**COMPUTER COMMUNICATION NETWORKS**

**LAB EXPERIMENT 6**

**NAME: RAHIL SHARMA**

**PRN: 18070123062**

**BATCH: 2018-2022**

**DIVISION: G2; EA 3**

**AIM:** Write a  program which will take a URL string from the user and split string into its 4 components and display the components on screen.

**Theory:** In this experiment we will be splitting the URL into 4 different components. This experiment can also be titled as URL Splitter or Parser. We will be performing this experiment on Python 3.

Parse is defined as to break something down into its parts, particularly for study of the individual parts. The URL parsing functions focus on splitting a URL string into its components, or on combining URL components into a URL string.

Parse a URL into six components, returning a 6-item [named tuple](https://docs.python.org/3/glossary.html#term-named-tuple). This corresponds to the general structure of a URL: scheme://netloc/path;parameters?query#fragment. Each tuple item is a string, possibly empty. The components are not broken up into smaller parts (for example, the network location is a single string), and % escapes are not expanded. The delimiters as shown above are not part of the result, except for a leading slash in the *path* component, which is retained if present.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Index** | **Value** | **Value if not present** |
| scheme | 0 | URL scheme specifier | *scheme* parameter |
| netloc | 1 | Network location part | empty string |
| path | 2 | Hierarchical path | empty string |
| params | 3 | Parameters for last path element | empty string |
| query | 4 | Query component | empty string |
| fragment | 5 | Fragment identifier | empty string |
| username |  | User name | [None](https://docs.python.org/3/library/constants.html#None) |
| password |  | Password | [None](https://docs.python.org/3/library/constants.html#None) |
| hostname |  | Host name (lower case) | [None](https://docs.python.org/3/library/constants.html#None) |
| port |  | Port number as integer, if present |  |

**Procedure:**

1. This experiment was made on Python 3.
2. The libraries used are urllib.
3. urllib is a Python module that can be used for opening URLs. It defines functions and classes to help in URL actions. With Python you can also access and retrieve data from the internet like XML, HTML, JSON, etc. You can also use Python to work with this data directly.
4. Urlparse is The URL parsing functions focus on splitting a URL string into its components, or on combining URL components into a URL string.
5. Enter the URL you want to parse and declare it as a URL.
6. Pass the URLparse with argument as URL and declare it as an obj.
7. Now since the URL has been parsed then use the print statement to declare all the required components.
8. Print the protocol, Host, Domain, Port and Path.

**Code and OUTPUT of the program:**

**Challenge 1:** The protocol can be optional. (Do you see http written before the web site address on this web page ?)

1. **Code for the URL: http://xyz.abc.com:8080/folder1/folder2/index.html**

from urllib.parse import urlparse

url = str(input())

obj = urlparse(url)

print("The input is " + url)

print("The protocol of the URL is " + obj.scheme)

print("The Netloc is " + obj.netloc)

print("The domain or hostname is " + obj.hostname)

print(obj.port)

print("The file path is " + obj.path)

**Screenshots and Output of the Program:**

Graphical user interface, text, application

Description automatically generated

**Challenge 2**: Port can also be optional (There is no port written in the URL of this web page)

1. **Code for the URL:** [**https://www.cisco.com/c/en/us/training-events/training-certifications/certifications/associate/ccna.html**](https://www.cisco.com/c/en/us/training-events/training-certifications/certifications/associate/ccna.html)

from urllib.parse import urlparse

url = str(input())

obj = urlparse(url)

print("The input is " + url)

print("The protocol of the URL is " + obj.scheme)

print("The Netloc is " + obj.netloc)

print("The domain or hostname is " + obj.hostname)

print(obj.port)

print("The file path is " + obj.path)

**Screenshots and Output of the Program:**

**Graphical user interface, text, application, email

Description automatically generated**

**Challenge 3:** The host / domain can be a plain IP address instead of textual string.

1. **Code of the URL is: 106.214.146.172**

from urllib.parse import urlparse

url = str(input())

obj = urlparse(url)

print("The input is " + url)

print("The protocol of the URL is " + obj.scheme)

print("The Netloc is " + obj.netloc)

print(obj.hostname)

print(obj.port)

print(obj.path)

**Screenshot of the Output and Program:  
  
Graphical user interface, text, application, email

Description automatically generated**

**Conclusion: From this experiment we have learnt how to parse a URL and print its different components on Python. Also learnt how to split the URL into different components such as Port, Domain and Protocol. We also learnt how to use different libraries such as URLLIB and its functions.**