**Урок 2: Условные операторы и операторы сравнения**

**Повторение прошлого материала (10 минут):**

На прошлом уроке мы познакомились с языком Python, установили его на наши компьютеры и написали первую программу 'Hello, World!'. Также мы научились использовать команды **print** и **input** выполнять простые математические операции."

**Обсуждение:**

Вопросы к ученикам:

"Кто помнит как работает print?"

"Как работает input?"

**Цели и задачи урока:**

"Сегодня мы будем изучать условные операторы и операторы сравнения в Python. Это важные концепции, которые позволяют нашим программам принимать решения в зависимости от определенных условий."

**Познание нового (25 минут):**

Основные понятия:

1. Условные операторы:

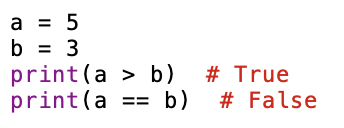
"Условные операторы позволяют программам выполнять разные действия в зависимости от того, истинно или ложно условие. Основной условный оператор в Python - это **if**."



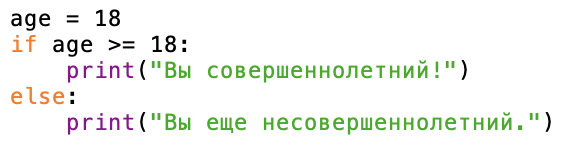
2. Операторы сравнения:

"Операторы сравнения используются для сравнения значений и возвращают **True** или **False**. Основные операторы сравнения в Python:"

* **==** (равно)
* **!=** (не равно)
* **>** (больше)
* **<** (меньше)
* **>=** (больше или равно)
* **<=** (меньше или равно)

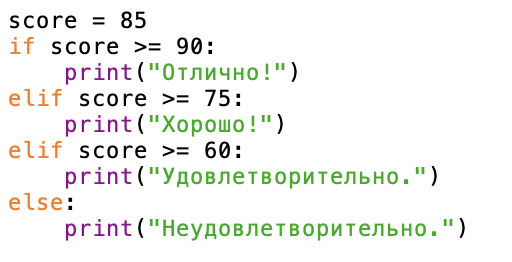


Пример программы, которая использует условный оператор if:



"В этой программе мы проверяем значение переменной age. Если age больше или равно 18, программа выводит 'Вы совершеннолетний!'. В противном случае используя оператор **else** выводится 'Вы еще несовершеннолетний.'"

Как добавить дополнительные условия с помощью elif:



"Здесь мы проверяем оценку score. Если score больше или равно 90, выводится 'Отлично!'. Если score меньше 90, но больше или равно 75, выводится 'Хорошо!'. Если score меньше 75, но больше или равно 60, выводится 'Удовлетворительно.'. В противном случае выводится 'Неудовлетворительно.'."

**Некомпьютерная активность (10 минут)**

Обсуждение:

Примеры использования условных операторов в реальной жизни:

"Условные операторы помогают принимать решения в зависимости от условий. Например, мы можем проверять, есть ли у нас достаточно денег для покупки товара или подходит ли погода для прогулки."

Вопросы для размышления:

"Можете ли вы привести примеры ситуаций, где вы используете условия для принятия решений в повседневной жизни?"

"Какие задачи можно решить с помощью условных операторов в программировании?"

Активность:

Ученики делятся на небольшие группы:

1. Учитель делит класс на группы по 3-4 человека и дает задание:

"Обсудите в группах и придумайте 2-3 примера использования условных операторов в реальной жизни или в программировании. У вас есть 5 минут."

Каждая группа представляет свои идеи классу:

2. Учитель вызывает каждую группу по очереди для представления своих идей. После каждого выступления учитель обсуждает идеи с классом и приводит дополнительные примеры.

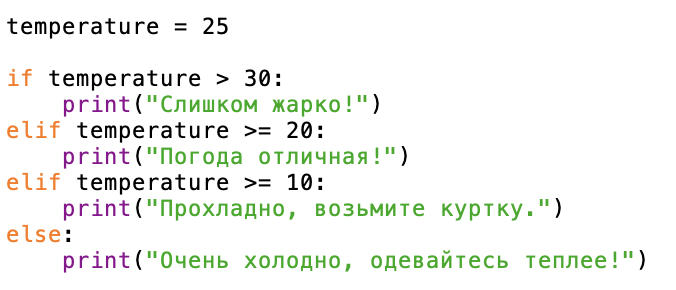
**Работа над проектом (25 минут)**

Практическое задание: Создание программы с использованием условных операторов

Создание нового файла в IDLE:

"Давайте создадим программу, которая будет проверять температуру и давать рекомендации. Откройте IDLE, выберите 'File' -> 'New File'."

Написание кода для проверки температуры:



Запуск программы:

Как сохранить файл с расширением .py и запустить его:

"Сохраните файл как weather.py, затем выберите 'Run' -> 'Run Module' или нажмите F5. Вы должны увидеть текст, соответствующий значению переменной temperature."

Структура программы:

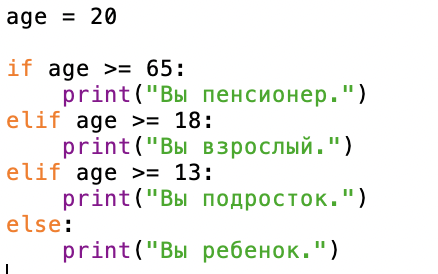
"Мы проверяем значение переменной temperature и выводим соответствующее сообщение. Это пример использования условных операторов для принятия решений в зависимости от условий."

**Дополнительное задание:**

Изменить программу для проверки других условий:

Задание:

"Теперь давайте изменим нашу программу, чтобы она проверяла возраст человека и выводила соответствующее сообщение. Попробуйте написать следующий код:"



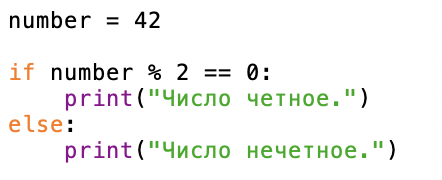
**Проблемная задача (10 минут)**

Задание: Создание программы для определения четности числа

Пример выполнения:

"Теперь давайте создадим программу, которая будет определять, является ли число четным или нечетным."

"Напишите следующий код:"



Этот код проверяет остаток от деления числа на 2. Если остаток равен 0, число четное, иначе нечетное.

"Запустите программу и посмотрите, что получится. Как Python определяет четность или нечетность числа?"

**Рефлексия (10 минут):**

"Что нового вы узнали сегодня?"

"Что вам показалось наиболее интересным или сложным?"