworzy listę przypisaną do zmiennej `first_list`.

`print(len(first_list))` : Zwraca długość listy.

`second_list = ['a', 5, 3.1415, [1,2]]` : Tworzy listę mieszanych typów (zagnieżdżoną).

`print(second_list)` : Wyświetla zawartość listy.

`second_list[0]`, `second_list[-1]` : Dostęp do elementów listy po indeksie (również indeksy ujemne).

`second_list[-1][0]` : Zagnieżdżone indeksowanie — dostęp do elementu wewnętrznej listy.

`matrix_2D = [[1,2], [3,4]]` : Tworzy listę list (macierz 2D).

`second_list[0] = 'b'` : Modyfikacja wartości elementu listy.

`first_list.append('1')` : `list.append(x)` — dodaje element `x` na koniec listy.

`first_list.extend(second_list)` : `list.extend(sekwencja)` — dołącza wszystkie elementy sekwencji.

`first.append(second)` : `append` doda całą listę jako pojedynczy element.

`first_list.count('b')` : `list.count(x)` — zlicza wystąpienia wartości `x`.

`first_list.insert(1, 'a')` : `list.insert(i, x)` — wstawia `x` na pozycję `i`.

`first_list.pop(1)` : `list.pop(i)` — usuwa i zwraca element spod indeksu `i`.

`first_list.remove('b')` : `list.remove(x)` — usuwa pierwsze wystąpienie wartości `x`.

`first_list.reverse()` : `list.reverse()` — odwraca kolejność elementów (in-place).

`first_list.sort()` : `list.sort()` — sortuje listę (in-place).

`second_list.index('b')` : `list.index(x)` — zwraca indeks pierwszego wystąpienia `x`.

`first_list.clear()` : `list.clear()` — usuwa wszystkie elementy listy.

`second_list = first_list` : Przypisanie referencji — nie tworzy kopii.

`first_list.copy()` : `list.copy()` — zwraca płytką kopię listy.

`[i for i in range(10)]` : List comprehension — tworzenie listy przez iterację.

`[i**2 for i in range(10) if i%2==0]` : List comprehension z warunkiem — filtrowanie elementów.

`[char for char in sentence if char in "ĄąĆćĘęŁłŃńÓóŚśŹźŻż"]` : Wyciąganie znaków spełniających warunek z łańcucha.

`first_tuple = ('a', 5, 3.1415, (1,2))` : Tworzenie krotki (immutable).

`first_tuple[0]`, `first_tuple[-1]` : Dostęp do elementów krotki.

`first_tuple[0] = 'b'` : Próba modyfikacji krotki — błąd (krotki są niemodyfikowalne).

`y = (1,)` : Krotka jednoelementowa (wymaga przecinka).

`seq[start:end]` (np. `first_list[1:3]`) : Slicing — pozyskiwanie podsekwencji.

`first_list[:]` : Pełne kopiowanie sekwencji (ekwiwalent `.copy()`).

`seq[start:end:step]` (np. `first_list[3:7:2]`) : Extended slicing z krokiem.

`seq[::step]` (np. `first_list[::2]`) : Wyciąganie elementów co `step`.

`seq[::-1]` (np. `first_string[::-1]`) : Odwrócenie kolejności elementów.

`empy_dict = {}` , `numbers = {'one':1}` , `colors = {...}` : Tworzenie słowników `dict`.

`colors['blue']` : Odczyt wartości po kluczu.

`colors['yellow'] = (255,255,0)` : Przypisanie wartości do klucza (dodanie/zmiana).

`[name for name in dir(dict) if not name.startswith('__')]` : Lista dostępnych metod `dict` bez metod magicznych.

`dict.values()`, `dict.keys()`, `dict.items()` : Zwracają widoki wartości, kluczy i par.

`{v:k for k,v in colors.items()}` : Dict comprehension — przykład przekształcenia słownika.