**Лекция 11-12: Списки**

**1. Списки (одномерные массивы)**

Списки (**list** в Python) — это одномерные массивы. Они могут хранить различные типы данных: целые числа, строки, и даже другие списки. К каждому элементу можно обратиться по его индексу. Индексация начинается с 0.

**Пример:**

python

Копировать код

Primes = [2, 3, 5, 7, 11, 13]

print(Primes[0]) # 2

print(Primes[1]) # 3

print(Primes[-1]) # Последний элемент: 13

print(len(Primes)) # Длина списка: 6

К элементам списка можно также обращаться с помощью отрицательных индексов. **Primes[-1]** — это последний элемент, **Primes[-2]** — предпоследний и так далее.

**2. Методы разделения и объединения**

Списки можно разделять и объединять с помощью различных методов. Эти методы в основном используются для работы с элементами списков.

* **split()**: Разделяет строку на список.
* **join()**: Объединяет элементы списка в строку.

**Пример:**

python

Копировать код

text = "Python, Java, C++"

languages = text.split(", ") # Разделение строки на список

print(languages) # ['Python', 'Java', 'C++']

new\_text = ", ".join(languages) # Объединение списка в строку

print(new\_text) # 'Python, Java, C++'

**3. Генераторы списков**

**Генераторы списков** — это способ кратко и эффективно создавать списки в Python. Они позволяют создавать элементы списка с помощью цикла в одной строке.

**Пример:**

python

Копировать код

squares = [x\*\*2 for x in range(10)] # Создаем квадраты чисел от 0 до 9

print(squares) # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

Это удобный способ создания списка с использованием генератора.

**4. Срезы**

**Срезы** позволяют извлекать часть списка. Это метод выбора сразу нескольких элементов списка.

**Пример:**

python

Копировать код

Primes = [2, 3, 5, 7, 11, 13]

subset = Primes[1:4] # Выбираем элементы с индексами от 1 до 4 (не включая)

print(subset) # [3, 5, 7]

Срезы позволяют легко выбрать часть списка и работать с ней.

**Пример с Tkinter: Поле ввода и окно сообщения**

Следующая программа на **Tkinter** позволяет вводить данные в поле и при нажатии кнопки отображать введенное сообщение в диалоговом окне:

python

Копировать код

from tkinter import \*

from tkinter import messagebox

# Функция, вызываемая при нажатии кнопки

def f1():

messagebox.showinfo("Сообщение", message.get())

root = Tk()

root.title("Python")

root.geometry("300x250")

# Настройка поля ввода и кнопки

message = StringVar()

message\_entry = Entry(textvariable=message)

message\_entry.place(relx=.5, rely=.1, anchor="c")

message\_button = Button(text="Нажми меня", command=f1)

message\_button.place(relx=.5, rely=.5, anchor="c")

root.mainloop()

**Как работает программа:**

1. Пользователь вводит текст в поле **Entry**.
2. При нажатии кнопки **"Нажми меня"** введенный текст отображается в диалоговом окне.

Эти методы и примеры можно использовать для выполнения задач, связанных со списками и их элементами.