УРОК 35. ОСНОВЫ РАБОТЫ С СЕТЬЮ

HTTP, URL

Протокол HTTP (Hypertext Transfer Protocol) является основным протоколом для обмена данными в Интернете.

Он определяет структуру и правила взаимодействия между клиентом и сервером. HTTP используется для передачи гипертекстовых документов, таких как веб-страницы, изображения и другие ресурсы.

URL (Uniform Resource Locator) является адресом ресурса в сети.

Он определяет местонахождение ресурса и используется для доступа к нему. DNS (Domain Name System) предоставляет механизм преобразования удобочитаемых имен доменов в IP-адреса, которые используются для связи с серверами.

DNS (Domain Name System) — это система доменных имен, которая предоставляет механизм преобразования удобочитаемых имен доменов в IP-адреса, которые используются для связи с серверами.

HTTP-запрос состоит из нескольких частей: метод запроса (GET, POST, PUT, DELETE и другие), URL ресурса, заголовки запроса (например, User-Agent, Content-Type), а также тело запроса (для POST-запросов). HTTP-ответ также имеет свою структуру и содержит код состояния, заголовки ответа (например, Content-Type, Content-Length) и тело ответа, которое содержит данные, отправленные сервером.

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Соотнесите:

1. HTTP	A. Механизм преобразования удобочитаемых имен доменов в IP-адреса, которые используются для связи с серверами.
2. DNS	B. Адрес, который определяет местоположение ресурса в интернете,
3. URL	C. Набор правил и соглашений, которые используются для передачи данных между клиентами и веб-серверами в интернете
4. Протокол	D. Набор правил и соглашений, определяющих формат и последовательность обмена данных между устройствами в компьютерных сетях.

БИБЛИОТЕКА REQUESTS

Библиотека requests является одной из наиболее популярных библиотек для работы с HTTP-запросами и HTTP-ответами в Python.

Она обеспечивает удобный и простой интерфейс для отправки запросов на серверы, обработки ответов и выполнения различных операций с веб-ресурсами.

```python
import requests

response = requests.get("https://example.com")
print(response.status_code) # Выводит код состояния ответа
print(response.text) # Выводит данные, полученные от сервера
```

Основные поля ответа на запрос через requests включают код состояния (status code), который указывает на успешность или ошибку выполнения запроса. Некоторые распространенные коды состояния: 200 (OK), 404 (Not Found), 500 (Internal Server Error). Другие важные поля ответа могут включать заголовки ответа, содержащие информацию о типе данных, размере ответа и других метаданных.

```python
import requests

response = requests.get("https://example.com")
print(response.status_code) # Выводит код состояния ответа (200 - успешный запрос)
print(response.headers) # Выводит заголовки ответа
```

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Объясните, что происходит в этом фрагменте кода:

```python
import requests
response = requests.get('https://httpbin.org/get')
print(response.text)
```

В чем разница с этим фрагментом?

```python
import requests
response = requests.get('https://httpbin.org/head')
print(response.status_code)
print(response.headers)
```

API, COOKIE

Понятие API (Application Programming Interface) относится к набору определенных правил и протоколов, которые позволяют различным приложениям взаимодействовать между собой.

Работа с API включает отправку HTTP-запросов на определенные URL-адреса, обработку полученных ответов и взаимодействие с данными, предоставляемыми API.

```python
import requests

response = requests.get("https://api.github.com/users/github")
data = response.json()

for k, v in data.items():
  print(f"{k}: {v}")
```

Cookie представляет собой небольшую текстовую информацию, которая отправляется сервером и хранится на стороне клиента.

Она используется для идентификации пользователя и хранения состояния между запросами. Библиотека requests позволяет удобно работать с cookie.

```python
import requests

response = requests.get("https://example.com")
print(response.cookies) # Выводит cookie, полученные от сервера

cookies = {"session_id": "123456789"}
```


```
response = requests.get("https://example.com", cookies=cookies)
```

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Объясните, что происходит в этом фрагменте кода:

```python
import requests

response = requests.get('https://random-data-api.com/api/v2/users?size=10')
res=response.json()

for person in res:
print(f"{person['last_name']:15} {person['first_name']:15} {person['email']}")
```

ПРАКТИЧЕСКАЯ РАБОТА

1. Написать программу, которая с помощью библиотеки requests будет проверять доступность сайта, адрес которого передан в аргументе.
2. Изучите API ресурса http://www.boredapi.com/

   Напишите программу, которая запрашивает у пользователя, чем он хотел бы заняться, и сколько всего будет участников, и с помощью изученного API предлагает вариант активности.

ПОЛЕЗНЫЕ МАТЕРИАЛЫ

1. Библиотека Python Requests: документация на русском, get и post
2. Простым языком об HTTP / Хабр