

## Лекция 11

Здравствуйте, уважаемые слушатели!

Тема нашей лекции – Практика использования стандартной библиотеки

Python План лекции:

1. Работа с текстовыми данными
2. Файловый ввод/вывод
3. Сетевое программирование
4. Многопоточность
5. Заключение

### Введение

Стандартная библиотека Python — это мощный набор модулей и пакетов, которые предоставляются вместе с интерпретатором Python. Эта библиотека содержит инструменты для решения множества разнообразных задач, начиная от работы с текстовыми данными и заканчивая сетевым программированием и многопоточностью. Понимание и умелое использование стандартной библиотеки позволяют значительно ускорить разработку программ и улучшить их надёжность. В этой лекции мы рассмотрим ключевые компоненты стандартной библиотеки Python и примеры их практического применения.

### 1. Работа с текстовыми данными

#### 1.1 Модуль `re` (регулярные выражения)

Модуль `re` предоставляет поддержку регулярных выражений для сложных операций поиска и замены в тексте. Освоение `re` открывает мощные возможности для анализа и обработки строк:

```
python
```

```
Копировать код
```

```
import re
```

```
pattern = re.compile(r'\d+')
matches = pattern.findall('12 drummers, 11 pipers, 10 lords')
print(matches) # ['12', '11', '10']
```

#### 1.2 Модуль `string`

Модуль `string` содержит множество полезных констант и функций для работы со строковыми значениями, включая predefined строки алфавита и цифр, а также функции для форматирования строк:

```
python
```

Копировать код

```
import string
```

```
print(string.ascii_uppercase) # ABCDEFGHIJKLMNOPQRSTUVWXYZ  
template = string.Template("Hello, $name!")  
print(template.substitute(name="Alice")) # Hello, Alice!
```

## 2. Файловый ввод/вывод

### 2.1 Модуль os и os.path

Модули os и os.path предлагают функции для взаимодействия с операционной системой, включая управление файлами и директориями:

```
python
```

Копировать код

```
import os
```

```
current_directory = os.getcwd()  
print(current_directory) # Выводит текущую рабочую директорию
```

```
# Проверка существования файла  
file_path = '/path/to/your/file.txt'  
print(os.path.exists(file_path))
```

### 2.2 Модуль io

Модуль io предоставляет утилиты для работы с потоками ввода-вывода. Это особенно полезно для работы с бинарными данными и для создания пользовательских потоков:

```
python
```

Копировать код

```
import io
```

```
# Чтение данных из строки как из файла  
data = io.StringIO("initial text")  
print(data.read())
```

## 3. Сетевое программирование

### 3.1 Модуль socket

Модуль `socket` используется для создания сетевых соединений. Эти возможности могут быть использованы для реализации клиент-серверных приложений:

```
python
```

Копировать код

```
import socket
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("www.python.org", 80))
```

### 3.2 Модуль `http`

Модуль `http` предоставляет инструменты для создания HTTP-клиентов и серверов. Например, `http.client` может использоваться для отправки запросов:

```
python
```

Копировать код

```
import http.client
```

```
conn = http.client.HTTPSConnection("www.python.org")
conn.request("GET", "/")
resp = conn.getresponse()
print(resp.status, resp.reason)
```

## 4. Многопоточность

### 4.1 Модуль `threading`

Многопоточное программирование в Python может быть реализовано с помощью модуля `threading`, который позволяет запускать несколько потоков выполнения:

```
python
```

Копировать код

```
import threading
```

```
def worker():
    """Функция, выполняемая потоком"""
    print("Worker")
```

```
t = threading.Thread(target=worker)
t.start()
```

## 5. Другие важные модули

## 5.1 Модуль `datetime`

Модуль `datetime` предоставляет классы для работы с датами и временем:

```
python
Копировать код
from datetime import datetime

now = datetime.now()
print(now) # Выводит текущие дату и время
```

## 5.2 Модуль `math`

Модуль `math` содержит функции для математических расчетов, включая тригонометрические функции, экспоненты и логарифмы:

```
python
Копировать код
import math

print(math.sqrt(16)) # 4.0
```

## Заключение

Стандартная библиотека Python обеспечивает разработчиков комплексным набором инструментов, необходимых для выполнения большинства задач программирования. От файлового ввода/вывода до сетевого программирования и многопоточности — знание и умение применять стандартную библиотеку ускоряют разработку и повышают качество программного продукта. Поэтому важно овладеть этими инструментами, чтобы эффективно использовать все возможности Python.

Литературы:

1. Программирование на Python для начинающих. Райтман М.А. Москва-ЭМОСКО-2015 стр 74-79
2. Python для «чайников». Джон Полль Мюллер. Диалектика-2022 стр. 78-83