Python funkcje wbudowane

- Funkcje do pracy na tablicach:
  - count(x) – zwraca liczbę wystąpień elementu x w tablicy
  - extend(a) – dodaje do końca tablicy elementy z tablicy a <tablice muszą być tego samego typu!>
  - fromlist(lista) – dodaje do tablicy elementy z listy lista
  - index(x) – zwraca indeks pierwszego wystąpienia elementu x
  - insert(i, x) – wstawia do tablicy nowy element x umieszczając go na pozycji i
  - pop(i) – usuwa z tablicy element o indeksie i, brak argumentu w tej metodzie usuwa ostatni element tablicy
  - remove(x) – usuwa z tablicy pierwsze wystąpienie elementu x
  - reverse() – odwraca porządek elementów tablicy
  - tolist() – przekształca tablice na listę
  - append(x) – dodaje na koniec tablicy nowy element x
  - sort() – sortuje tablice, domyślne rosnąco <aby ustawić malejąco należy zastosować sort(reverse=True)
- Przydatne funkcje:
  - abs(x) – zwraca wartość bezwzględną liczby x
  - bin(x) – konwertuje liczbę całkowitą na ciąg znaków binarnych z prefixem „0b”
  - chr(x) – zwraca ciąg znaków dla podanej liczby według standardu Unicode np. chr(97) zwróci ‘a’
  - hex(x) – konwertuje liczbę całkowitą na heksadecymalną z prefixem ‘0x’
  - len(x) – zwraca długość obiektu x np. ciągu znaków, listy, tupla
  - max() – zwraca największy obiekt z pośród podanych argumentów, bądź z listy
  - min() – zwraca najmniejszy obiekt z pośród podanych argumentów, bądź z listy
  - oct(x) – konwertuje liczbę całkowitą x na system ósemkowy z prefixem ‘0o’
  - ord(c) – zamienia podany znakc na wartość liczbową według Unicode
  - pow(x, y) – podnosi liczbę x do potęgi y
  - print(x) – wyświetla obiekt x w terminalu

    - round(x, y) – zaokrągla liczbę x do y miejsc po przecinku
    - zip(a, b) – łączy obiekt a i b w tuple
    - tuple(*argumenty) – tworzy z podanych argumentów obiekt typu tuple, który wiąże ze sobą obiekty i nie pozwala na ich dalszą modyfikacje <przydaje się przy porządkowaniu danych)
- Obsługa plików
    - Do otwierania plików służy funkcje wbudowana open() jako argumenty podajemy nazwepliku i tryb działania.
    - Np. open(„plik.txt”, ‘r’) gdzie pierwszy atrybut to nazwa pliku a ‘r’ oznacza tryb odczytu (read).
    - Tryby otwarcia pliku: r – odczyt (domyślny tryb), w – zapis (nadpisuje cały plik), a – zapis (dopisuje nowe dane na koniec pliku)
    - Do operacji na pliku współcześnie używamy struktury with !
    - Przykład odczytu danych z pliku:

```
dane = ""
with open("plik.txt", "r") as plik:
    dane=plik.read()
```

    - Przykład zapisu danych do pliku:

```
dane = "moje dane"
with open("plik.txt", "w") as plik:
    plik.write(dane)
```

    - Funkcja split rozdziela nam dane na liste po podanym separatorze: dane.split(„#”)
    - Funkcje splitlines() rozdziela nam dane liste po podanym separatorze gdzie każdy element to osobna linijka
- Inne przydatne triki:
    - tab = [int(x) for x in tab] <- zamiana wszystkich elementów tablicy na liczby całkowite <zamiast int(x) można zastosować dowolną instrukcję!>