

Temat: Programowanie w języku PYTHON

1. Standardowe funkcje:
 - a. int - zamiana tekstu na liczbę
 - b. str - zamiana liczby na tekst
 - c. abs – wartość bezwzględna
 - d. sqrt – pierwiastek
 - e. ord - funkcja zamieniająca znak na liczbę wg ASCII
 - f. chr - funkcja liczbę na znak wg ASCII
2. Instrukcja wejścia
 - a. a=int(input("Podaj a=")) - dla liczb całkowitych
 - b. a=float(input("Podaj a=")) - dla liczb rzeczywistych
 - c. a=input("Podaj a=") - dla danych tekstowych
3. Instrukcja wyjścia
 - a. print(a) - wyświetla wartość zmiennej a
 - b. print("a") - wyświetla tekst, czyli literę a
 - c. print("a=",a) - wyświetla tekst i wartość zmiennej a
4. Operatory arytmetyczne
 - a. + dodawanie
 - b. - odejmowanie
 - c. * mnożenie
 - d. / dzielenie
 - e. % reszta z dzielenia
 - f. ** potęgowanie
 - g. // dzielenie całkowite
5. Operatory inkrementacji
 - a. += zwiększenie o podaną wartość
 - b. -= zmniejszenie o podaną wartość
 - c. *= mnożenie przez podaną wartość
 - d. /= dzielenie przez podaną wartość
6. Operatory relacji:
 - a. == równe
 - b. > większe
 - c. >= większe lub równe
 - d. < mniejsze
 - e. <= mniejsze lub równe
 - f. != różne
7. Operatory logiczne
 - a. **and** koniunkcja, część wspólna zbiorów, i
 - b. **or** alternatywa, suma zbiorów, lub
 - c. **not** negacja, zaprzeczenie, nie
8. Instrukcja warunkowa, składnia:
if warunek:
 instrukcja1
else:
 instrukcja2

9. Pętla for – stosujemy w przypadku znanej ilości powtórzeń pętli
- Składnia:
for iterator in zakres lub lista:
instrukcje
.....
 - Iterator to liczba całkowita z zakresu <początek, koniec)
lub przyjmuje wartości z listy
10. Pętla while – stosujemy w przypadku nieznanego ilości powtórzeń pętli
- while warunek:
instrukcje
.....
11. Listy w języku Python
- Definicja listy: uporządkowany zbiór danych
 - Deklaracja pustej listy: nazwa_listy = []
 - Kasowanie listy: **del** lista
 - Uwaga: pierwszy element listy ma index **0**
 - Funkcje do obsługi listy:
 - wynik = **max**(lista) – największy wartość w liście
 - wynik = **min**(lista) – najmniejsza wartość w liście
 - wynik = **sum**(lista) – suma wartości elementów listy
 - lista2 = **sorted**(lista) – utworzenie posortowanej listy
 - wynik = **len**(lista) – długość listy
 - Metody obiektów klasy **list**
 - lista.**append**(wartość) - dodawanie elementu na koniec listy
 - lista.**insert**(index, wartość) - dodawanie elementu do listy
 - lista.**sort**() – sortowanie listy
 - lista.**reverse**() – odwrócenie kolejności elementów w liście
 - wynik = lista.**count**(wartość) – wyznaczenie ilości elementów o podanej wartości
 - Operatory:
 - x **in** lista – operator sprawdzający czy x znajduje się w liście
12. Odczyt z pliku tekstowego do listy
- ```
plik = open("liczby.txt")
lista = plik.readlines()
plik.close()
```
13. Zapis listy do pliku tekstowego
- ```
lista2 = [ ]  
for i in lista1:  
    lista2.append(int(i))
```
14. plik = open("liczby.txt","w")
- ```
for i in lista:
 plik.write(str(i))
 plik.write("\n")
plik.close()
```

## *Podstawowe funkcje*

```
def czy_pierwsza(n):
 if n==1:
 return False
 koniec=int(sqrt(n))
 for i in range(2, koniec+1):
 if n%i==0:
 return False
 return True
```

```
def nwd(a,b):
 while b>0:
 reszta=a%b
 a=b
 b=reszta
 return a
```

```
def suma_cyfr(n):
 wynik=0
 while n>0:
 reszta=n%10
 wynik+=reszta
 n//=10
 return wynik
```