**Урок 10: Работа с регулярными выражениями**

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

“Прежде чем мы начнем, давайте вспомним, что мы изучали на прошлом уроке. Кто может объяснить, что такое множество и словарь?”

“Правильно, множество — это коллекция уникальных элементов, а словарь — это коллекция пар ключ-значение.”

“Кто может рассказать, как мы использовали множества и словари в наших программах?”

“Верно, мы использовали множества для хранения уникальных значений и словари для организации данных в виде пар ключ-значение.”

**Цели и задачи урока:**

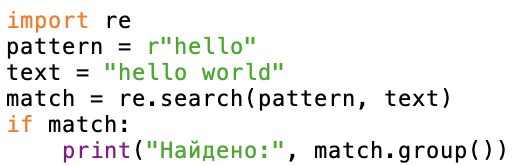
“Сегодня мы будем изучать регулярные выражения в Python. Регулярные выражения — это мощный инструмент для поиска и обработки текста.”

**Познание нового (25 минут):**

Основные понятия:

1. “Регулярные выражения (regex) — это шаблоны, которые используются для поиска и обработки текста. С их помощью можно находить и заменять подстроки в строках.”

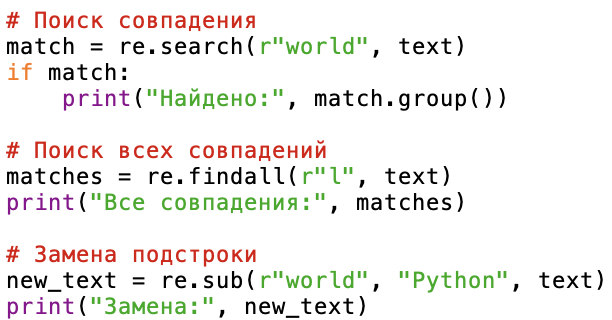
Пример простого регулярного выражения:



2. Основные функции модуля re:

“В Python для работы с регулярными выражениями используется модуль re. Основные функции этого модуля: search, match, findall, sub.”

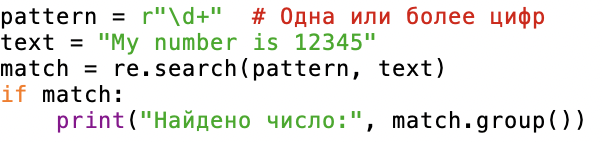
Примеры использования функций:



3. Метасимволы и специальные последовательности:

“Метасимволы и специальные последовательности помогают создавать более сложные шаблоны. Например, \d соответствует любой цифре, \w — любому словесному символу, \s — пробелу.”

Примеры метасимволов:



**Некомпьютерная активность (10 минут):**

Обсуждение:

Примеры использования регулярных выражений в реальной жизни:

“Регулярные выражения можно сравнить с инструментом для поиска и замены текста в документах. Например, вы можете искать все номера телефонов в тексте и форматировать их.”

Вопросы для размышления:

“Почему важно уметь работать с регулярными выражениями? Какие задачи можно решать с их помощью?”

Активность:

Ученики делятся на небольшие группы:

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

1. “Обсудите в группах, какие данные вы могли бы искать и заменять с помощью регулярных выражений. У вас есть 5 минут.”

Каждая группа представляет свои идеи классу:

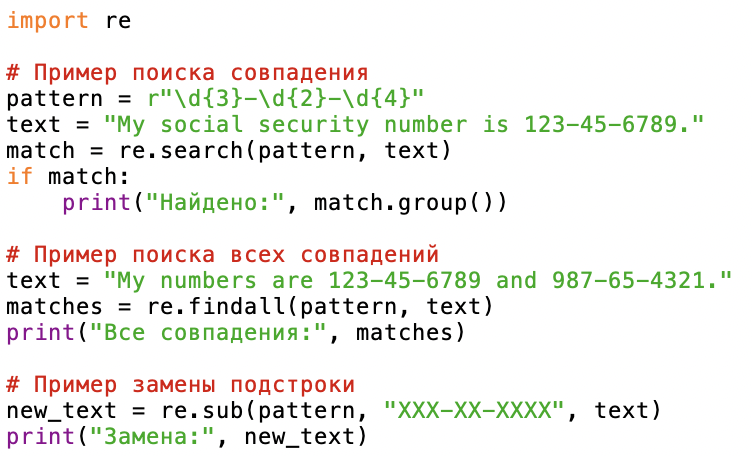
2. Учитель вызывает каждую группу по очереди для представления своих идей. После каждого выступления учитель обсуждает идеи с классом и приводит дополнительные примеры.

**Работа над проектом (25 минут):**

Практическое задание: Создание программ с использованием регулярных выражений

“Давайте создадим программу, которая использует регулярные выражения. Откройте IDLE, выберите ‘File’ -> ‘New File’.”

Написание кода для выполнения задач:



“Сохраните файл как regex\_example.py, затем выберите ‘Run’ -> ‘Run Module’ или нажмите F5. Вы должны увидеть вывод, соответствующий выполнению программы.”

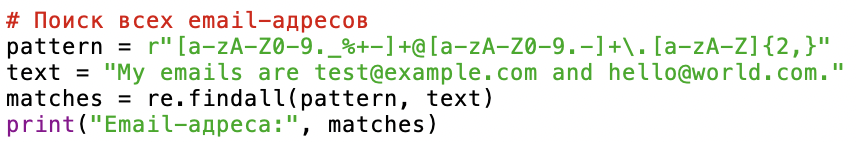
Структура программы:

“Мы создали программу, которая использует регулярные выражения для поиска и замены подстрок. Это показывает, как можно обрабатывать текст с помощью шаблонов.”

**Дополнительное задание:**

Задание: Изменить программу для выполнения других задач

“Теперь давайте изменим нашу программу, чтобы она находила все email-адреса в тексте. Попробуйте написать следующий код:”



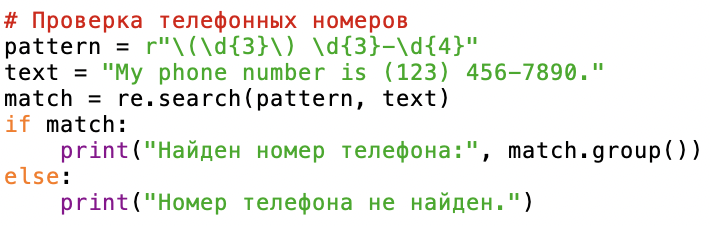
**Проблемная задача (10 минут):**

Задание: Написание программы с использованием регулярных выражений

Пример выполнения:

“Теперь давайте создадим программу, которая будет использовать регулярные выражения для проверки телефонных номеров.”

“Напишите следующий код:”



“Запустите программу и посмотрите, как она работает. Какие операции были выполнены?”

**Рефлексия (10 минут):**

“Сегодня мы узнали о регулярных выражениях в Python. Кто может объяснить, как создать регулярное выражение и какие функции модуля re мы использовали?”