

## Конспект

### Знакомство с условными конструкциями

#### Условные конструкции:

Условные конструкции в Python - это механизмы, которые позволяют программе принимать решения и выполнять определенные действия на основе определенных условий. Они позволяют создавать разветвления в коде, где определенные блоки кода выполняются только в случае, если определенное условие истинно.

Основным элементом условных конструкций в Python является оператор `if`, который проверяет условие и выполняет определенные действия, если условие истинно. Условие может быть выражением, возвращающим булево значение (`True` или `False`). Если условие истинно, то выполняется блок кода, следующий за оператором `if`.

Кроме оператора `if`, в Python также используются ключевые слова `elif` (сокращение от "else if") и `else`. Конструкция `elif` позволяет проверить дополнительные условия, если предыдущие условия не были истинными. Конструкция `else` выполняется, если ни одно из предыдущих условий не было истинным.

#### Основные элементы условных конструкций:

**Условие:** это выражение, которое должно быть оценено как истинное или ложное.

**Блок кода:** это группа инструкций, которые должны быть выполнены, если условие истинно.

**Ключевые слова:** в языке Python используются ключевые слова, такие как `if`, `else` и `elif` (сокращение от "else if"), для создания условных конструкций.

#### 2. Условный оператор `if`:

Условный оператор `if` позволяет выполнить блок кода, только если определенное условие истинно.

Синтаксис оператора if:

```
if условие:  
    блок_кода
```

**Примеры использования оператора if:**

```
x = 5  
  
if x > 0:  
    print("x положительное число")  
  
y = -2  
  
if y < 0:  
    print("y отрицательное число")
```

**Вложенные условия и else-if конструкция:**

Условные конструкции могут быть вложенными, что означает, что одна конструкция может находиться внутри другой. Также можно использовать ключевое слово `elif` для создания цепочки условий, в которой каждое последующее условие проверяется только в случае, если предыдущее условие было ложным.

**Пример использования вложенных условий и else-if конструкции:**

```
x = 10

if x > 0:
    print("x положительное число")
else:
    if x < 0:
        print("x отрицательное число")
    else:
        print("x равно нулю")

y = 5

if y > 0:
    print("y положительное число")
elif y < 0:
    print("y отрицательное число")
else:
    print("y равно нулю")
```

В этих примерах демонстрируется использование вложенных условий и ключевого слова `elif`. В первом примере, если `x` больше 0, выводится сообщение "x положительное число". Если `x` меньше 0, выводится сообщение "x отрицательное число". Во втором примере, если `y` больше 0, выводится сообщение "y положительное число". Если `y` меньше 0, выводится сообщение "y отрицательное число". Если ни одно из условий не выполняется, выводится сообщение "y равно нулю".

### 3. Повторим логические операторы в условных конструкциях:

Логические операторы (`and`, `or`, `not`) могут быть использованы в условных конструкциях для комбинирования условий и создания более сложных проверок.

## Логические операторы and, or, not:

and: возвращает True, если оба условия являются истинными.

```
x = 5
y = 10

if x > 0 and y
```

or: возвращает True, если хотя бы одно из условий является истинным.

not: возвращает логическое отрицание условия.

Примеры использования логических операторов:

### Примеры с использованием условных конструкций с логическими операторами:

**Пример 1:** Определение дня недели по его номеру

```
day = int(input("Введите номер дня недели (1-7): "))

if day == 1:
    print("Понедельник")
elif day == 2:
    print("Вторник")
elif day == 3:
    print("Среда")
elif day == 4:
    print("Четверг")
elif day == 5:
    print("Пятница")
elif day == 6:
    print("Суббота")
elif day == 7:
    print("Воскресенье")
else:
    print("Некорректный номер дня")
```

В этой задаче пользователю предлагается ввести номер дня недели, а затем программа определяет соответствующий день недели. Решение использует операторы сравнения и ключевое слово `elif` для проверки условий и вывода соответствующего сообщения.

**Пример 2:** Определение типа треугольника

```

a = float(input("Введите длину первой стороны: "))
b = float(input("Введите длину второй стороны: "))
c = float(input("Введите длину третьей стороны: "))

if a == b == c:
    print("Треугольник является равносторонним")
elif a == b or b == c or a == c:
    print("Треугольник является равнобедренным")
else:
    print("Треугольник является разносторонним")

```

В этой задаче пользователю предлагается ввести длины сторон треугольника, а затем программа определяет его тип: равносторонний (если все стороны равны), равнобедренный (если две стороны равны) или разносторонний (если все стороны различны). Решение использует логические операторы and, or, операторы сравнения и ключевое слово elif для проверки условий и вывода соответствующего сообщения.

### Пример 3: Определение времени суток

```

hour = int(input("Введите текущий час (0-23): "))

if hour >= 0 and hour < 6:
    print("Сейчас ночь")
elif hour >= 6 and hour < 12:
    print("Сейчас утро")
elif hour >= 12 and hour < 18:
    print("Сейчас день")
else:
    print("Сейчас вечер")

```

В этой задаче пользователю предлагается ввести текущий час (в формате 24-часового времени), а затем программа определяет время суток: ночь (с 0:00 до 5:59), утро (с 6:00 до 11:59), день (с 12:00 до 17:59) или вечер (с 18:00 до 23:59). Решение использует логические операторы `and`, операторы сравнения и ключевое слово `elif` для проверки условий и вывода соответствующего сообщения.

#### Пример 4: Определение возрастной группы

```
age = int(input("Введите возраст: "))

if age < 18:
    print("Вы находитесь в возрастной группе 'Несовершеннолетние'")
elif age >= 18 and age < 30:
    print("Вы находитесь в возрастной группе 'Молодежь'")
elif age >= 30 and age < 60:
    print("Вы находитесь в возрастной группе 'Взрослые'")
else:
    print("Вы находитесь в возрастной группе 'Пожилые'")
```

В этой задаче пользователю предлагается ввести свой возраст, а затем программа определяет его возрастную группу: "Несовершеннолетние" (менее 18 лет), "Молодежь" (от 18 до 29 лет), "Взрослые" (от 30 до 59 лет) или "Пожилые" (60 лет и старше). Решение использует логические операторы `and`, операторы сравнения и ключевое слово `elif` для проверки условий и вывода соответствующего сообщения.