Лекция 3

Здравствуйте, уважаемые слушатели!

Тема нашей лекции – Управляющие конструкции: условные операторы и циклы

План лекции:
1. Введение
2. Условные операторы
3. Операторы if, elif и else
4. Цикл for и while
5. Заключение

Введение

Управляющие конструкции Python, включая условные операторы и циклы, обеспечивают эффективное управление потоком выполнения программы, позволяя адаптироваться к различным условиям и обрабатывать данные через повторяющиеся процессы. Условные операторы if, elif, и else играют ключевую роль в реализации ветвлений в программе, позволяя коду реагировать на разнообразные сценарии и принимать решения на основе специфических условий. Эти операторы оценивают условия и направляют выполнение программы в соответствующие блоки кода, что делает программу интеллектуально адаптивной и способной к самостоятельным изменениям поведения в зависимости от внешних или внутренних данных.

Циклы for и while расширяют возможности Python, позволяя многократно исполнять блок кода, что необходимо для обработки коллекций данных, выполнения задач до наступления определенных условий, или просто для повторения операций фиксированное количество раз. Цикл for идеально подходит для итераций по элементам последовательности, таким как списки или строки, в то время как цикл while продолжает выполнение, пока его условие остается истинным, что идеально для выполнения блоков кода до наступления изменений в определенных условиях.

Использование этих управляющих конструкций позволяет создавать не только функциональные, но и интеллектуально гибкие программы, способные адаптироваться к разнообразным условиям и эффективно обрабатывать большие объемы данных. Эти элементы языка являются фундаментом для разработки программ, которые не только выполняют указанные задачи, но и реагируют на изменения контекста, что делает Python мощным инструментом в руках современных программистов.

Условные операторы

Условные операторы в Python являются фундаментальным инструментом для управления логикой выполнения программ. Они позволяют программе динамически реагировать на различные условия и сценарии, выполняя определенные блоки кода в зависимости от того, истинно или ложно данное условие.

Основной условный оператор в Python — это `if`. Он проверяет условие, и если это условие истинно (`True`), то Python выполняет блок кода, который следует непосредственно за этим условием. Структура оператора `if` проста, что делает его особенно удобным для начинающих программистов.

python
Копировать код
```
if условие:
    блок_кода_1
elif условие_2:
    блок_кода_2
else:
    блок_кода_3
```

Оператор if

Оператор if используется для проверки истинности условия. Если условие истинно, то выполняется блок кода, расположенный непосредственно после условия. Пример использования:

python
Копировать код
```
x = 10
if x > 5:
    print("x больше 5")
```

В этом примере сообщение будет напечатано, так как условие (x > 5) истинно.

Операторы elif и else

Оператор elif (сокращение от "else if") позволяет проверить дополнительные условия, если предыдущие условия оказались ложными. Оператор else выполняется, если ни одно из предшествующих условий не было истинным.

python
Копировать код
```
x = 10
if x > 10:
    print("x больше 10")
elif x == 10:
    print("x равен 10")
else:
    print("x меньше 10")
```

В этом случае будет напечатано "x равен 10", так как первое условие ложно, но второе условие истинно.

Циклы

Циклы в Python являются неотъемлемой частью программирования, позволяя многократно выполнять блок кода в зависимости от заданных условий. Это ключевой элемент для обработки данных, автоматизации задач и создания функций, которые требуют повторения действий до достижения определённого результата. Python предлагает два основных типа циклов: `for` и `while`, каждый из которых имеет свои особенности и области применения. Python предлагает два основных типа циклов: for и while.

Цикл for

Цикл for в Python используется для итерации по элементам последовательности (например, списка, кортежа, словаря, строки) или любого другого итерируемого объекта.

python
Копировать код
```
for i in range(5):
    print(i)
```

В этом примере цикл будет печатать числа от 0 до 4. Функция range(5) создает последовательность этих чисел, которые последовательно присваиваются переменной i, и для каждого значения переменной выполняется блок кода внутри цикла.

Цикл while

Цикл while продолжает выполнение, пока заданное условие истинно.

python
Копировать код
```
x = 5
```

```
while x > 0:
    print(x)
    x -= 1
```

В этом примере цикл будет печатать уменьшающиеся значения x от 5 до 1. Как только x станет равным 0, условие (x > 0) станет ложным, и цикл прекратится.

Заключение

Управляющие конструкции являются фундаментальной частью программирования на Python, позволяющей создавать программы, которые могут принимать решения и выполнять код многократно в зависимости от внутренних и внешних условий. Они обеспечивают не только базовый контроль над потоком выполнения программы, но и служат строительными блоками для более сложных логических структур, которые необходимы в современных программах. Надлежащее понимание и использование условных операторов и циклов является ключевым навыком для каждого разработчика, стремящегося к эффективному и оптимальному программированию.

Литературы:

1. Программирование на Python для начинающих. Райтман М.А. Москва-ЭМОСКО-2015 стр 31-42
2. Python для «чайников». Джон Полль Мюллер. Диалектика-2022 стр. 31-38