**Ustawienia pliku, start**

|  |  |
| --- | --- |
| #-\*-coding: utf-8-\*- | polskie znaki, kodowanie Unikod UTF-8 przy zapisywaniu, dodać komentarz jaki przyjęto tryb kodowania w pliku |

**Listy, krótkie, słowniki, tablice z haszowaniem**

|  |  |  |  |
| --- | --- | --- | --- |
| zmienna  STAŁA  dir(polecenie)  #komentarz  ‘’’ ‘’’ lub ””” ”””  \n  \"  Lista = [a,b,c]  Krotka = (a,b,c,d)  slownik = {'A1' : 'AA1','B2' : 'BB2'}  slownik = {'Imie':['Ania','Michał','Przemek'], 'Wiek':[18,25,40]}  slownik['B2']  slownik.add(‘aaa’)  slownik.update(dodawane\_elementy)  słownik.remove(usuwane\_elementy)  slownik["klucz"]  slownik.get("klucz", "tekst jak nie znajdzie klucza")  zmienna.upper()  zmienna.lower()  if zmienna isupper():  zmienna.index("a")  lista = lista + [6]  lista.append(6)  lista.append(“nowy element listy”)  lista.extend(druga\_lista)  nowa\_lista = stara\_lista.copy()  lista.insert(nr\_indeksu\_gdzie, "nowy\_element")    lista.clear()  lista.count("element")  lista.sort()  lista.reverse()  lista.insert(miejsce,"element")  lista[miejsce] = element  tekst.remove("co")  lista.pop()  del lista[które\_elementy]  tekst.replace("co", "naco")  tekst =’ciag znakow {} oraz {}  tekst.format  print(tekst.format(zm1, zm2))  hash\_table [znaczek] = 1  hash\_table [znaczek] += 1 | Piszemy małymi literami  Piszemy DUZYMY LITERAMI  wypisuje co można z danym poleceniem można zrobić i jaka jest składnia  komentarz wiele linii, trzy średniki  przejście do nowej linii w tekście  po ukośniku każdy znak jest traktowany jako tekst  lista – definicja w nawiasach  krótka to lista, której nie można zmieniać  słownik, nawiasy, elementy rozdzielane dwukropkiem  zapytanie o element słownika  dodanie elementu do słownika  dodanie listy elementów  usunięcie elementów ze słownika  zwraca wartość klucza ze słownika  zwraca wartość klucza ze słownika lub komunikat  zmiana na DUŻE LITERY  zmiana na małe litery  sprawdza czy duża LITERA  zwraca indeks/numer litery w tekście  dodanie elementu na koniec listy  dodaje do listy na końcu drugą listę  kopiuje wartości do nowej listy  wstawia w miejscu nowy element li.insert(2,’aa’)  usuwa wszystkie elementy z listy  liczy ile jest podanych elementów w liście  sortuje rosnąco listę  podaje listę od końca  dodanie elementu w miejscu lista.insert(2,’aa’)  zamiana elementu na danym miejscu  usunięcie wskazanego tekstu  usuwa ostatni element listy  usuwa elementy z listy, np. del.lista[2]  zmiana elementów co w naco  wpisanie w tekst wartości zmiennych   |  | | --- | | dodawanie elmentu do tablicy/słownika, w nawiasie 1 kolumna po równa się wartość | | zwiększanie elementu słownika | |

**Wyświetlanie na ekran, zczytywanie od użytkownika**

|  |  |
| --- | --- |
| print (zawartosc)  print('\n')  print(zawartosc, end=' ')  print('a','b','c',sep='……')  input() | wyświetlenie na ekran  pusta linia, pierwsze użycie to przejście do kolejnej linii  wartości rozdzielane znakami między apostrofami ‘znak’  separator między elementami  wprowadzenie wartości przez użytkownika |

**Zakresy zmiennej**

|  |  |
| --- | --- |
| "tekst" lub 'tekst'  tekst[indeks element lub zakres]  [0]  [-1]  [-3]  [:]  [:N]  [M:]  [N:M]  [::2]  [::-1]  str (zmienna\_numeryczna)  int (zmienna\_tekstowa)  float (zmienna\_tekstowa)  from math import \*  floor (liczba)  ceil (liczba)  sqrt (liczba) | ciąg tekstowy  określenie pozycji w liście  pierwszy element  ostatni element  trzeci od końca  wszystkie elementy  wszystkie elementy od 0 do N-1  wszystkie elementy od M (uwgz. Indeks 0)  elementy z zakresu N do M (start 0, wyś. M-1)  wszystkie co 2  wszystkie od końca do początku  zamienia na string  zamienia na liczbę całkowitą  zamienia na liczbę rzeczywistą z przecinkiem  zaokrągla do całkowitej w dół  zaokrągla do całkowitej w górę  pierwiastek kwadratowy |

**Operacje na tekstach, zmiennych i listach**

|  |  |
| --- | --- |
| + - \* /  a+=1 -= \*= /=  //  % np. i%2  2\*\*3 lub pow(2,3)  min(1,2,3,4,-2)  max(1,8,-10,5)  abs(-2)  round(liczba,dokładność)  len(tekst)  ‘ciag’ in tekst  ‘ciag’ not in tekst | operacje na liczbach  zmiana wartości zmiennej  dzielenie do wartości całkowitych  dzielenie, podanie reszty z dzielenia  potęgowanie 2^n  zwraca wartość minimalną  zwraca wartość maksymalną ze zbioru  zwraca wartość bezwzględną |-2|  zaokrąglenie z zadaną dokładnością, 2-miejsc po przecinku, -1 do pełnych 10, -2 do pełnych 100 itd., bez dokładności - wówczas z domyślną  długość zmiennej, np. ilość znaków  sprawdza zawartość, wyrzuca True/False  sprawdza brak zawartości, wyrzuca True/False |

**Warunki, pętle**

|  |  |
| --- | --- |
| if warunek: zdarzenie  elif warunek:zdarzenie  else:  == !=  and or and not not domyślnie True  for liczba in range(1,11):  for liczba in range(1,11,3):  for element in lista:  while warunek:  while True:  break  najmniejsza = None | prosty warunek  operatory równy, różny  łączenie warunków, lub  pętla, wartości od 1 do 10 bez ostatniej  pętla, krok co 3  przechodzi przez całą listę  pętla while  wyjście z pętli  zmianna pusta None |

**Funkcje**

|  |  |
| --- | --- |
| def NazwaFunkcji (parametr1, parametr2):  return  return zmienna  nowaZmienna = wynikFunkcji (a,b)  lambda <parametry> : <wyrażenie> | definiowanie funkcji  zakończenie funkcji  wynik funkcji jako zmienna  funkcja lambda |

**Różne biblioteki, wykorzystanie modułów**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | import time  time.sleep(1) | |  | | import random  random.choice(["orzeł", "reszka"])  import nazwa\_pliku  nazwa\_pliku.nazwa\_funkcji  from nazwa\_pliku import nazwa\_funkcji  nazwa\_funkcji | |  | | import biblioteki time  oczekiwanie 1 sek  losowanie z listy  wykorzystanie z modułów z innych plików  można zaimportować i używać nazw bez nazw plików |