

## Упражнение 2 – Съставяне, въвеждане и настройка на програми, илюстриращи операциите в езика и основните типове данни

---

### Пример 1 – първа програма

```
# first python code  
print('Hello world !!!')
```

Програмата е последователност от команди, които указват на компютъра какво да прави, за да се постигне даден резултат.

В нашия пример програмата се състои от една команда, която принтира на конзолата текста ***Hello world !!!***

Първият ред от програмата представлява коментар.

Програмите на Python са файлове с разширение .py

В езика Python не се слага точка и запетая (;) в края на командите. Командите се разделят една от друга чрез нов ред или индентация (отстъп).

### Понятие за променлива

Променливата в програмните езици има за цел да съхранява данни. Променливите са именувани области от паметта, които пазят данни от определен тип, например число или текст. Всяка една променлива в Python има име и стойност.

### Пример2: дефиниране на променлива и присвояване на стойност:

```
a = 10  
b = 2  
c = a * b  
print(a + b)  
print(c)
```

## Типове данни

В Python съществуват типове данни. При присвояване на стойност на променливата типът данни се избира автоматично, съобразно присвоената стойност .

Вградени типове : булев, низ, цяло число , число с плаваща запетая

Колекции : списък, кортеж(неизменяем списък), речник

**bool**- логически тип данни. Може да съдържа само 2 стойности: true или false, които съответстват на числата 1 и 0.

**complex**- комплексни числа

**dict**- речник

**float**- реални числа

**int**- цели числа. Размерът на числото може да бъде толкова голям , колкото позволява размерът на достъпната оперативна памет

**list**- списък. Аналогичен е на масива в другите езици за програмиране

**set**- множество (набор от уникални обекти)

**str**- низ

**tuple**- кортеж (наредени n-торки)

Всички типове данни в Python могат да се разделят на : изменяеми и неизменяеми. Към неизменяемите типове данни спадат числата, низовете, кортежите. Към изменяемите се отнасят списъците и речниците.

## Оператори

Операторите извършват определени действия с данни. Например математическите оператори изпълняват аритметични изчисления.

### Оператор за присвояване =

За присвояване на стойност се използва оператора =

На променливите може да бъде присвоена :

- Обикновена стойност (константата)

```
a = 1      # на променливата a се присвоява стойност 1
name = ' Mladen'      # на променливата се присвоява
константата Mladen
```

-стойност на друга променлива

```
b = a
```

- Резултат от изчисление на израз

```
c = a + b * 5
```

- Резултат от действието на функция

```
sum=func(a)
```

## Математически оператори

-събиране +

-изваждане -

-умножение \*

-обикновено деление /

-деление с остатък //

-остатък от деление %

## Оператори за работа с последователности

**+** конкатенация- обединява две последователности

Пример: 'Hello ' + 'World ' -> Hello World **# низове**

[1,2,3] + [4,5,6] -> [1,2,3,4,5,6] **# списъци**

**\*** повторение -създава нова последователност

Пример: 'b '\*4 -> ' bbbb '

[5] \* 3 -> [5,5,5]

**In** проверка за наличие

Пример: 'e' in 'hello' -> True

's' in 'hello' -> False

## Побитови оператори- използват се за манипулиране на отделните битове

~ побитова инверсия(стойността на бита се променя-> 1 става 0, 0 става 1)

**&** побитово И

**|** побитово ИЛИ

**^** побитово изключващо ИЛИ

**<<** изместване вляво

**>>** изместване вдясно

## Печатане на резултат на екрана

Функция **print (...)**- с нея можем да принтираме - стойността на променлива, текст или число

```
print(33)      # печатане на число
print('Hello World!')  # печатане на текст
word = 'Hello World!'
print(word)    # печатане на стойност на променлива
```

## Четене на потребителски вход

Използваме вградената **функция input(...)** за четене на текстов ред от конзолата и **функция int(...)** за преобразуване на текстова стойност към цяло число или **функция float(...)** за преобразуване на текстова стойност към дробно число.

Ако не преобразуваме входа към число, за програмата всяко едно число ще бъде просто текст, с който не можем да извършваме аритметични операции. При извикването на `input (...)` можем да подадем подканващо съобщение за потребителя, с което му казваме какво трябва да въведе,

Например: `s = int(input('Size : '))`

## Отпечатване на форматиран текст в Python

В езика Python има няколко начина да отпечатаме форматиран текст, т.е. текст, смесен с числа, стойности на променливи и изрази.

### 1. Конкатенация на текст с оператора +

Вече знаем как да съединяваме текст и числа с оператора +.

```
a = 3
b = 9
```

```
print('a = ' + str(a) + ' b = ' + str(b) + ' S = ' + str(a * b))
```

Резултатът е :

a=3 b=9 S=27

### Форматиращи низове %d, %s, %f

```
a= 3  
b= 9  
s= "area"  
print('a = %d ; b = %d ; %s = %d' % (a, b, s, a * b))
```

Използваме оператор %, който замества в текста стойности, подадени като поредица от елементи в скоби. Използват се следните основни форматни спецификатори:

**%d** обозначава цяло число

**%f** обозначава дробно число

**%s** обозначава текст

Когато форматираме дробни числа можем да закръгляме до определен брой цифри след десетичната запетая, например с %.3f отпечатваме дробно число с 3 знака след десетичната запетая.

### Форматиране с .format(...)

Можем да форматираме текст и числа чрез метод **.format(...)** като използваме номерирани шаблони {0}, {1}, {2} и т.н. Ето пример демонстриращ този метод:

```
a= 3  
b= 9  
print('a = {} b = {} S = {}'.format(a, b, a * b))
```

### Форматиране с f-string

Поставяме префикс **f** пред стринга и в него поставяме на желани позиции стойности на променливи и изрази във фигурни скоби { }.

Пример :

```
a= 3
b= 9
print(f'a = {a} ; b = {b} ; area = {a * b}')
```

### Задачи :

1. Напишете програма, която чете от конзолата три числа a, b и h и пресмята лицето на трапец с основи a и b и височина h.  
$$(a + b) * h / 2$$

Принтирайте резултата на екрана, като го закръглите до втория знак след десетичната запетая.
2. Напишете програма, която чете от конзолата число r и пресмята и отпечатва лицето и периметъра на окръжност с радиус r. Принтирайте резултата, като го закръглите до 3 знака след десетичната запетая.
3. Напишете програма, която да подкани потребителя да въведе броя на часовете и тарифа за час. Да се изчисли и принтира брутното заплащане.

Примерен вход:

Enter Hours: 35

Enter Rate: 2.75

Изход:

Pay: 96.25