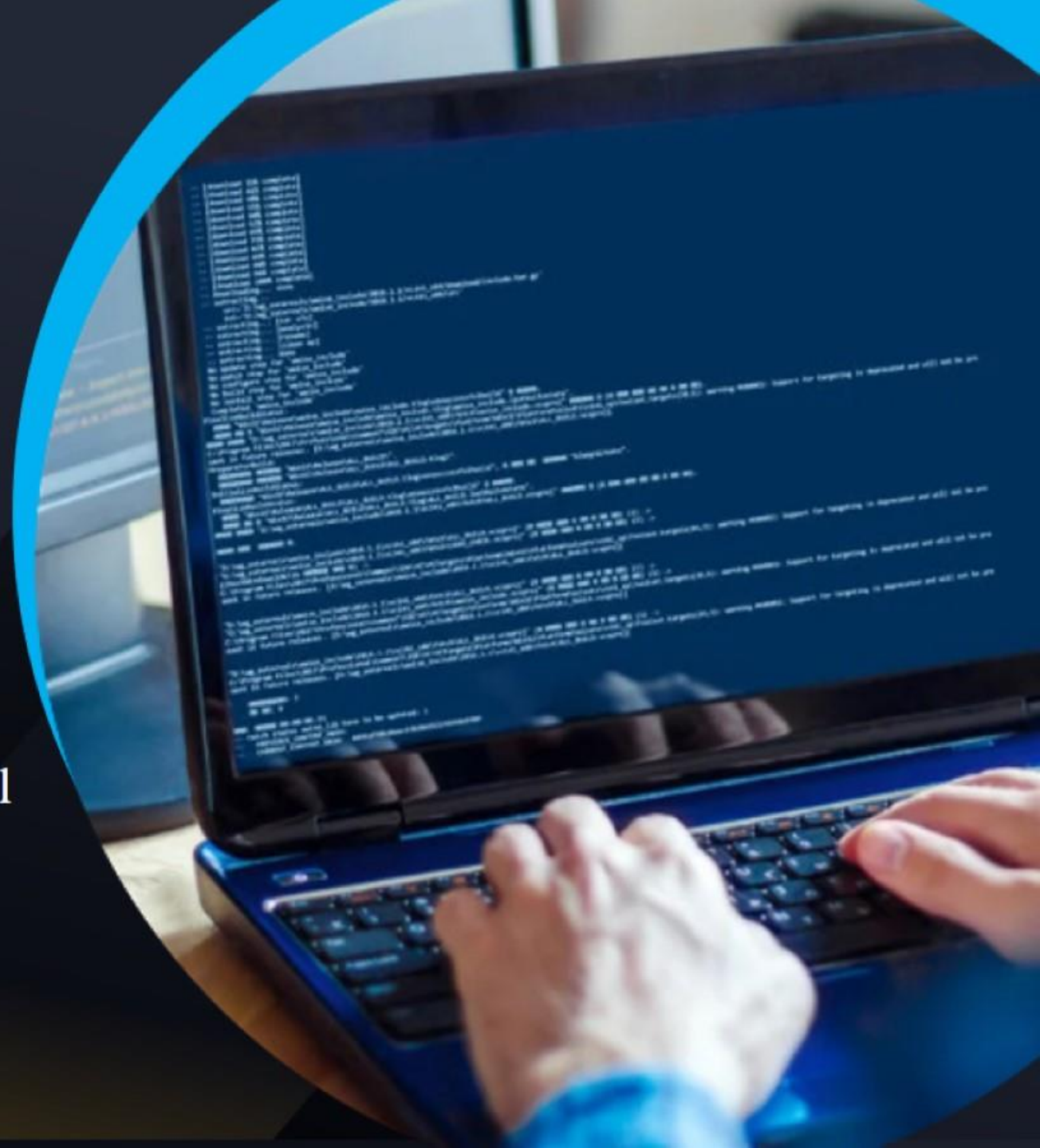


Network Programming in Python

Prepared by Dr. Hamada I. AbdulWakel

Lecture 2



The background features a series of vertical panels with rounded ends. The first panel on the left shows a person's hands typing on a laptop. The subsequent panels are semi-transparent and contain various white line-art icons: a graduation cap, a play button, a target with an arrow, an open book, a magnifying glass, and a group of people. The overall color scheme is warm, with shades of orange and yellow.

Basics of Network Programming – Part I

Lecture 2

Prepared by
Dr. Hamada I. AbdulWakel

Lecture

02

RECAP: LECTURE 1

- Network definition, importance, components, and types.
- Network programming levels.
- The virtual environments.

Lecture

02

LECTURE MAIN POINTS

- Service Oriented Architecture (SOA).
- Application Program Interface (API).
- Wire Protocol Formats.
- API modules in Python.

Lecture

02

SERVICE-ORIENTED ARCHITECTURE (SOA)

Service Orientation?

A way of integrating your business as linked services and the outcomes that they bring

A Composite Application

A set of related & integrated services that support a business process built on an SOA



A Service?

A repeatable business task - e.g., check customer; open new account

Service-Oriented Architecture

An IT architectural style that supports service orientation

SERVICE-ORIENTED ARCHITECTURE (SOA)

Cont'd

- **SOA** is a method of software development that uses software components called services to create business applications.
- Each **service** provides a business capability, and services can also communicate with each other across platforms and languages.
- Developers use SOA to **reuse services** in different systems or combine several independent services to perform complex tasks.

COMPONENTS OF SOA

- **Services:** the basic building blocks of SOA.
 - They can be private — available only to internal users of an organization
 - Or public — accessible over the internet to all.
- **Heterogeneous application** which developed using different platform cannot communicate with each other because they did not know the language of each other.
- **Web services** are language independent and platform independent. Web services are mainly used for cross communication between heterogeneous technologies.

Lecture

02

COMPONENTS OF SOA

Cont'd



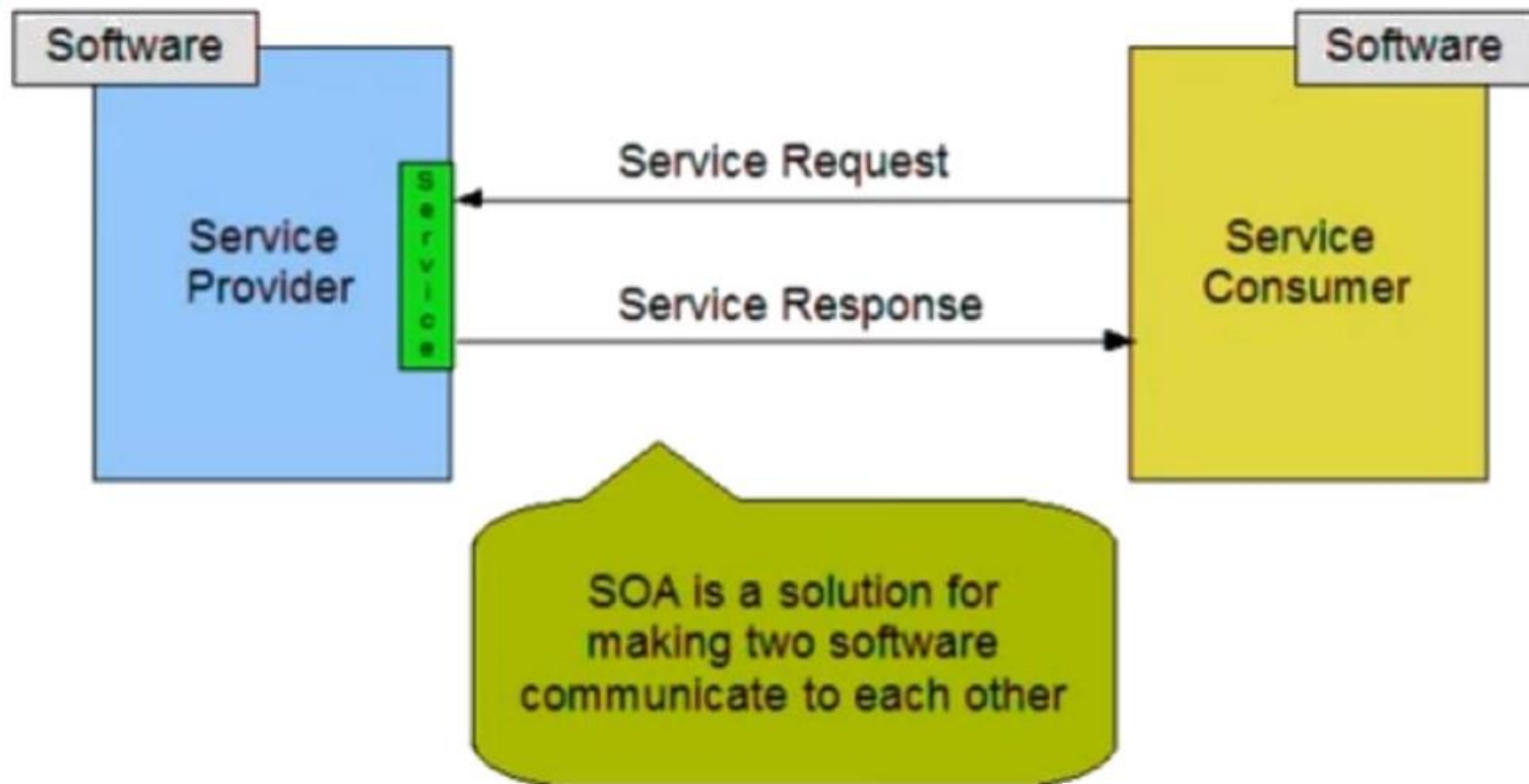
COMPONENTS OF SOA

Cont'd

- **Service provider:** creates, maintains, and provides one or more services that others can use.
- **Service consumer:** requests the service provider to run a specific service. It can be an entire system, application, or other service.
- **The service contract:** defines the nature of the service and its associated terms and conditions, such as **the prerequisites** for using the service, service cost, and quality of service provided.

COMPONENTS OF SOA

Cont'd



COMPONENTS OF SOA

Cont'd

- Service's prerequisites are the rules that determine how the data transmission over a network.
- These rules are called communication protocols.
 - Simple Object Access Protocol (SOAP).
 - RESTful HTTP.
 - Java Message Service (JMS).

Lecture

02

APPLICATION PROGRAMMING INTERFACE (API)

- **API** could be a **software system interface** that enables two applications to act with one another with **no user intervention**.
 - APIs offer a **secure and standardized** means for applications to communicate.
- It defines:
 - **URL** style to make requests.
 - The **required data** for the service.
 - The **data formats** that will be received.
- All web services are APIs, but not all APIs are web services (e.g., when Photoshop needs to **save as** the image).

Lecture

02

SOA SERVICE EXAMPLE

- Retrieving a credit card statement online.
- Reservation in a hotel.
- Car renting.



API



DB



WIRE PROTOCOL

- A **protocol** is a set of rules that all parties follow, so that we can predict each other's behavior.
- Whenever you transmit information over a network, that information is transformed from a binary representation into another representation that was designed for network use.
- The form that it takes for network transmission is called its **wire representation**.

- We need to **unify the format** between programs so that program can expect this format and thus extract the information from it easily.
- A **wire protocol** refers to a way of getting data from point to point.
- There are two common wire representations:
 - eXtensible Markup Language (XML) format.
 - JavaScript Object Notation (JSON) format.

Lecture

02

WIRE PROTOCOL

Cont'd

Python program
uses a dictionary
data structure

Data written in wire protocol

PHP program
uses an array
data structure

Lecture

02

XML WIRE PROTOCOL

Python program
uses a dictionary
data structure

```
<car>  
  <type>  
    Mercedes  
  </type>  
  <color>  
    Black  
  </color>  
</car>
```

PHP program
uses an array
data structure

- XML is a markup language much like HTML and was designed to store and transport data.
- With XML, the author must define both the tags and the document structure, where XML tags are not predefined like HTML tags.
- XML provides a software- and hardware-independent way of storing, transporting, and sharing data.

Lecture

02

JSON WIRE PROTOCOL

Python program
uses a dictionary
data structure

```
{  
  "type": "Mercedes",  
  "color": "Black"  
}
```

PHP program
uses an array
data structure

JSON WIRE PROTOCOL

Cont'd

- JSON is a lightweight data-interchange format.
- It is commonly used for transmitting data in web applications
- Each property has two parts: **a name and a value**, separated by a colon.
- XML is hard to read and leaves much to be desired in terms of coding. JSON was created to resolve these issues.

Lecture

02

JSON IN PYTHON

- Python has a built-in package/module called `json`, which can be used to work with JSON data.
- The `json.loads()` method is used to **parse** a JSON string and the result is a Python dictionary.
- The `json.dumps()` method is used to **convert** a Python dictionary to a JSON string.

JSON IN PYTHON

Cont'd

- Code:

```
import json
data = """
    { "Name": "Midocean university",
      "Location": "United Arab Emirates",
      "Contacts":
        { "email": "contact@midocean.ae",
          "tel": "+966-505281134"}
    }"""
```

JSON IN PYTHON

Cont'd

- Code:

```
import json
.....
universityinfo = json.loads(data)
print("Name:", universityinfo["name"])
print("Email:", universityinfo["Contact"]["email"])

universityinfo = json.dumps(data)
print(universityinfo)
```

Lecture

02

WEB PAGE ACCESSING AND
READING

<https://en.wikipedia.org/wiki/URL?key=value&life=42#History>

- Web pages are full of original, high-quality content that could be used for different reasons.
- URL parts:
 - Protocol part, such as **https**.
 - Host name part, such as **en.wikipedia.org**.
 - Port number part, getting from the protocol.
 - Path part, such as **wiki/URL**.
 - Querystring, such as **key and life**.
 - Fragment, such as **#History**.

URLLIB MODULE

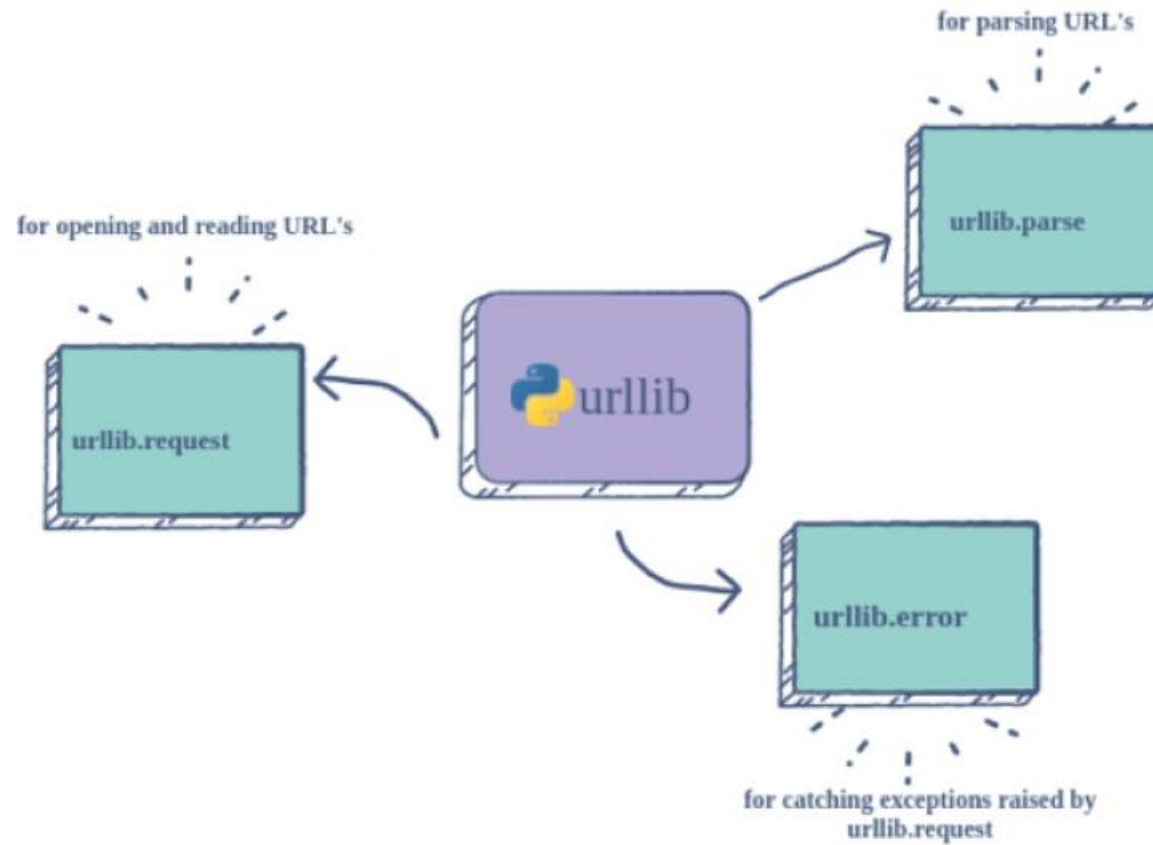
- **Urllib** is a Python package that allows you to access and interact with web pages using their URL's (Uniform Resource Locator).
- This facility of urllib provides the flexibility to run GET, POST, and PUT methods and mock the JSON data in the Python code.
- Urllib module is not a built-in module, you need to install it manually.

`pip install urllib3`

Lecture

02

URLLIB MODULE COMPONENTS



URLLIB MODULE COMPONENTS

Cont'd

- **Urllib.request**
 - Used for opening and reading web pages. The main method is `urlopen()` to open the specified URL.
- **Urllib.parse**
 - Used for parsing the parts of the URLs.
- **Urllib.error**
 - Used for catching exceptions encountered when opening URLs and other errors.

Lecture

02

URLLIB DESCRIPTION

- Code:

```
import urllib
from urllib import request

help(urllib)
print(dir(request))
resp = request.urlopen('https://www.wikipedia.org')
print(type(resp))
print(dir(resp))
```

URLLIB DESCRIPTION

Cont'd

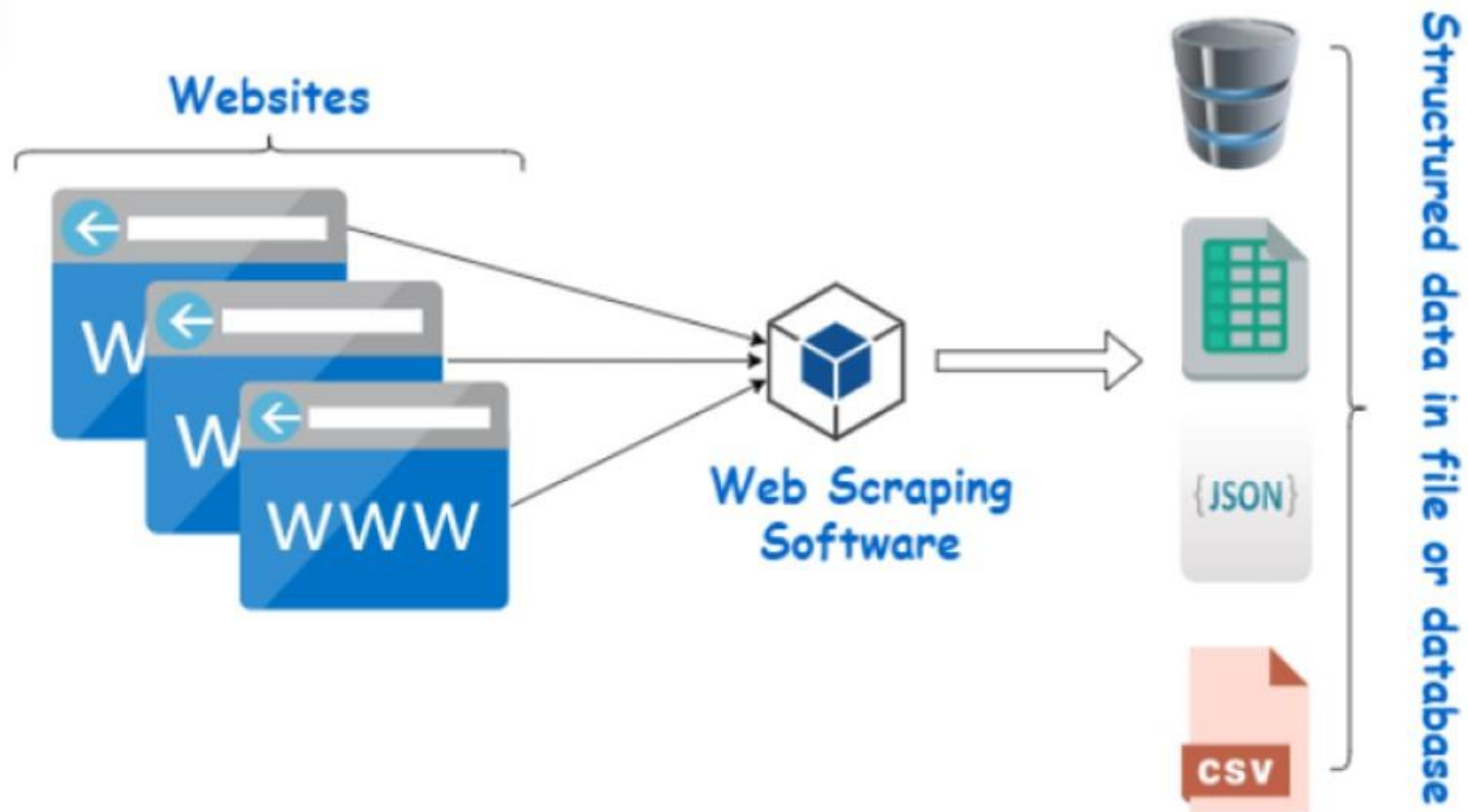
- Code:

```
.....  
print(resp.code)  
print(resp.length)  
data = resp.read()  
print(type(data)) #page as bytes  
html = data.decode()  
print(type(html)) #page as string  
print(html) #page as html code
```

Lecture

02

WEB SCRAPING



- A **web scraping** software will automatically crawl and extract data from multiple pages of websites based on your requirement.
 - Usually, the web pages are described as unstructured data.
- Web Scraping is an automatic way to retrieve unstructured data from a website and store them in a structured format.
 - By automating this process, these software can extract huge amounts of data in a very short time.

- **A web scraper** works as following:
 - Step 1: Making an HTTP request to a server.
 - Step 2: Extracting and parsing (or breaking down) the website's code.
 - Step 3: Saving the relevant data locally

Lecture

02

WEB SCRAPING EXAMPLE

- Code:

```
from urllib.request import urlopen
page = urlopen('http://data.pr4e.org/romeo.txt')
word_count = dict()
for line in page:
    words = line.decode().split()
    for word in words:
        word_count[word] = word_count.get(word, 0) + 1
print(word_count)
```

Lecture

02

REQUESTS MODULE

- The de facto standard for making HTTP requests in Python.
- One of the most common HTTP methods is GET. The GET method indicates that you're trying to get or retrieve data from a specified resource.
- It abstracts the complexities (in code writing and HTTP requests handling) of making requests behind a beautiful, simple API.
- To install the requests module.

`pip install requests`

Lecture

02

COMPLETE API EXAMPLE

- Using TVmaze.com API.



If you want to add TV information to your website or app then you've come to the right place!

We provide a free, fast and clean REST API that's easy to use, returns JSON and conforms to the HATEOAS and HAL principles. The root url is <https://api.tvmaze.com> and the available endpoints are documented below. If you have any questions or suggestions regarding the API, please post them on [our forums](#).

In addition to the free public API, there's a user-level API available for all [Premium](#) members. The documentation for the user API can be viewed [here](#).

COMPLETE API EXAMPLE

Cont'd

- Code:

```
import requests, json, pprint
url = "https://api.tvmaze.com/singlesearch/shows"
params = {"q": "the flash"}
response = requests.get(url, params)
if response.status_code == 200:
    print(response.text)    #without json style
    data = json.loads(response.text)
    print(data)             #json, but without good formating
    pprint.pprint(data)
```

COMPLETE API EXAMPLE

Cont'd

- Code:

```
if response.status_code == 200:
    data = json.loads(response.text)
    name = data['name']
    premiered = data['premiered']
    summary = data['summary']
    print(f'{name} premiered on {premiered}.')
    print(summary)
else:
    print(f'Error: {response.status_code}')
```

Lecture

02

QUESTIONS FOR ONLINE DISCUSSION

- Write a program to get the current weather for your location using <https://openweathermap.org/> website.



Thank You

