



# УСЛОВНЫЕ ОПЕРАТОРЫ

# Проверка пройденного на занятии №1

1. Кто такой программист и на какие три группы их можно поделить?
2. Что вы знаете о языке программирования Python?
3. Дайте определение терминам программа и команда.
4. Чем отличается низкоуровневый язык программирования от высокоуровневого?
5. Из чего состоит программа?
6. Из чего могут состоять имена программы, констант, переменных?
7. Что запрещено использовать в именах программы, констант, переменных?
8. Какие 4 основные типа данных?
9. Какой символ в Python является признаком комментария? С помощью какого сочетания клавиш можно закомментировать много строк кода?
10. Какие встроенные функции вы изучили на прошлом занятии и за что они отвечают?
11. Как можно изменить переменную в Python?
12. Какие арифметические операции вы знаете?
13. Как в Python используется приведение типов?

# Проверка домашнего задания

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

`math.acos(X)` - арккосинус  $X$ . В радианах.

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

`math.asin(X)` - арксинус  $X$ . В радианах.

`math.atan(X)` - арктангенс  $X$ . В радианах.

`math.cos(X)` - косинус  $X$  ( $X$  указывается в радианах).

`math.sin(X)` - синус  $X$  ( $X$  указывается в радианах).

`math.tan(X)` - тангенс  $X$  ( $X$  указывается в радианах).

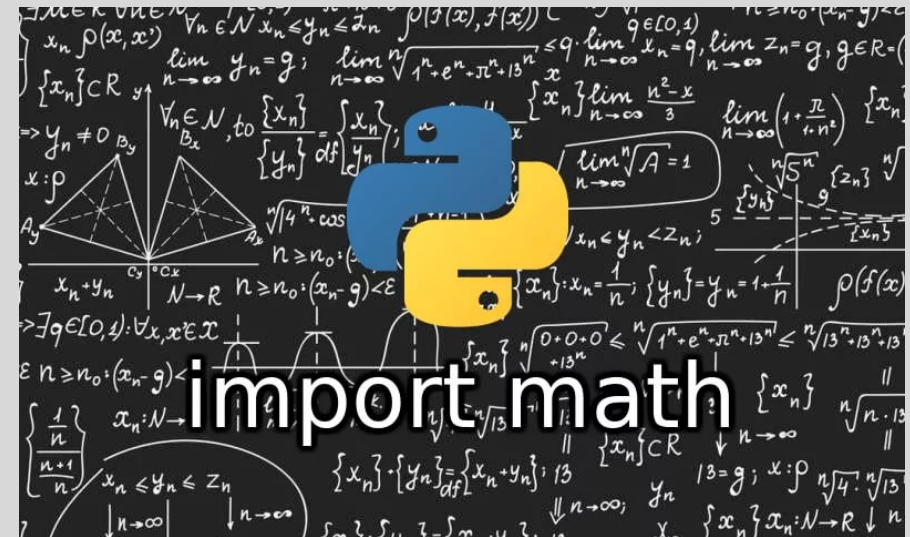
`math.factorial(X)` - факториал числа  $X$ .

`math.floor(X)` - округление вниз.

`math.ceil(X)` – округление до ближайшего большего числа.

`math.sqrt(X)` - квадратный корень из  $X$ .

`math.fabs(X)` - модуль  $X$ .



# Модуль random

Модуль random предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

**random.randint(A, B)** - случайное целое число  $N$ ,  $A \leq N \leq B$ .

**random.uniform(A, B)** - случайное число с плавающей точкой,  $A \leq N \leq B$ .

**random.random()** - случайное число от 0 до 1.

**random.choice(sequence)** - случайный элемент непустой последовательности.

# Вычислить сумму цифр случайного трёхзначного числа

```
import random

# При умножении на 900 получается случайное число от 0 до 899.(9).
# Если прибавить 100, то получится число от 100 до 999.(9).

n = random.randint(100, 999)
print(n)

# Извлекается первая цифра (старший разряд) числа путем
# деления нацело на 100

a = n // 100

# Деление нацело на 10 удаляет последнюю цифру числа.
# Затем нахождение остатка при делении на 10 извлекает
# последнюю цифру, которая в исходном числе была средней.

b = (n // 10) % 10

# Последняя цифра (младший разряд) числа находится
# путем нахождения остатка при делении нацело на 10.

c = n % 10

# Вычисляется сумма цифр и выводится на экран

print(a + b + c)
```

687

21

*Process finished with exit code 0*

# План занятия

- 1)Операторы сравнения
- 2)Операторы ветвления (условия)
- 3)Тип данных bool
- 4)Операторы or, and и not

# Операторы сравнения

Оператор	Описание	Примеры
<code>==</code>	Проверяет равны ли оба операнда. Если да, то условие становится истинным.	5 == 5 в результате будет True True == False в результате будет False "hello" == "hello" в результате будет True
<code>!=</code>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 != 5 в результате будет True False != False в результате будет False "hi" != "Hi" в результате будет True
<code>&lt;&gt;</code>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 <> 5 в результате будет True. Похоже на оператор !=
<code>&gt;</code>	Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	5 > 2 в результате будет True. True > False в результате будет True. "A" > "B" в результате будет False.
<code>&lt;</code>	Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	3 < 5 в результате будет True. True < False в результате будет False. "A" < "B" в результате будет True.
<code>&gt;=</code>	Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	1 >= 1 в результате будет True. 23 >= 3.2 в результате будет True. "C" >= "D" в результате будет False.
<code>&lt;=</code>	Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	4 <= 5 в результате будет True. 0 <= 0.0 в результате будет True. -0.001 <= -36 в результате будет False.

# Операторы ветвления (условия)

## *Условный оператор ветвления if*

Оператор ветвления if позволяет выполнить определенный набор команд в зависимости от некоторого условия

Синтаксис оператора if выглядит так.

**if** выражение:

**команда\_1**

**команда\_2**

**...**

**команда\_n**

После оператора if записывается выражение. Если это выражение истинно, то выполняются команды, определяемые данным оператором. Выражение является истинным, если его результатом является число не равное нулю, непустой объект, либо логическое True. После выражения нужно поставить двоеточие :

**ВАЖНО:** блок кода, который необходимо выполнить, в случае истинности выражения, отделяется четырьмя пробелами слева!



# Пример программы

```
a = 3  
if a > 1:  
    print("a > 1 ")  
  
if True:  
    print("Hello!")
```

## ***Конструкция if – else***

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т.е. при истинном условии нужно выполнить один набор команд, при ложном – другой. Для этого используется конструкция if – else.

**if выражение:**

**команда\_1**

**команда\_2**

**...**

**команда\_n**

**else:**

**команда\_a**

**команда\_b**


**...**





**команда\_x**

# Пример программы

```
1 a = 2
2 if a < 1:
3     print("a < 1 ")
4 else:
5     print("a > 1")
```

else

Run:  2 ×

  C:\Users\User\Desktop  
  a > 1

## Задание №1

Необходимо написать программу, которая требует у пользователя ввести целое число и проверяет чётное оно или нет

# Решение:

```
# Необходимо написать программу, которая требует у пользователя  
# ввести целое число и проверяет чётное оно или нет  
  
# пользователь вводит число  
a = input()  
  
# преобразовываем его к целочисленному типу  
a = int(a)  
  
# оператор % возвращает остаток от деления,  
# чётные числа нацело делятся 2,  
# т.е остаток от деления равен 0.  
# Нечётные числа имеют остаток, равный 1.  
  
if a % 2 == 0:  
    print("Чётное")  
else:  
    print("Нечётное")
```

## ***Конструкция if – elif – else***

Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif – else.

Проще говоря, она выбирает, какое действие следует выполнить, в зависимости от значения переменных в момент проверки условия.

**if выражение\_1:**

**команды\_(блок\_1)**

**elif выражение\_2:**

**команды\_(блок\_2)**

**elif выражение\_3:**

**команды\_(блок\_3)**

**else:**

**команды\_(блок\_4)**

# Пример программы

```
a = int(input('Введите число:'))  
if a < 0:  
    print('Число меньше 0')  
elif a == 0:  
    print('Число равно 0')  
else:  
    print('Число больше 0')
```

# Операторы or, and и not

Оператор	Описание	Примеры
and	Логический оператор "И". Условие будет истинным если оба операнда истина.	True and True равно True. True and False равно False. False and True равно False. False and False равно False.
or	Логический оператор "ИЛИ". Если хотя бы один из операндов истинный, то и все выражение будет истинным.	True or True равно True. True or False равно True. False or True равно True. False or False равно False.
not	Логический оператор "НЕ". Изменяет логическое значение операнда на противоположное.	not True равно False. not False равно True.



# Пример программы

```
1 a = int(input())
2 b = int(input())
3 c = bool(input())
4
5 if a > 0 and b < 0:
6     print("and")
7 elif a > 0 or b < 0:
8     print("or")
9
10 if not c == False:
11     print("not")
12
13 ~~~~~
```

Run: 6 ×  
C:\Users\User\Desktop\overone\  
3  
-2  
True  
and  
not

## Задание №2

Вводятся три целых числа. Определить какое из них наибольшее.

# Решение

```
# Вводятся три целых числа. Определить какое из них наибольшее.
```

```
a = int(input())
```

```
b = int(input())
```

```
c = int(input())
```

```
if a > b and a > c:
```

```
    print('Наибольшее: ', a)
```

```
else:
```

```
    if b > c:
```

```
        print('Наибольшее:', b)
```

```
    else:
```

```
        print('Наибольшее:', c)
```

### Задание №3

Написать примитивный калькулятор. Пользователь должен ввести число, потом операцию (+-/\*) и потом ещё одно число, после этого пользователь получает ответ. Числа могут быть дробными

# Решение

```
4 a = int(input("Введите первое число: "))
5 b = int(input("Введите второе число: "))
6 s = input("Введите операцию которую вы хотите сделать: ")
7
8 if s == "+":
9     print(a+b)
10 elif s == "-":
11     print(a-b)
12 elif s == "*":
13     print(a*b)
14 elif s == "/":
15     print(a/b)
16 elif s == "%":
17     print(a%b)
18 elif s == "^":
19     print(a**b)
```

elif s == "^"

Run: 5 ×

C:\Users\User\Desktop\overone\venv\Scripts\python.exe "C:/Users  
Введите первое число: 6  
Введите второе число: 2  
Введите операцию которую вы хотите сделать: ^  
36

#### Задание №4

Дана следующая функция  $y = f(x)$ .

$y = 2x - 10$ , если  $x > 0$

$y = 0$ , если  $x = 0$

$y = 2 * |x| - 1$ , если  $x < 0$

*Примечание: для нахождения модуля используйте встроенную функцию  $abs(x)$*

# Решение:

```
1  #Дана следующая функция  $y = f(x)$ :  
2  #  $y = 2x - 10$ , если  $x > 0$   
3  #  $y = 0$ , если  $x = 0$   
4  #  $y = 2 * |x| - 1$ , если  $x < 0$   
5  
6  x = int(input())  
7  
8  if x > 0:  
9      y = 2 * x - 10  
10 elif x == 0:  
11     y = 0  
12 else:  
13     y = 2 * abs(x) - 1  
14  
15 print(y)
```

# Тип данных bool

Логический тип данных (bool) (или булевый тип) это примитивный тип данных, который принимает 2 значения – истина или ложь.

В Python имеется самостоятельный логический тип bool, с двумя предопределенными значениями:

True - истина

False - ложь

**Важно!** True и False пишутся с большой буквы. Если написать с маленькой true, интерпретатор выдаст ошибку: `NameError: name 'true' is not defined`



### Задание №5

Напишите программу, которая выполняет сравнение двух переменных , если условие верно то выполняется вложенный блок части if.

Обязательное условие: if a > b писать нельзя. Только использование булевого значения выражения.

# Решение:

```
1  # Напишите программу, которая выполняет сравнение
2  # двух переменных , если условие верно то выполняется
3  # вложенный блок части if.
4  |
5  x = int(input())
6  y = int(input())
7
8  two_values = (x < y)
9
10 if two_values:
11     print(x, "меньше", y)
```

## Домашнее задание

Прочитать про форматированный ввод и вывод переменных.

**Задача №1:** Определить, является ли год високосным

**Задача №2:** Определить существование треугольника и его тип

**Задача №3:** Определить принадлежность точки кругу с центром в начале координат.

Вводятся координаты  $(x, y)$  точки и радиус круга  $r$ .

Определить принадлежит ли данная точка кругу, если его центр находится в начале координат.

**Задача №4:** Творческое задание

Придумать свою задачу на тему занятия. Обязательно использовать несколько вложений `if-else(elif)`

