

Для создания комментария используется знак '#'

Пример комментария:

```
#какой-то комментарий
```

Для вывода данных используется функция 'print()'

Пример вывода строки и чисел:

```
print("Результат: ", 228, 55, ".")
```

Для экранирования спецсимволов используется '\'

Пример экранирования двойных кавычек:

```
print("Когда вы говорите: \"Я написал программу, от которой упала Windows\", ")
```

\n – используется для перехода на новую строку

\t – используется для вывода табуляции

## Математические операции в Python

```
#Сложение:
print(10 + 4)
#Вычитание:
print(10 - 4)
#Умножение:
print(10 * 4)
#Деление:
print(10 / 4)
#Целочисленное деление:|
print(10 // 4)
#Возведение в степень:
print(10 ** 4)
#Получение остатка от деления:
print(10 % 4)
```

Найти минимальное число:

```
print("Наименьшее число: ", min(2, 10, 552, 67, -13, 0))
```

Найти максимальное число:

```
print("Максимальное число: ", max(2, 10, 552, 67, -13, 0))
```

Число по модулю:

```
print("Число по модулю: ", abs(-1))
```

## Побитовые операторы

Бинарный оператор & (И) копирует бит в результат только если бит присутствует в обоих операндах

Результат операции a & b даст нам число 12 которое соответствует числу 00001100 в двоичном формате:

a     00111100

b     00001101

-----

&     00001100

Пример использования:

```
a = 60
b = 13
print( a & b)
```

Бинарный оператор |

Бинарный оператор | (ИЛИ) копирует бит если тот присутствует хотя в одном операнде.

Результат операции  $a|b$  даст нам число 61 которое соответствует числу 00111101 в двоичном формате:

a     00111100

b     00001101

-----

|     00111101

Бинарный оператор ^

Бинарный оператор ^ (XOR) копирует бит только если бит присутствует в одном из операндов, но не в обоих сразу.

Результат операции  $a^b$  даст нам число 49 которое соответствует числу 00110001 в двоичном формате

a     00111100

b     00001101

-----

^     00110001

Бинарный оператор ~

Бинарный оператор ~ является унарным (то есть нужен только один операнд) меняет биты на обратные там, где была единица становится ноль и наоборот

Результат операции  $\sim a$  даст нам число -61

a     000000111100

-----

~     111111000011

Бинарный оператор <<

Бинарный оператор << (сдвиг влево) «сдвигает» значение левого операнда влево на количество бит указанных в правом операнде.

Результат операции  $a \ll$  даст нам 240 которое соответствует числу 11110000

```
a      00111100
-----
<<    11110000
```

Бинарный оператор >>

Бинарный оператор >> (сдвиг вправо) «сдвигает» значение левого операнда вправо на количество бит указанных в правом операнде.

Результат операции  $a \gg 2$  даст нам число 15 которое соответствует числу 00001111 в двоичном формате:

```
a      00111100
-----
>>2   00001111
```

## Переменные и типы данных

Пример инициализации переменной:

```
test_var = 54
```

Вывод значения переменной:

```
print(test_var)
```

Булевая переменная:

```
bool_var = True
```

В python не существует строгой типизации

Приведение типов:

```
a = 22.4  
print(int(a))
```

## Ввод и вывод данных

Для получения пользовательского ввода используется функция `input()`

Пример ввода значения переменной:

```
name = input("Введите имя: ")
```

Пример вывода значения переменной

```
print("Добро пожаловать", name, "!")
```

## Системы счисления в python:

```
a = 10
print(a) #десятичная
print(bin(a)) #двоичная
print(oct(a)) #восьмиричная
print(hex(a)) #шестнадцатиричная
```

## Округление чисел:

Пример округления с помощью функции `format()`

```
print("{:.3f}".format(a)) #округлит до трёх знаков после запятой
```

Округление с помощью функции `round()`

```
print(round(a, 2)) #округлит до двух знаков после запятой
```

## Модуль **math**

Модуль **math** – один из наиважнейших в Python. Этот модуль предоставляет обширный функционал для работы с числами.

**math.pow(X, Y)** -  $X^Y$ .

**math.sqrt(X)** - квадратный корень из X.

**math.cos(X)** - косинус X

**math.sin(X)** - синус X

**math.tan(X)** - тангенс X

**math.pi** -  $\pi = 3,1415926...$

**math.e** -  $e = 2,718281...$

## Условный оператор if:

```
if a > b:
    print("max a")
elif a == b:
    print("равны")
else:
    print("max b")
```

## Цикл for

Пример 1:

```
for i in range(6):#сработает 6 раз
    print(i)
```

Пример 2:

```
for i in range(2,6):#сработает 4 раза
    print(i)#выведет 2, 3, 4, 5
```

Пример 3:

```
for i in range(2,6, 2):#сработает 2 раза, будет добавлять 2, вместо 1|
    print(i)#выведет 2,4
```

Пример 4:

посимвольно выведет строку

```
word = "Hello World"
for i in word:
    print(i)
```

## Цикл while

Пример:

```
i = 0
while i < 15:
    print(i)
    i += 3
```



## Случайные числа

необходимо подключить модуль random (прописать import random)

Пример генерации случайного вещественного числа в диапазоне от 10 до 4:

```
import random  
  
print(random.random() * (10 - 4) + 4)
```

Пример генерации случайного целого числа от 0 до 10:

```
print( random.randint(0, 10))
```