

# Тема № 2.

## Основы программирования

### Лекция 2.2

#### Условный оператор **if**.

#### Логические операторы.

Работа интерпретатора.

Операторы сравнения.

Логические операторы.

Условные конструкции.

Операторы присваивания

Операторы членства.

# Контрольные вопросы по предыдущей теме

Что такое программирование?

Какие бывают языки программирования?

Где применяется Питон?

Функции ввода-вывода?

Что такое переменная?

Какие типы данных Вам известны?

Функции преобразования типов

Что такое комментарий?

# Как работает интерпретатор?

После запуска программы, интерпретатор проверяет код на наличие синтаксических ошибок, если есть ошибки, то программа не запустится.

```
1 print("hello, World"
```

```
    print("hello, World"  
          ^
```

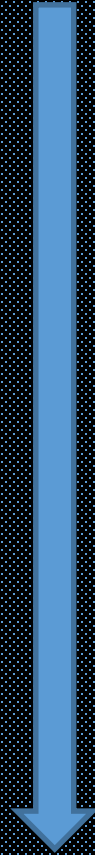
```
SyntaxError: '(' was never closed
```

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

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

```
s = a * b
```

```
print("площадь = ", s)
```



Если синтаксических ошибок нет, интерпретатор читает код и выполняет команды построчно, по очереди сверху вниз.

# Операторы сравнения

оператор	значение	выражение
>	больше	<code>a &gt; b</code>
<	меньше	<code>a &lt; b</code>
==	равно (не путать с =)	<code>a == b</code>
>=	больше или равно	<code>a &gt;= b</code>
<=	меньше или равно	<code>a &lt;= b</code>
!=	не равно	<code>a != b</code>

В результате операций сравнения возвращается булево значение (True / False).

Сравнения могут быть записаны в цепочку

# Операторы сравнения

```
x = 10
```

```
y = 20
```

```
print(y > 0)
```

```
# True
```

```
print(y != x)
```

```
# True
```

```
print(x >= y)
```

```
# False
```

```
x = "a"
```

```
y = "b"
```

```
print(y > x)
```

```
# True
```

```
x = "abz"
```

```
y = "abb"
```

```
print(y > x)
```

```
# False
```

# Логические операторы.

## 1. and

Логическое И – возвращает True, только когда оба операнда True

## 2. or

Логическое ИЛИ – возвращает True, когда хотя бы один операнд True

## 3. not

Логическое НЕ – возвращает булево значение, противоположное операнду

```
x = False
```

```
y = True
```

```
print(x and y)
```

```
# False
```

```
x = False
```

```
y = True
```

```
print(x or y)
```

```
# True
```

```
x = 0
```

```
print(not x)
```

```
# True
```

```
x = 10
```

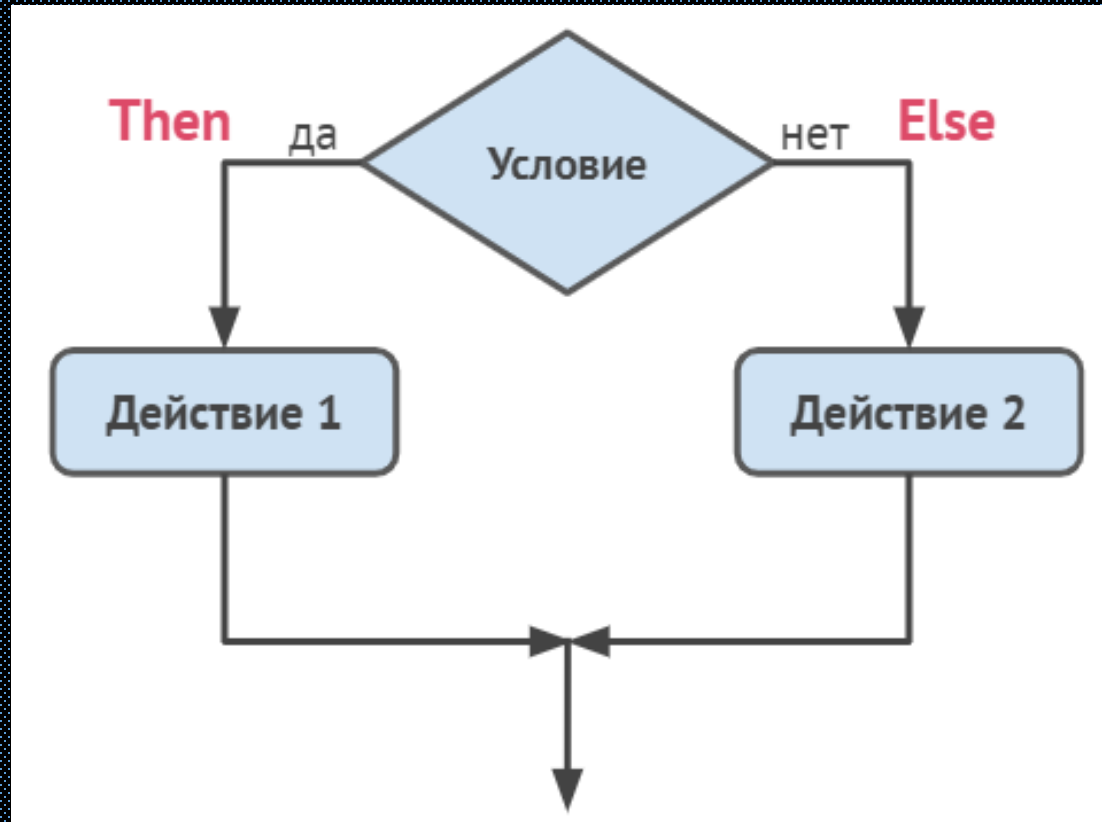
```
y = 20
```

```
print(y > 0 and x > 0)
```

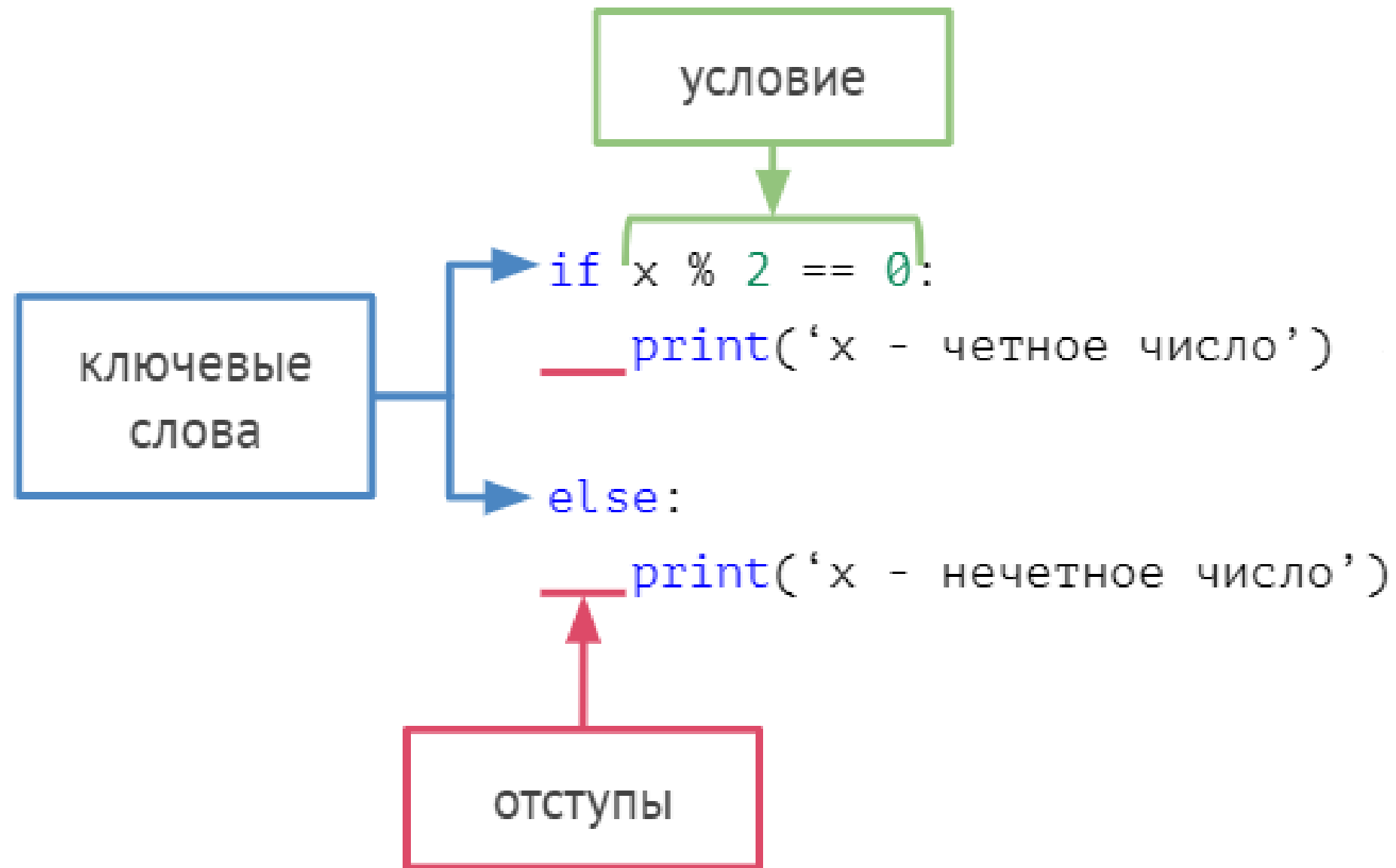
```
# True
```



# Условные оператор if



это логическое выражение, после которого пишутся команды, которые выполняются, если условие истинно.



# Каскадные условные конструкции

```
x = int(input('Введите координату X:'))
```

```
y = int(input('Введите координату Y:'))
```

```
if x > 0 and y > 0:
```

```
    print('Первая четверть')
```

```
elif x > 0 and y < 0:
```

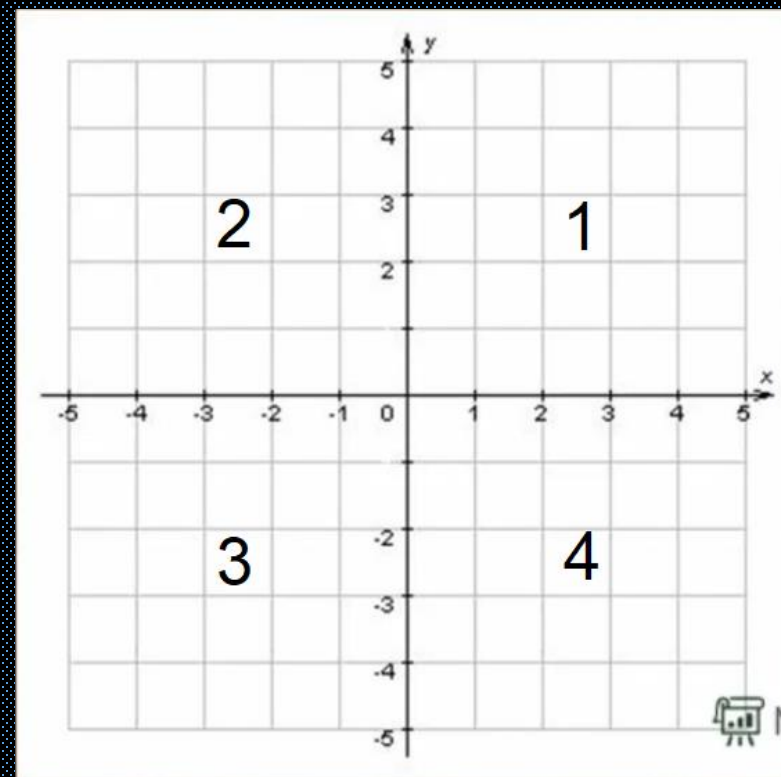
```
    print('Четвертая четверть')
```

```
elif y > 0:
```

```
    print('Вторая четверть')
```

```
else:
```

```
    print('Третья четверть')
```



# Вложенные условные конструкции

```
x = int(input('Введите координату X:'))
```

```
y = int(input('Введите координату Y:'))
```

```
if x > 0:
```

```
    if y > 0: # x > 0, y > 0
```

```
        print('Первая четверть')
```

```
    else: # x > 0, y < 0
```

```
        print('Четвертая четверть')
```

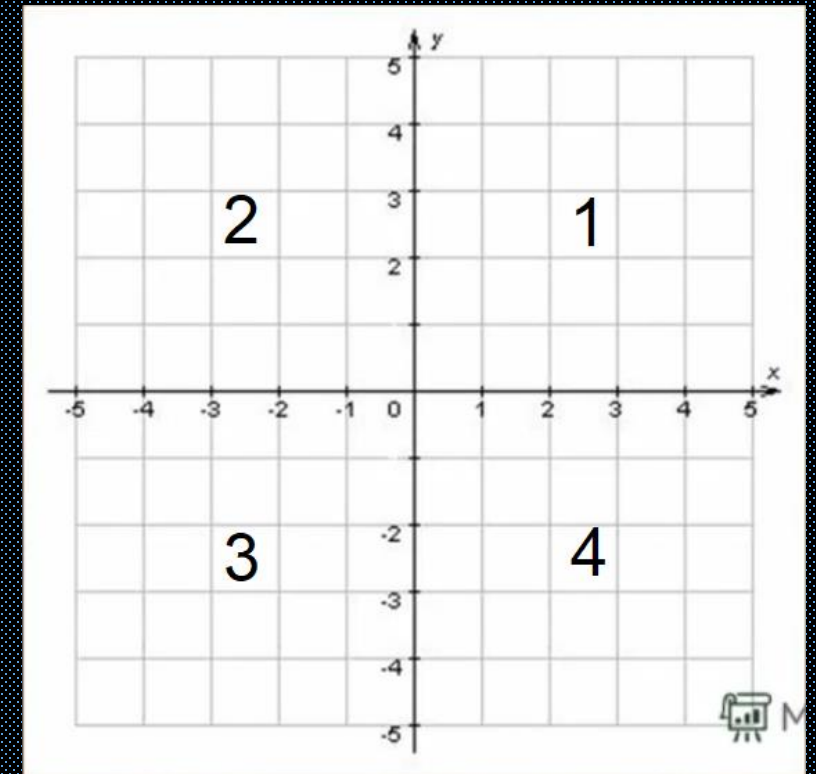
```
else:
```

```
    if y > 0: # x < 0, y > 0
```

```
        print('Вторая четверть')
```

```
    else: # x < 0, y < 0
```

```
        print('Третья четверть')
```



## Оператор членства

Оператор членства возвращает логическое значение `True` когда элемент найден в данной коллекции элементов, и `False`, если не найден.

Этот оператор могут использоваться с любой итерируемой структурой данных (последовательностью) в Python.

`in` возвращает `True`, если данный элемент присутствует в структуре данных.

`not in` возвращает `True`, если данный элемент отсутствует в структуре данных.

```
print("l" in "Hello, World!" )
```

```
# True
```

```
if "l" in "Hello, World!":
```

```
    print("simbol found!")
```

```
# simbol found!
```

## Оператор тождественности

`is` - возвращает `True`, если переменные являются одним объектом.

`not is` - возвращает `True`, если переменные разные.

```
x = "Hello"
```

```
y = x
```

```
if x is y:
```

```
    print("yes ", id(x), id(y))
```

```
# yes 2993917815664 2993917815664
```

# Оператор тождественности

```
x = "Hello"
```

```
y = x
```

```
print(x is y)
```

```
# True
```



# Приложение «Военкомат»

Приложение принимает возраст призывника и рекомендует род войск:

- до 170 – танковые войска;
- 171 – 180 – мотострелковые войска;
- 181 – 190 – десантные войска;
- выше 190 – другие.

# Литература:

Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2022. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493261> (дата обращения: 26.08.2022).

<https://github.com/ViktorViktorovitsh/lessons>

# Задание на дом:

1. Повторить изученный материал.
2. На сайте [hackerrank.com](https://hackerrank.com) решить задачу "Python If-Else«
3. Доработать приложение «Военкомат», нужно учесть следующие параметры призывника:
  - возраст;
  - количество детей;
  - является ли он студентом .

Решение присылать сюда: [hhomework@list.ru](mailto:hhomework@list.ru)

**СМОТРИТЕ  
В СЛЕДУЮЩЕЙ СЕРИИ**

**Циклы for, while**