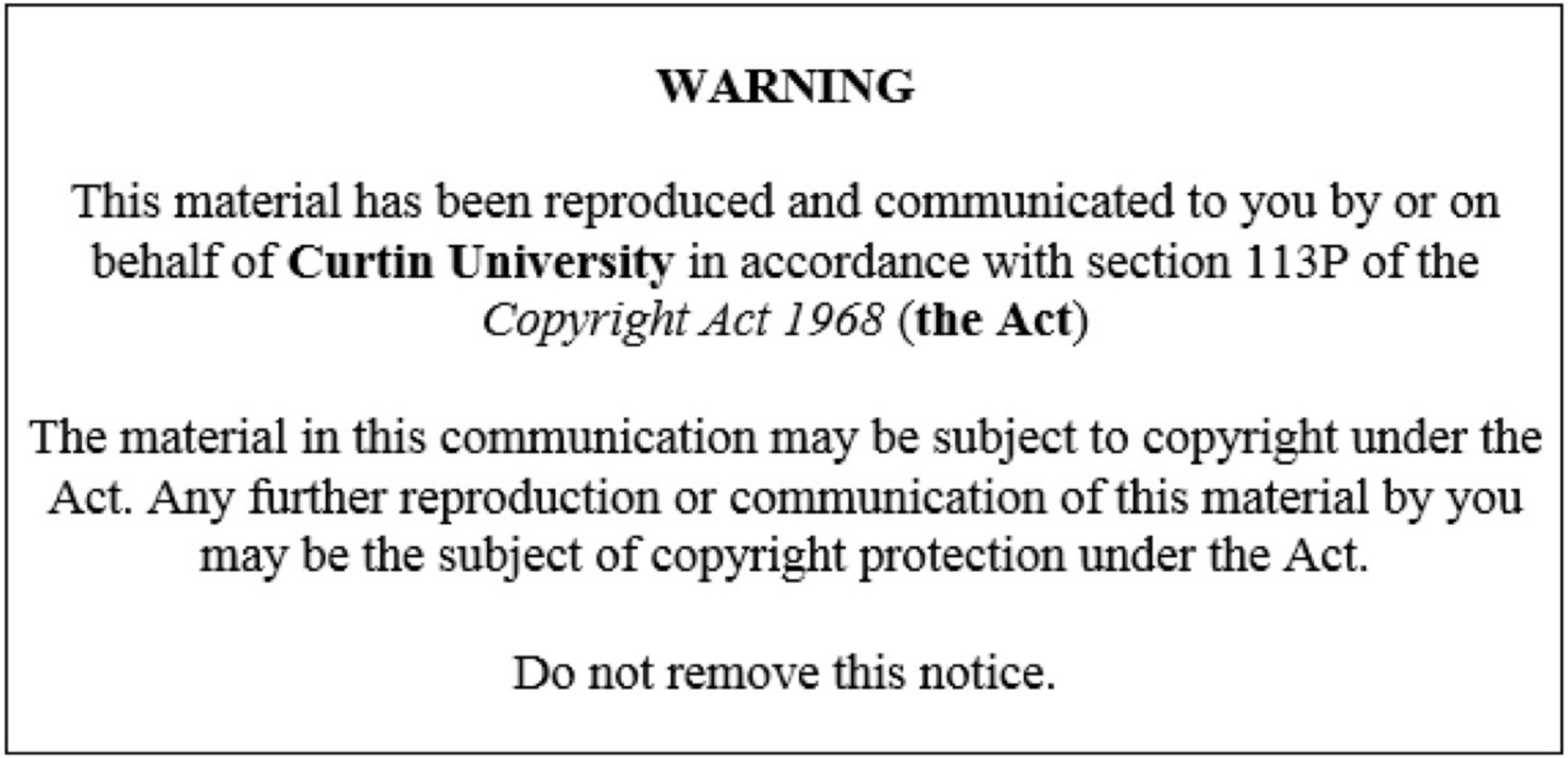
Curtin University

Using the Web

ISYS5002, School of Marketing and Management

ELECTRONIC WARNING NOTICE FOR COPYRIGHT STATUTORY LICENCES



I acknowledge the traditional custodians of the land on which I work and live, and recognise their continuing connection to land, water and community. I pay respect to elders past, present and emerging.

Today

- Accessing/Download Data

  3 methods in Python

- Create Webpage
- Web Scraping

urllib.request.urlretrieve

```
from urllib import request

# Define the remote file to retrieve
remote_url = 'https://www.google.com/robots.txt’

# Define the local filename to save data
local_file = 'local_copy.txt’

# Download remote and save locally
request.urlretrieve(remote_url, local_file)
```

requests.get + manual save

```
import requests
# Define the remote file to retrieve remote_url =
'https://www.google.com/robots.txt’

# Define the local filename to save data
local_file = 'local_copy.txt’

# Make http request for remote file data
data = requests.get(remote_url)

# Save file data to local copy
with open(local_file, 'wb')as file:
   file.write(data.content)
```

wget.download

```
import wget
# Define the remote file to retrieve
remote_url = 'https://www.google.com/robots.txt’

# Define the local filename to save data
local_file = 'local_copy.txt’

# Make http request for remote file data
wget.download(remote_url, local_file)
```

Website Development

- Django

  Fast, secure

- Flask

  minimalistic

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
   return "Hello, World!"

if __name__ == "__main__":
    app.run(debug=True)
```

What is Web Scraping

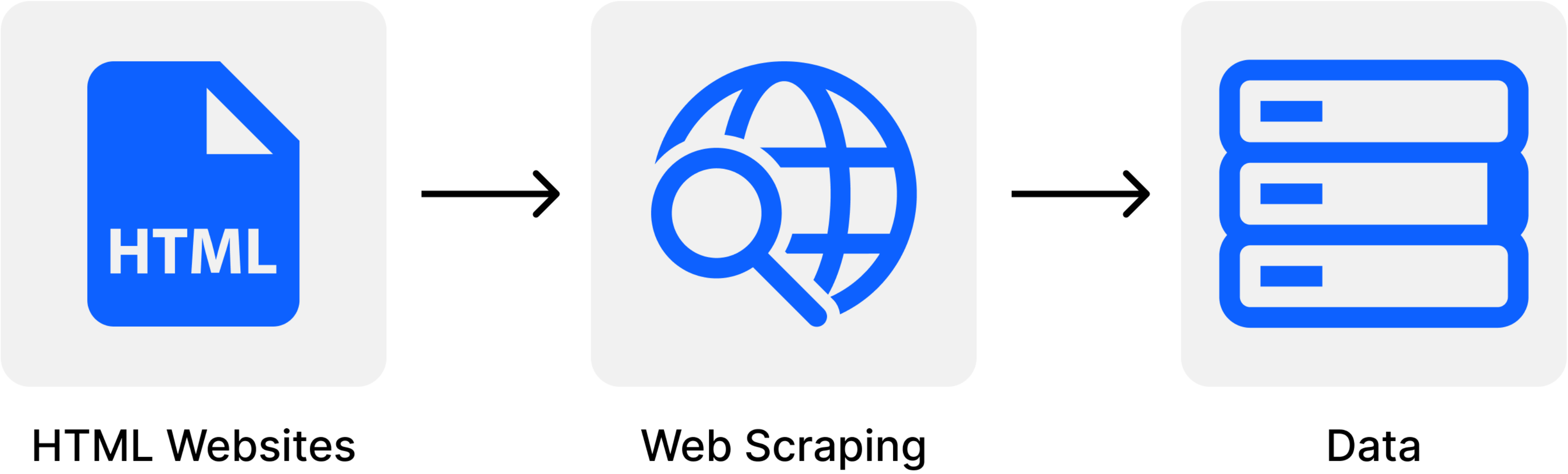


Image Source: https://i.imgur.com/6zM7JBq.png

Why is Web Scraping Used?

- Price Comparison
- Email address gathering
- Social Media Scraping
- Research and Development
- Job Listings

Is Web Scraping Legal?

- Copyright law protects original work
    - Fair dealing
- Data not consider original work
- Read Website Terms of Service
- Check robots.txt

Is Python Good for Web Scraping

- Ease of Use
- Large Collection of Libraries
- Dynamically Typed
- Small code large Task
- Community

How Scrape

- Find URL
- Inspect the Page
- Find the data want to extract
- Write the code
- Run code and extract the data
- Store data in the required format

Libraries

- Selenium – automate browser activities
- BeautifulSoup – parsing HTML and XML documents
- Requests – make HTTP request simpler

Anatomy of URL

- Scheme

  https://developer.mozilla.org

- Domain

  https://**developer.mozilla.org**/en-US/docs/Learn/

- Path

  https://developer.mozilla.org**/en-US/docs/Learn/**

- Parameters

  https://developer.mozilla.org/en-US/**search?q=URL**

HTML Tags

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Can you

- Explain Download Data
- Understand Create Webpage
- Describe Web Scraping