

Object Oriented Programming (Using Python)

UNIT- IV

Python to access Web Data : cont'd.

- Regular Expressions
- Extracting data
- Sockets
- Using the Developer Console to Explore HTTP and Retrieving Web Page
- Parsing Web Pages

Prof. R. MADANA MOHANA

Professor, Artificial Intelligence & Data Science

<http://rmadanamohana.com/>

Reference

<https://www.coursera.org/learn/python-network-data>

<https://github.com/I-am-Harsh/Using-Python-to-Access-Web-Data>

<https://www.py4e.com/code3/>

[https://eng.libretexts.org/Bookshelves/Computer Science/Programming Languages/Book%3A Python for Everybody \(Severance\)](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book%3A_Python_for_Everybody_(Severance))

<http://docs.python.org/library/socket.html>

<https://youtu.be/T0rYSFPAR0A>

Object Oriented Programming (Using Python)

Using the Developer Console to Explore HTTP

HTTP - Hypertext Transfer Protocol

- The dominant **Application Layer Protocol** on the Internet
- Invented for the **Web** - to **Retrieve HTML, Images, Documents**, etc.
- Extended to retrieve data in addition to documents – **RSS (Really Simple Syndication)**, a **web feed** that allows users and applications to access updates to websites in a standardized, computer-readable format), **Web Services**, etc.
- **Basic Concept:**
 - ✓ Make a Connection
 - ✓ Request a document
 - ✓ Retrieve the Document
 - ✓ Close the Connection

HTTP - Hypertext Transfer Protocol

- The **H**yper**T**ext **T**ransfer **P**rotocol is the set of rules to allow browsers to retrieve web documents from servers over the Internet
- In **Python**, we can interact with the **HTTP protocol** using various libraries, but one commonly used library is `'requests'`

HTTP - Hypertext Transfer Protocol

- The **H**yper**T**ext **T**ransfer **P**rotocol is the set of rules to allow browsers to retrieve web documents from servers over the Internet
- In **Python**, we can interact with the **HTTP protocol** using various libraries, but one commonly used library is '**requests**'
- **Example:**

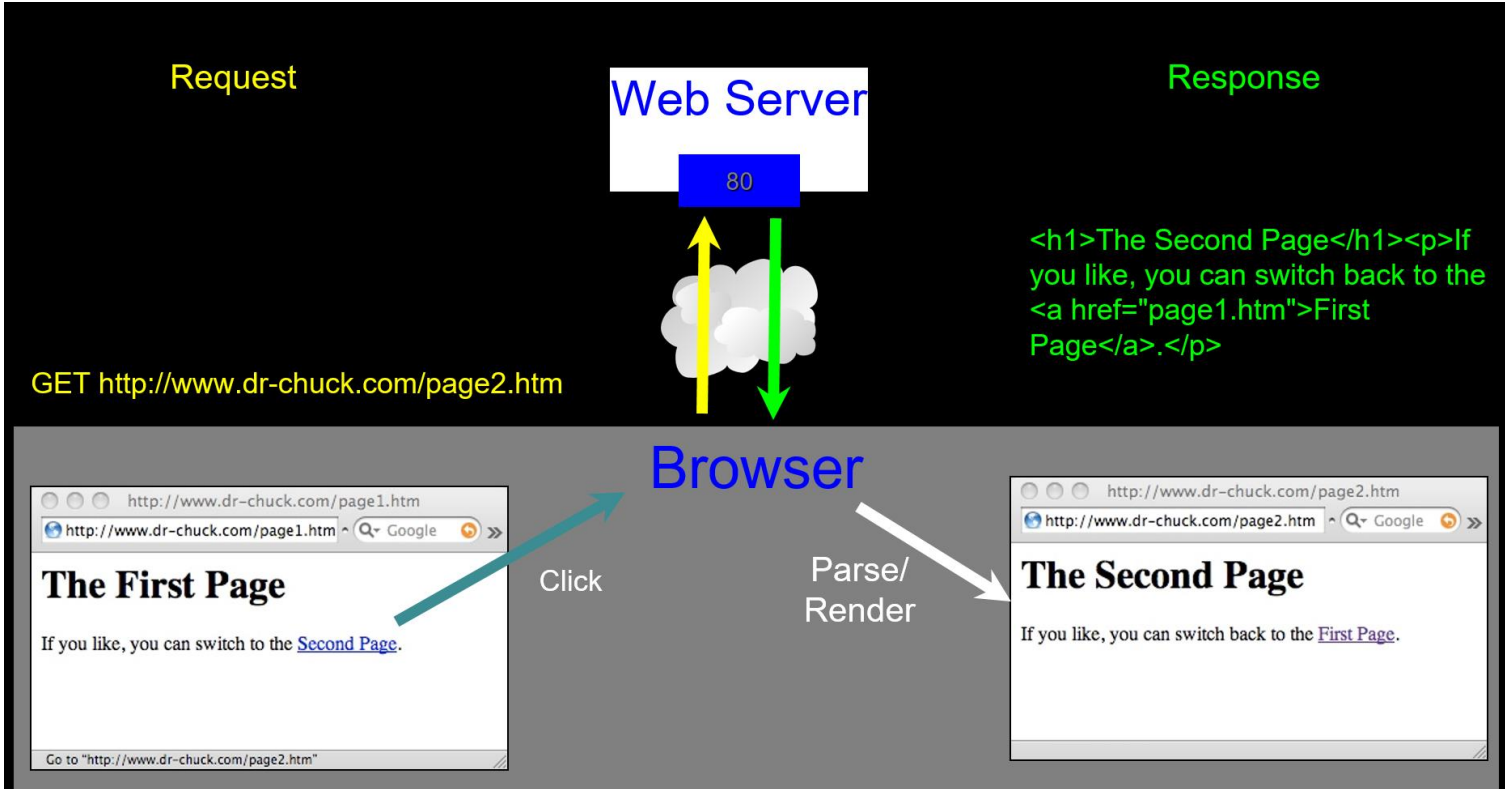
`http://www.dr-chuck.com/page1.htm`

protocol **host** **document**

Getting Data from the Server

- Each time the user clicks on an **anchor tag** with an **href= value** (**href-hypertext reference**) to switch to a **new page**, the **browser** makes a connection to the **web server** and issues a **“GET”** request - to **GET** the content of the page at the specified **URL**
- The **server** returns the **HTML** document to the **browser**, which formats and displays the document to the user

Getting Data from the Server - Example

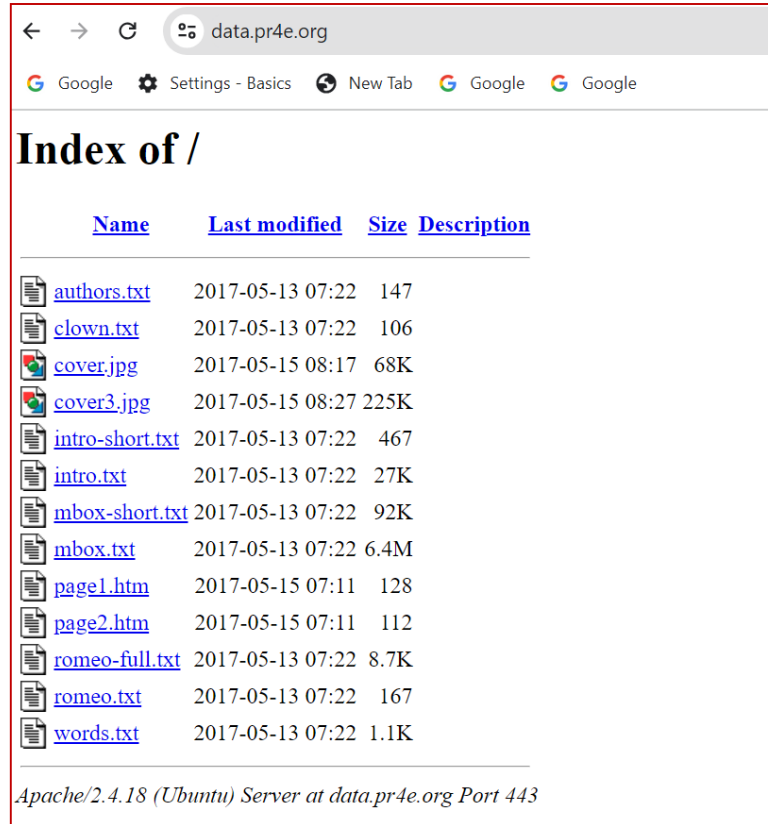


Making an HTTP request














- To request a **response** from the **server**, there are mainly two methods:
 - ✓ **GET**: To request data from the server
 - ✓ **POST**: To submit data to be processed to the server
- To make **HTTP requests** in **Python**, we can use several **HTTP libraries** like:
 - ✓ `httplib`
 - ✓ `urllib`
 - ✓ `requests`

An HTTP request in Python using Sockets

Example URL: `data.pr4e.org`



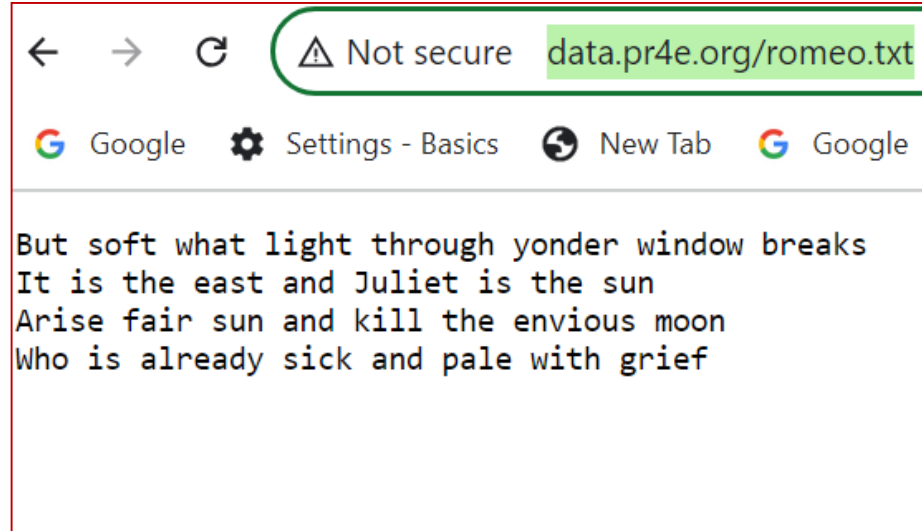
The screenshot shows a web browser window with the address bar containing `data.pr4e.org`. The browser tabs include "Google", "Settings - Basics", "New Tab", and two "Google" tabs. The main content area displays the "Index of /" page, which is a directory listing. The listing has columns for "Name", "Last modified", and "Size". Each entry includes a small icon representing the file type (e.g., a document icon for .txt files, a picture icon for .jpg files, and a web page icon for .htm files).

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 authors.txt	2017-05-13 07:22	147	
 clown.txt	2017-05-13 07:22	106	
 cover.jpg	2017-05-15 08:17	68K	
 cover3.jpg	2017-05-15 08:27	225K	
 intro-short.txt	2017-05-13 07:22	467	
 intro.txt	2017-05-13 07:22	27K	
 mbox-short.txt	2017-05-13 07:22	92K	
 mbox.txt	2017-05-13 07:22	6.4M	
 page1.htm	2017-05-15 07:11	128	
 page2.htm	2017-05-15 07:11	112	
 romeo-full.txt	2017-05-13 07:22	8.7K	
 romeo.txt	2017-05-13 07:22	167	
 words.txt	2017-05-13 07:22	1.1K	

Apache/2.4.18 (Ubuntu) Server at data.pr4e.org Port 443

An HTTP request in Python using Sockets

Example data: <http://data.pr4e.org/romeo.txt>



An HTTP request in Python using Sockets

Python Program: `htmlsocket.py`

```
*htmlsocket.py - C:\Users\rmmna\AppData\Local\Programs\Python\Python310\htmlsocket.py (3.10.0)*
File Edit Format Run Options Window Help
import socket

mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(('data.pr4e.org', 80))
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0\r\n\r\n'.encode()

mysock.send(cmd)

while True:
    data = mysock.recv(512)
    if (len(data) < 1):
        break
    print(data.decode(), end='')
mysock.close()
```

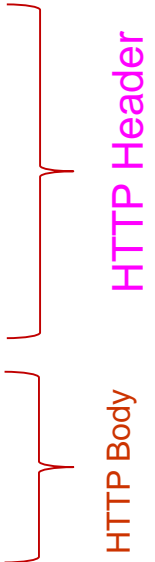
An HTTP request in Python using Sockets

Python Program: `htmlsocket.py`

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\rmmna\AppData\Local\Programs\Python\Python310\htmlsocket.py
HTTP/1.1 200 OK
Date: Thu, 13 Jul 2023 04:21:50 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sat, 13 May 2017 11:22:22 GMT
ETag: "a7-54f6609245537"
Accept-Ranges: bytes
Content-Length: 167
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Connection: close
Content-Type: text/plain

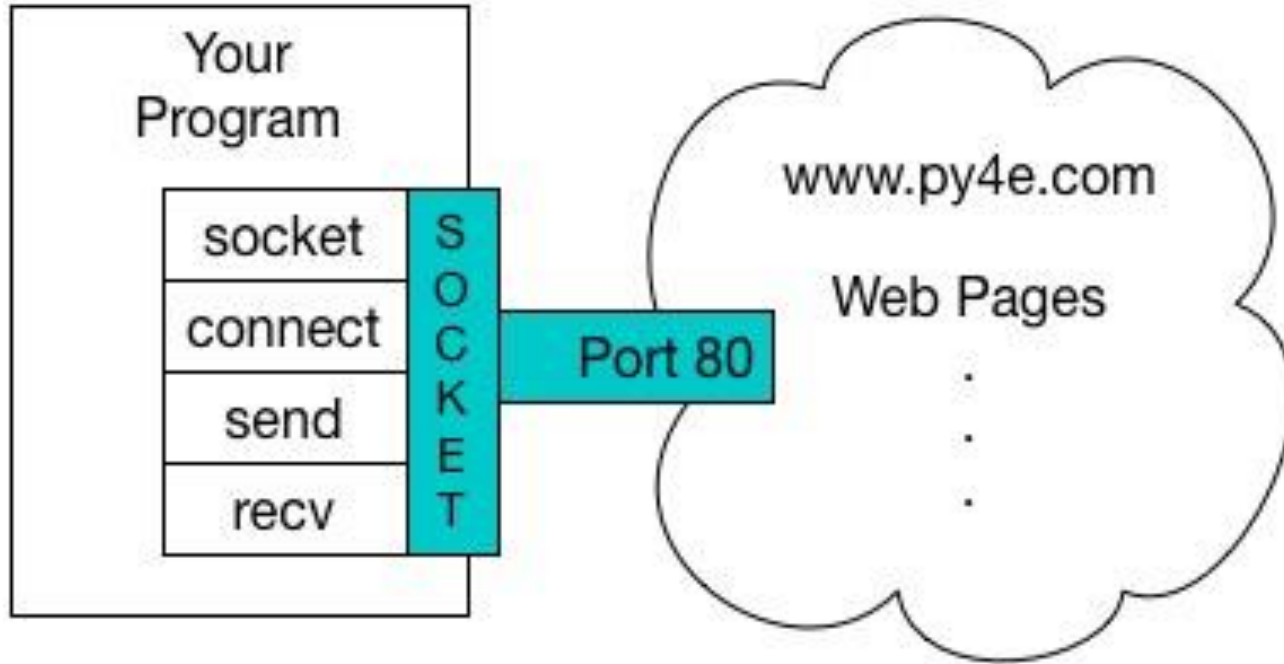
But soft what light through yonder window breaks
It is the east and Juliet is the sun
Arise fair sun and kill the envious moon
Who is already sick and pale with grief

>>>
```



An HTTP request in Python using Sockets

Python Program: [htmlsocket.py](#)



An HTTP request in Python using Sockets

Python Program: `htmlsocket.py`

```
bytes.decode(encoding="utf-8", errors="strict")
```

```
bytearray.decode(encoding="utf-8", errors="strict")
```

Return a string decoded from the given bytes. Default encoding is `'utf-8'`. `errors` may be given to set a different error handling scheme. The default for `errors` is `'strict'`, meaning that encoding errors raise a `UnicodeError`. Other possible values are `'ignore'`, `'replace'` and any other name registered via `codecs.register_error()`, see section [Error Handlers](#). For a list of possible encodings, see section [Standard Encodings](#).

```
str.encode(encoding="utf-8", errors="strict")
```

Return an encoded version of the string as a bytes object. Default encoding is `'utf-8'`. `errors` may be given to set a different error handling scheme. The default for `errors` is `'strict'`, meaning that encoding errors raise a `UnicodeError`. Other possible values are `'ignore'`, `'replace'`, `'xmlcharrefreplace'`, `'backslashreplace'` and any other name registered via `codecs.register_error()`, see section [Error Handlers](#). For a list of possible encodings, see section [Standard Encodings](#).

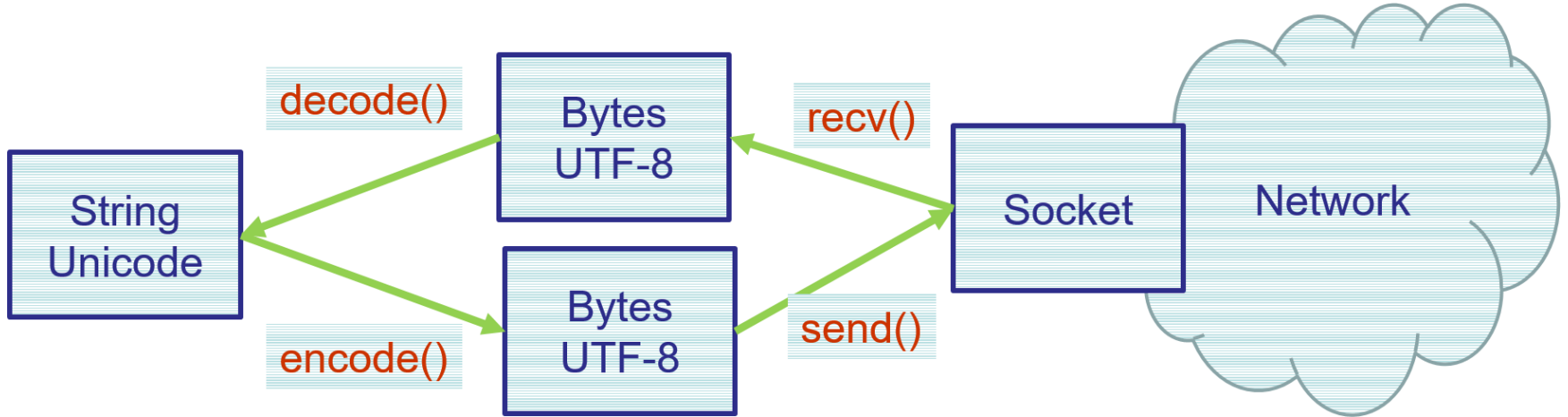
UTF-8



UTF-8 is a variable-length character encoding standard used for electronic communication. Defined by the Unicode Standard, the name is derived from Unicode Transformation Format – 8-bit. UTF-8 is capable of encoding all 1,112,064 valid character code points in Unicode using one to four one-byte code units. [Wikipedia](#)

An HTTP request in Python using Sockets

Python Program: `htmlsocket.py`



Retrieving Web Pages

Making HTTP Easier With urllib

- To make **HTTP requests** in **Python**, we can use several **HTTP libraries** like:
 - ✓ `httplib`
 - ✓ `urllib`
 - ✓ `requests`

httplib - HTTP protocol client

- The `httplib` module has been renamed to `http.client` in Python 3. The `2to3` tool will automatically adapt imports when converting our sources to Python 3.
- This module defines classes which implement the **client side** of the `HTTP` and `HTTPS` protocols.
- It is normally not used directly - the module `urllib` uses it to handle `URLs` that use `HTTP` and `HTTPS`.

httplib - HTTP protocol client

- Here's an example program demonstrating how to use `http.client` to make a simple **HTTP GET** request:

httplib - HTTP protocol client

```
httpclient.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/httpclient.py (3.10.0)
File Edit Format Run Options Window Help
import http.client

# Create an HTTP connection
conn = http.client.HTTPSConnection("www.cbit.ac.in")

# Send an HTTP request
conn.request("GET", "/")

# Get the response from the server
response = conn.getresponse()

# Print the response status code
print("Status:", response.status)

# Read the response data
data = response.read()

# Print the response data
print("Response:")
print(data.decode())

# Close the connection
conn.close()
```

httplib - HTTP protocol client

Output:

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/httpclient.py
Status: 200
Response:
Squeezed text (2219 lines).
>>>
```

httplib - HTTP protocol client

Output:

```
Squeezed Output Viewer
<!DOCTYPE html>
<html>
<head>
  <title>Home - CBIT</title>
<meta charset='utf-8'>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!--Styles-->
<link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/bootstrap.min.css"/>
<link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/styles.css"/>
<link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/unslider.css"/>
<link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/owl.carousel.min.css"/>
<!--fonts url-->
<link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/font-awesome.min.css"/>

<!--Favicon-->
<link rel="icon" type="image/png" href="favicon/favicon-32x32.png" sizes="32x32" />
<link rel="icon" type="image/png" href="favicon/favicon-16x16.png" sizes="16x16" />
<meta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1' />
<meta property="og:title" content="Home"/>
<meta property="og:description" content="" />
<meta property="og:type" content="website" />
<meta name="twitter:card" content="summary">
<meta name="twitter:title" content="Home"/>
<meta name="twitter:description" content="" />
<meta name="author" content="Admin"/>

  <!-- This site is optimized with the Yoast SEO plugin v1.7.7 - https://yoast.com/wordpress/plugins/seo/ -->
<title>Home - CBIT</title>
<link rel="canonical" href="https://www.cbit.ac.in/" />
<meta property="og:url" content="https://www.cbit.ac.in/" />
<meta property="og:site_name" content="CBIT" />
<meta property="article:publisher" content="https://www.facebook.com/CBIThyderabad/" />
<meta property="article:modified_time" content="2020-03-14T04:42:31+00:00" />
<meta name="twitter:site" content="@CBITHyd" />
<script type="application/ld+json" class="yoast-schema-graph">{"@context":"https://schema.org","@graph":[{"@type":"Organization","@id":"https://www.c
<!-- / Yoast SEO plugin. -->

<link rel='dns-prefetch' href='//www.google.com' />
<link rel='dns-prefetch' href='//s.w.org' />
```

httplib - HTTP protocol client

- In this example, we create an `HTTPSConnection` object using the server address ("`www.cbit.ac.in`" in this case).
- We then send an `HTTP GET` request using the `request()` method, specifying the path ("`/`" in this case).
- The `server` responds with an `HTTP response`, which we retrieve using the `getresponse()` method.
- We can access various properties of the response, such as the `status code` (`response.status`) and the `response data` (`response.read()`).
- In this example, we `print` the `status code` and `decode` the `response data` as a `string` using the `decode()` method before printing it.
- Finally, we `close` the `connection` using the `close()` method.

Using urllib in Python

- In `Python`, the `urllib` module provides a collection of modules for working with `URLs`.
- One of the commonly used `submodules` is `urllib.request`, which allows you to make `HTTP requests` and handle `responses`.

Using urllib in Python

- Here's an example program demonstrating how to use `urllib.request` to make a simple **HTTP GET request**:

Using urllib in Python

```
urllibex.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urllibex.py (3.10.0)
File Edit Format Run Options Window Help
import urllib.request

# Make an HTTP GET request
response = urllib.request.urlopen("https://www.cbit.ac.in/")

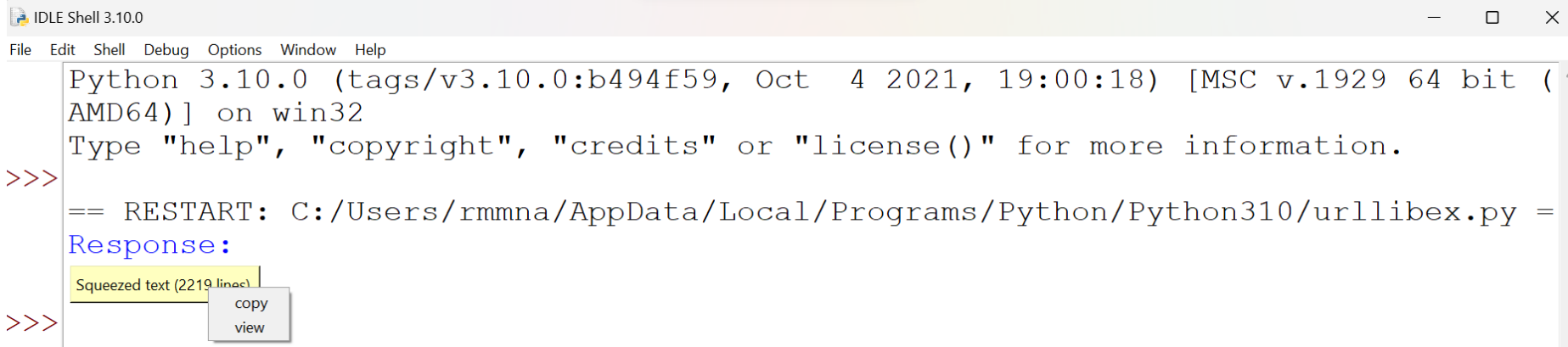
# Read the response data
data = response.read()

# Print the response data
print("Response:")
print(data.decode())

# Close the response
response.close()
```

Using urllib in Python

- **Output:**



```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urllibex.py =
Response:
>>>
```

Squeezed text (2219 lines)

copy
view

Using urllib in Python

- **Output:**

```
Squeezed Output Viewer
<!DOCTYPE html>
<html>
<head>
  <title>Home - CBIT</title>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!--Styles-->
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/bootstrap.min.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/styles.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/unslider.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/owl.carousel.min.css"/>
  <!--fonts url-->
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/font-awesome.min.css"/>

  <!--Favicon-->
  <link rel="icon" type="image/png" href="favicon/favicon-32x32.png" sizes="32x32" />
  <link rel="icon" type="image/png" href="favicon/favicon-16x16.png" sizes="16x16" />
  <meta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1' />
  <meta property="og:title" content="Home"/>
  <meta property="og:description" content="" />
  <meta property="og:type" content="website" />
  <meta name="twitter:card" content="summary" />
  <meta name="twitter:title" content="Home" />
  <meta name="twitter:description" content="" />
  <meta name="author" content="Admin" />

  <!-- This site is optimized with the Yoast SEO plugin v17.7 - https://yoast.com/wordpress/plugins/seo/ -->
  <title>Home - CBIT</title>
  <link rel="canonical" href="https://www.cbit.ac.in/" />
  <meta property="og:url" content="https://www.cbit.ac.in/" />
  <meta property="og:site_name" content="CBIT" />
  <meta property="article:publisher" content="https://www.facebook.com/CBITHyderabad/" />
  <meta property="article:modified_time" content="2020-03-14T04:42:31+00:00" />
  <meta name="twitter:site" content="@CBITHyd" />
  <script type="application/ld+json" class="yoast-schema-graph">{"@context":"https://schema.org","@graph":[{"@type":"Organization","@id":"https://www.cbit.ac.in/"}]}
  <!-- / Yoast SEO plugin. -->

  <link rel='dns-prefetch' href='//www.google.com' />
  <link rel='dns-prefetch' href='//s.w.org' />
```

Using urllib in Python

- In this example, we use the `urlopen()` function from `urllib.request` to open a connection to the specified URL ("<https://www.cbit.ac.in/>" in this case) and make an **HTTP GET request**.
- The `urlopen()` function returns a file-like object that represents the response.
- We can then use the `read()` method on the response object to read the response data as bytes.
- In this example, we **decode** the **response data** as a **string** using the `decode()` method before printing it.
- Finally, we **close** the response using the `close()` method to free up system resources.
- **Note** that the `urllib` module provides various other functionalities for working with **URLs**, such as **URL encoding**, **parsing**, and **handling errors**.

Using urllib in Python – Example2

urllibsocket.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urllibsocket.py (3.10.0)

File Edit Format Run Options Window Help

```
import urllib.request, urllib.parse, urllib.error

fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')
for line in fhand:
    print(line.decode().strip())
```

IDLE Shell 3.10.0

File Edit Shell Debug Options Window Help

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urllibsocket.py
But soft what light through yonder window breaks
It is the east and Juliet is the sun
Arise fair sun and kill the envious moon
Who is already sick and pale with grief
>>>
```

Using urllib in Python – Example3

```
urlwords.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urlwords.py (3.10.0)
File Edit Format Run Options Window Help
import urllib.request, urllib.parse, urllib.error

fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')

counts = dict()
for line in fhand:
    words = line.decode().split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1
print(counts)
```


Using urllib in Python – Example3

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/urlwords.py =
{'But': 1, 'soft': 1, 'what': 1, 'light': 1, 'through': 1, 'yonder': 1, 'window':
1, 'breaks': 1, 'It': 1, 'is': 3, 'the': 3, 'east': 1, 'and': 3, 'Juliet': 1, 'sun
': 2, 'Arise': 1, 'fair': 1, 'kill': 1, 'envious': 1, 'moon': 1, 'Who': 1, 'alread
y': 1, 'sick': 1, 'pale': 1, 'with': 1, 'grief': 1}
>>>
```

http requests

- To make **HTTP requests** in **Python**, we can use the popular '**requests**' **library**.
- The **requests library** provides a convenient and user-friendly **API** for making **HTTP requests** and **handling responses**.
- To install the **requests library** before running the program, we can install it using **pip**:

```
pip install requests
```

- By using the **requests library**, we can easily make **HTTP requests** and process the **responses** in our **Python** programs.

http requests - Example

```
*httprequest.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/httprequest.py (3.10.0)*
File Edit Format Run Options Window Help
import requests

# Make an HTTP GET request
response = requests.get("https://www.cbit.ac.in/")

# Print the response status code
print("Status Code:", response.status_code)

# Print the response headers
print("Headers:")
print(response.headers)

# Print the response body
print("Response:")
print(response.text)
```

http requests - Example

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/httprequest.py
Status Code: 200
Headers:
{'Date': 'Fri, 14 Jul 2023 09:22:23 GMT', 'Server': 'Apache/2.4.41 (Ubuntu)', 'Link': '<https://www.cbit.ac.in/wp-json/>; rel="https://api.w.org/", <https://www.cbit.ac.in/wp-json/wp/v2/pages/67>; rel="alternate"; type="application/json", <https://www.cbit.ac.in/>; rel=shortlink', 'Vary': 'Accept-Encoding', 'Content-Encoding': 'gzip', 'Content-Length': '15475', 'Keep-Alive': 'timeout=5, max=100', 'Connection': 'Keep-Alive', 'Content-Type': 'text/html; charset=UTF-8'}
Response:
Squeezed text (2219 lines).
>>>
```

http requests - Example

```
Squeezed Output Viewer
<!DOCTYPE html>
<html>
<head>
  <title>Home - CBIT</title>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!--Styles-->
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/bootstrap.min.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/styles.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/unslider.css"/>
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/owl.carousel.min.css"/>
  <!--fonts url-->
  <link rel="stylesheet" type="text/css" href="https://www.cbit.ac.in/wp-content/themes/CBIT/css/font-awesome.min.css"/>
  <!--Favicon-->
  <link rel="icon" type="image/png" href="favicon/favicon-32x32.png" sizes="32x32" />
  <link rel="icon" type="image/png" href="favicon/favicon-16x16.png" sizes="16x16" />
  <meta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1' />
  <meta property="og:title" content="Home"/>
  <meta property="og:description" content=""/>
  <meta property="og:type" content="website"/>
  <meta name="twitter:card" content="summary">
  <meta name="twitter:title" content="Home"/>
  <meta name="twitter:description" content=""/>
  <meta name="author" content="Admin"/>

  <!-- This site is optimized with the Yoast SEO plugin v17.7 - https://yoast.com/wordpress/plugins/seo/ -->
  <title>Home - CBIT</title>
  <link rel="canonical" href="https://www.cbit.ac.in/" />
  <meta property="og:url" content="https://www.cbit.ac.in/" />
  <meta property="og:site_name" content="CBIT" />
  <meta property="article:publisher" content="https://www.facebook.com/CBITHyderabad/" />
  <meta property="article:modified_time" content="2020-03-14T04:42:31+00:00" />
  <meta name="twitter:site" content="@CBITHyd" />
  <script type="application/ld+json" class="yoast-schema-graph">{"@context":"https://schema.org","@graph":[{"@type":"Organization","@id":"https://www.c
  <!-- / Yoast SEO plugin. -->

<link rel='dns-prefetch' href='//www.google.com' />
<link rel='dns-prefetch' href='//s.w.org' />
```

Close

Object Oriented Programming (Using Python)

Reading Web Pages

Reading web pages - Example

```
*readingwebpages.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/readingwebpages.py (3.10.0)*
File Edit Format Run Options Window Help
import urllib.request, urllib.parse, urllib.error

fhand = urllib.request.urlopen('http://www.dr-chuck.com/page1.htm')
for line in fhand:
    print(line.decode().strip())
```

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/readingwebpages.py
<h1>The First Page</h1>
<p>
If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">
Second Page</a>.
</p>
>>>
```