CM2015 Programming With Data

Python Programming Summary

September 16 - 2023

1. Functions

Functions in Python are blocks of organized and reusable code that perform a single, related action. They are crucial for breaking down complex tasks into smaller, manageable sub-tasks.

In [2]:
```python
# Example of defining and calling a function
def function_name(parameters):
    # body of the function
    return value

# result = function_name(arguments)
```

Key Points:

- Functions help modularize and reuse code.
- Defined using the `def` keyword.
- Accept parameters and can return values.
- Should be invoked/called to execute. *result = function_name(arguments)*

2. Data Types

In [3]:
```python
# Strings: Sequence of characters
my_str = "Hello, World!"

# Lists: Ordered collections of items (mutable)
my_list = [1, 2, 3, "Python", [4, 5]]

# Dictionaries: Unordered key-value pairs
my_dict = {"key1": "value1", "key2": "value2"}
```

Key Points:

- Strings represent text data.
- Lists are ordered collections, can have mixed types.
- Dictionaries store key-value pairs.
- Both lists and dictionaries are mutable.

3. Loops

In [4]:
```python
# For Loop example
for item in my_list:
    print(item)

# While Loop example
count = 0
while count < 3:
    print(count)
    count += 1
```

```
1
2
3
Python
[4, 5]
0
1
2
```

Key Points:

- `For` loops iterate over sequences (like lists or strings).
- `While` loops execute as long as a condition is true.
- Loops can use `break` to exit early, `continue` to skip an iteration.

4. File Handling

In [ ]:
```python
# Reading from a file
with open('filename.txt', 'r') as file:
    content = file.read()

# Writing to a file
with open('filename.txt', 'w') as file:
    file.write("Hello, Python!")
```

Key Points:

- Use the `open()` function to read or write to files.
- `with` statement ensures file closure after operations.
- Two common modes: `'r'` for reading, `'w'` for writing.

5. Conditions

In [5]:
```python
condition1, condition2 = True, False

# Example of conditions
if condition1:
    print("Condition1 is true")
elif condition2:
    print("Condition2 is true")
else:
    print("Both conditions are false")
```

```
Condition1 is true
```

Key Points:

- Use `if`, `elif`, and `else` for conditional logic.
- Python supports comparison (`<`, `>`, `==`, `!=`, `<=`, `>=`) and logical (`and`, `or`, `not`) operators.

6. User Input

In [7]:
```
# Uncomment below lines to test user input
user_input = input("Enter something: ")
print(f"You entered: {user_input}")
```

```
Enter something: Hello World!
You entered: Hello World!
```

Key Points:

- `input()` function retrieves user input as a string.
- Can cast to other data types like int or float if needed.

7. Error Handling

In [11]:
```
try:
    numerator = 10
    denominator = 0
    result = numerator / denominator
    print(result)
except ZeroDivisionError:
    print("Can't divide by zero!")
except Exception as e:
    print(f"An error occurred: {e}")
finally:
    print("Execution finished!")
```

```
Can't divide by zero!
Execution finished!
```

Key Points:

- `try`, `except`, and `finally` are used for error handling.
- Catch specific errors or use a general `Exception`.
- `finally` block executes no matter what, useful for cleanup.

DataFrames (Pandas)

> Central to Pandas is the DataFrame — a two-dimensional, size-mutable, heterogeneous tabular data structure.
>
> Think of it like an Excel spreadsheet or SQL table in memory.

1. DataFrames

Creation

- There are many ways to construct a DataFrame. It can be created from a dictionary, lists, external files, or even another DataFrame.

In [14]:
```
import pandas as pd

# Read CSV file into a DataFrame
#UNCOMMENT to load from local file
# df = pd.read_csv('path_to_file.csv')

# Example of creating a DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'San Francisco', 'Los Angeles']
}

df = pd.DataFrame(data)
print(df)
```

```
      Name  Age           City
0    Alice   25       New York
1      Bob   30  San Francisco
2  Charlie   35    Los Angeles
```

Manipulation

- One of the reasons for the popularity of DataFrames is the ease of data manipulation. You can easily add, delete or modify the data in a DataFrame.

In [15]:
```
# Add a new column
df['Salary'] = [50000, 60000, 70000]
print(df)

# Delete a column
# del df['Age']
# print(df)
```

```
      Name  Age           City  Salary
0    Alice   25       New York   50000
1      Bob   30  San Francisco   60000
2  Charlie   35    Los Angeles   70000
```

Key Points:

- DataFrames store data in tabular form (rows & columns).
- Columns can be of different types.
- Operations like filtering, aggregation, and transformation can be performed.

2. Handling Missing Data

> Real-world data is often messy and contains missing values. Pandas provides tools to handle such scenarios.

In [16]:
```
# Example DataFrame with missing values
data_with_nan = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [25, None, 35, 40],
    'City': ['New York', 'San Francisco', None, 'Chicago'],
    'Salary':[50000,60000,70000,None]
}

df_nan = pd.DataFrame(data_with_nan)

# Handling missing values
df_dropped = df_nan.dropna()  # Removes rows with NaN
df_filled = df_nan.fillna("Unknown")  # Fills NaN with "Unknown"
print(df_dropped)
print("\n")
print(df_filled)
```

```
    Name   Age      City   Salary
0  Alice  25.0  New York  50000.0


      Name      Age           City   Salary
0    Alice     25.0       New York  50000.0
1      Bob  Unknown  San Francisco  60000.0
2  Charlie     35.0        Unknown  70000.0
3    David     40.0        Chicago  Unknown
```

Key Points:

- Real-world data often has missing values.
- `dropna()` removes rows/columns with missing values.
- `fillna()` replaces missing values with specified values.

3. Data Selection

> Select specific rows and columns based on criteria or location.

In [17]:
```
# Selecting data using .iloc and .loc
print(df.iloc[0])      # First row
print("\n")
print(df.loc[0, 'Name']) # Value at first row, Name column
```

```
Name         Alice
Age             25
City      New York
Salary       50000
Name: 0, dtype: object


Alice
```

Key Points:

- `.iloc` is purely integer-based location indexing.
- `.loc` is label-based indexing.

4. Aggregations

> Aggregate data for analysis (e.g., sum, average).

In [18]:
```
# Aggregations on the DataFrame
print(df['Age'].mean())  # Average age
print(df['Age'].sum())   # Total age
```

```
30.0
90
```

Key Points:

- Aggregations provide summarized data.
- Common functions: `sum()`, `mean()`, `min()`, `max()`, etc.

5. Basic Stats

> Gain insights into data distributions, tendencies, and other statistical measures.

In [19]:
```
# Basic statistics on the DataFrame
print(df['Age'].describe())  # Summary stats for Age column
print("\n")
print(df["Name"].describe())
print("\n")
print(df["City"].describe())
```

```
count     3.0
mean     30.0
std       5.0
min      25.0
25%      27.5
50%      30.0
75%      32.5
max      35.0
Name: Age, dtype: float64


count          3
unique         3
top        Alice
freq           1
Name: Name, dtype: object


count             3
unique            3
top       New York
freq              1
Name: City, dtype: object
```

Key Points:

- `describe()` gives count, mean, std deviation, min, 25th percentile, median, 75th percentile, and max.
- Useful for initial exploratory analysis.

Regular Expressions

1. Definition and Purpose of Metacharacters

- Metacharacters are special characters in regular expressions that have a specific meaning and are not treated as literals.
- Common metacharacters include: . ^ $ * + ? { } [ ] \ | ( )

Key Points:

- Metacharacters give regular expressions their power and flexibility.
- They allow for more advanced and specific pattern matching.

2. Basic Pattern Matching Using Python's re Module

In [21]:
```python
import re

pattern = r"\d{3}-\d{2}-\d{4}"  # Matches Social Security Number format
result = re.search(pattern, "Her SSN is 123-45-6789.")

if result:
    print("Match found:", result.group())
else:
    print("Match not found.")
```

```
Match found: 123-45-6789
```

Key Points:

- Use the `re` module for regex operations in Python.
- `re.search()` searches a string for a match, and returns a `Match` object if found.

3. E-Mail Pattern Matching

In [22]:
```python
import re

# Define the email pattern
email_pattern = r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"

email = "example.email+filter@domain.com"
```

Here's a breakdown of the pattern:

- ### ^: Start of the string.
- ### [a-zA-Z0-9._%+-]+: Matches the user part of the email. It can include letters (both lower and upper case), numbers, dots, underscores, percentages, pluses, and hyphens.
- ### @: Literal match for the @ symbol.
- ### [a-zA-Z0-9.-]+: Matches the domain part of the email before the dot. It can include letters (both lower and upper case), numbers, dots, and hyphens.
- ### .: Literal match for the dot (.) symbol.
- ### [a-zA-Z]{2,}: Matches the top-level domain (like com, org, net). It consists of at least two letters.
- ### $: End of the string.

In [23]:
```
result = re.match(email_pattern, email)

if result:
    print("Valid email!")
else:
    print("Invalid email!")
```

```
Valid email!
```

> We use the match function from the re module to check if the entire email string conforms to the pattern. If it does, match returns a match object; otherwise, it returns None.

Web Scraping

1. Basics

- Web scraping is the process of extracting data from web pages.
- It involves parsing the HTML content of the page.

In [24]:
```
# This is a basic example using Python's BeautifulSoup library.

from bs4 import BeautifulSoup
import requests

url = "https://www.apple.com"
# response = requests.get(url)
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')

title = soup.title.string
print(title)
```

```
Apple
```

In [25]:
```
print(soup.getText().splitlines())
```

```
['', '', '', '', '', '', '', '', '', '', '', '', '', 'Apple', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', 'Apple', '',
'AppleStoreMaciPadiPhoneWatchVisionAirPodsTV & HomeEntertainmentAccessori
esSupport', '', '', '0+', '', '', '', '', '', '', '', '', '', '\t\t\t\t\t
\t\t\t\tGet $200–$650 in credit toward iPhone\xa015 or\xa0iPhone\xa015\xa
0Pro when you trade in iPhone\xa011 or\xa0higher.1 See\xa0what\xa0your\xa
0device\xa0is\xa0worth', '', '', '', '', '', '', '', '', '', '', '\xa0',
'', 'iPhone 15 Pro', 'Titanium. So strong. So light. So Pro.', 'Available
\xa0starting\xa09.22', '', 'Learn more', 'Pre-order', '', '', '', '', '',
'', '', '', '', '', '', '', '\xa0', '', 'iPhone 15', 'New camera. New des
ign. Newphoria.', 'Available\xa0starting\xa09.22', '', 'Learn more', 'Pre
-order', '', '', '', '', '', '', '', '', '', '', '', '', '\xa0', '', 'App
le Watch Series 9', 'Smarter. Brighter. Mightier.', 'Available starting 9
.22', '', 'Learn more', 'Order now', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '\xa0', '', 'Apple Watch Ultra 2', 'Next level advent
ure.', 'Available starting 9.22', '', 'Learn more', 'Order now', '', '',
'', '', '', '', '', '', '', '', '', '', '\xa0', '', 'Carbon Neutral', 'A
first for Apple\xa0Watch. And Apple.', '', 'Learn more', 'Watch the film'
, '', '', '', '', '', '', '', '', '', '', '', '', '\xa0', '', 'Apple Watc
```

h SE', 'A great deal to love.', 'Available starting 9.22', '', 'Learn mor
e', 'Order now', '', '', '', '', '', '', '', '', '', '', '', '', '', '\xa0',
'\t\t\t\t\t\t\t\t\t\t', '', 'Save on Mac or iPad for\xa0college.', '', '\t\
t\t\t\t\t\t\t\t\t\tPlus get a gift card up to $150.2', '', '', 'Shop now'
, '', '', '', '', '', '\t\t\t\t\t\t\t\t\t\t\tAn image of a MacBook Air,
iPad, and Apple Store Gift Card walking.', '\t\t\t\t\t\t\t\t\t\t\t', '',
'\t\t\t\t\t\t\t\t\t\t\t\tAn animation of a MacBook Air, iPad, and Apple S
tore Gift Card walking.', '\t\t\t\t\t\t\t\t\t\t\t', '', '', '', '', '
', '', '', '', '', '', '', '', '', '', '', '', '\xa0', '', 'AirPods Pro',
'Adaptive Audio. Now playing.', 'Available starting 9.22', '', 'Learn mor
e', 'Order now', '', '', '', '', '', '', '', '', '', '', '', '', '', '\xa0',
'', '', 'Apple TV plus', '', '', 'The Morning Show', '', 'Start streaming
Season 3 now', '', '', '', '', '', '', '', '', '', '', '', '', '', '\xa0'
, '', 'Apple Card', 'Get up to 3% Daily\xa0Cash back with every purchase.
', '', 'Learn more', 'Apply now', 'Apply now', '', '', '', '', '', '', ''
, '', '', '', '', '', '', '', '', 'Apple TV+', '', '', '', '', '', '', ''
, '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', 'Apple Footer', '', '', '1. Trade-in valu
es will vary based on the condition, year, and configuration of your elig
ible trade-in device. Not all devices are eligible for credit. You must b
e at least 18 years old to be eligible to trade in for credit or for an A
pple\xa0Gift\xa0Card. Trade-in value may be applied toward qualifying new
device purchase, or added to an Apple\xa0Gift\xa0Card. Actual value award
ed is based on receipt of a qualifying device matching the description pr
ovided when estimate was made. Sales tax may be assessed on full value of
a new device purchase. In-store trade-in requires presentation of a valid
photo ID (local law may require saving this information). Offer may not b
e available in all stores, and may vary between in-store and online trade
-in. Some stores may have additional requirements. Apple or its trade-in
partners reserve the right to refuse or limit quantity of any trade-in tr
ansaction for any reason. More details are available from Apple’s trade-i
n partner for trade-in and recycling of eligible devices. Restrictions an
d limitations may apply.', '2. Qualified\xa0Purchasers receive an Apple\x
a0Gift\xa0Card when they purchase an eligible Mac or iPad at a Qualifying
\xa0Location. Only one Apple\xa0Gift\xa0Card per eligible Mac or iPad per
Qualified\xa0Purchaser. Offer subject to availability. While supplies las
t. View full terms and conditions of offer here.', '', 'To access and use
all Apple\xa0Card features and products available only to Apple\xa0Card u
sers, you must add Apple\xa0Card to Wallet on an iPhone or iPad that supp
orts and has the latest version of iOS or iPadOS. Apple\xa0Card is subjec
t to credit approval, available only for qualifying applicants in the Uni
ted States, and issued by Goldman Sachs Bank USA, Salt Lake City Branch.
', 'If you reside in the U.S. territories, please call Goldman Sachs at 8
77-255-5923 with questions about Apple\xa0Card.', 'Learn more about how A
pple\xa0Card applications are evaluated at support.apple.com/kb/HT209218.
', 'A subscription is required for Apple\xa0TV+.', '', 'Major League Base
ball trademarks and copyrights are used with permission of MLB Advanced M
edia, L.P. All rights reserved.', '', '', '', '', '', '', '', '', 'Shop a
nd Learn', '', 'Shop and Learn', '+', '', '', '', 'Store', 'Mac', 'iPad',
'iPhone', 'Watch', 'Vision', 'AirPods', 'TV & Home', 'AirTag', 'Accessori
es', 'Gift Cards', '', '', '', '', 'Apple Wallet', '', 'Apple Wallet', '+
', '', '', '', 'Wallet', 'Apple\xa0Card', 'Apple\xa0Pay', 'Apple\xa0Cash'
, '', '', '', '', '', '', 'Account', '', 'Account', '+', '', '', '', 'Man
age Your Apple\xa0ID', 'Apple Store Account', 'iCloud.com', '', '', '', '
', 'Entertainment', '', 'Entertainment', '+', '', '', '', 'Apple\xa0One',
'Apple\xa0TV+', 'Apple\xa0Music', 'Apple\xa0Arcade', 'Apple\xa0Fitness+',
'Apple\xa0News+', 'Apple Podcasts', 'Apple\xa0Books', 'App\xa0Store', '',
'', '', '', '', '', 'Apple Store', '', 'Apple Store', '+', '', '', '', 'F
ind a Store', 'Genius Bar', 'Today at Apple', 'Apple Camp', 'Apple Store
App', 'Certified Refurbished', 'Apple\xa0Trade\xa0In', 'Financing', 'Carr
ier Deals at Apple', 'Order Status', 'Shopping Help', '', '', '', '', '',

```
'', 'For Business', '', 'For Business', '+', '', '', '', 'Apple and Busin
ess', 'Shop for Business', '', '', '', '', 'For Education', '', 'For Educ
ation', '+', '', '', '', 'Apple and Education', 'Shop for K-12', 'Shop fo
r College', '', '', '', '', 'For Healthcare', '', 'For Healthcare', '+',
'', '', '', 'Apple in Healthcare', 'Health on Apple\xa0Watch', 'Health Re
cords on iPhone', '', '', '', '', 'For Government', '', 'For Government',
'+', '', '', '', 'Shop for Government', 'Shop for Veterans and Military',
'', '', '', '', '', '', 'Apple Values', '', 'Apple Values', '+', '', '',
'', 'Accessibility', 'Education', 'Environment', 'Inclusion and Diversity
', 'Privacy', 'Racial Equity and Justice', 'Supplier Responsibility', '',
'', '', '', '', 'About Apple', '', 'About Apple', '+', '', '', '', 'Newsroom'
, 'Apple Leadership', 'Career Opportunities', 'Investors', 'Ethics & Comp
liance', 'Events', 'Contact Apple', '', '', '', '', '', '', '', '\t\t\tMo
re ways to shop: Find an Apple Store or other retailer near you. Or call
1-800-MY-APPLE.', '', '', 'United States', '', '', 'Copyright ©', '\t\t\t
\t', '\t\t\t\t2023', '\t\t\t\t Apple Inc. All rights reserved.', '\t\t\t'
, '', '', 'Privacy Policy', '', '', 'Terms of Use', '', '', 'Sales and Re
funds', '', '', 'Legal', '', '', 'Site Map', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '', '']
```

Key Points:

- Web scraping uses HTTP requests to fetch web pages.
- Libraries like BeautifulSoup can parse and navigate HTML content.

2. Ethics

- Not all websites appreciate or allow scraping. Respect robots.txt and terms of service.
- Overloading a server with rapid, frequent requests is unethical and can be illegal.

Key Points:

- Always check `robots.txt` of a website before scraping.
- Avoid causing harm or inconvenience. Rate-limit your requests.

In [26]:
```python
url = "https://www.apple.com/robots.txt"
response = requests.get(url)


# Ensure that we successfully fetched the data
if response.status_code == 200:
    print(response.text)
else:
    print(f"Failed to retrieve the content. HTTP Status Code: {response.s
```

```
# robots.txt for http://www.apple.com/

User-agent: *
Disallow: /*/includes/*
Disallow: /*retail/availability*
Disallow: /*retail/availabilitySearch*
Disallow: /*retail/pickupEligibility*
Disallow: /*shop/signed_in_account*
Disallow: /*shop/sign_in*
Disallow: /*shop/sign_out*
```

```
Disallow: /*shop/answer/vote*
Disallow: /*shop/bag*
Disallow: /*shop/browse/overlay/*
Disallow: /*shop/browse/ribbon/*
Disallow: /*shop/browse/campaigns/mobile_overlay*
Disallow: /*shop/button_availability*
Disallow: /*shop/favorites*
Disallow: /*shop/iphone/payments/overlay/*
Disallow: /*shop/mobile/olss_error*
Disallow: /*shop/mobilex/*
Disallow: /*shop/np/order*
Disallow: /*shop/np/giftorguestorder*
Disallow: /*shop/np/sign_in*
Disallow: /*shop/order/*
Disallow: /*shop/rs-mvt/rel/*
Disallow: /*shop/sentry*
Disallow: /*shop/store/feeds/*
Disallow: /*shop/variationSelection
Disallow: /*_adc_*/shop/
Disallow: /*_aoc_*/shop/
Disallow: /*_enterprise*/shop/
Disallow: /*_internal-epp-discounted*/shop/
Disallow: /*_k12nonbts*/shop/
Disallow: /*_kiosk*/shop/
Disallow: /*_nonbts*/shop/
Disallow: /*_qpromo*/shop/
Disallow: /*_refurb-discounted*/shop/
Disallow: /cn/*/aow/*
Disallow: /go/awards/*
Disallow: /newsroom/notifications/*
Disallow: /tmall*
Allow: /ac/globalnav/2.0/*/images/ac-globalnav/globalnav/search/*

User-agent: Baiduspider
Disallow: /*
Allow: /cn/*
Allow: /cn-edu/*
Allow: /cn-k12/*

User-agent: HaoSouSpider
Disallow: */product-red/
Disallow: /*/retail/availability*
Disallow: /*/retail/availabilitySearch*
Disallow: /*/retail/pickupEligibility*
Disallow: /*/shop/account/setup*
Disallow: /*/shop/answer/vote*
Disallow: /*/shop/browse/campaigns/mobile_overlay*
Disallow: /*/shop/button_availability*
Disallow: /*/shop/bag*
Disallow: /*/shop/change_password*
Disallow: /*/shop/checkout*
Disallow: /*/shop/create_account*
Disallow: /*/shop/favorites*
Disallow: /*/shop/identify_user*
Disallow: /*/shop/mobile/checkout/start*
Disallow: /*/shop/mobilex/*
Disallow: /*shop/np/order*
Disallow: /*shop/np/giftorguestorder*
Disallow: /*shop/np/sign_in*
Disallow: /*/shop/rs-mvt/rel/*
Disallow: /*/shop/sentry*
```

```
Disallow: /*/shop/sentryx/change_password*
Disallow: /*/shop/sentryx/create_account*
Disallow: /*/shop/sentryx/create_account_confirm*
Disallow: /*/shop/sentryx/identify_user*
Disallow: /*/shop/sentryx/sign_in*
Disallow: /*/shop/signed_in_account*
Disallow: /*/shop/sign_in*
Disallow: /*/shop/sign_out*
Disallow: /*/shop/storeConfig*
Disallow: /*/shop/variationSelection*
Disallow: /*/shop/vieworder*
Disallow: /apple-watch-nike/
Disallow: /apple-watch-hermes/
Disallow: /cn/*/aow/*
Disallow: /newsroom/notifications/*
Disallow: /retail/availability*
Disallow: /retail/availabilitySearch*
Disallow: /retail/pickupEligibility*
Disallow: /shop/bag*
Disallow: /tmall/*
Disallow: /cn_cmb*
Disallow: /cn_abc*
Disallow: /cn_icbc*
Disallow: /cn_ccb*

User-agent: Sogou web spider
Disallow: /*
Allow: /cn/*
Allow: /cn-k12/*

User-agent: Sogou inst spider
Disallow: /*
Allow: /cn/*
Allow: /cn-k12/*

User-agent: Sogou spider2
Disallow: /*
Allow: /cn/*
Allow: /cn-k12/*

#DaumWebMasterTool:fe46641ef2e4f3f25544ad9d70c6029df24dd184fad54154abaa3c
263cf5a09a:h7Tb+WCGBcuKBnKRAHQGEQ==

Sitemap: https://www.apple.com/shop/sitemap.xml
Sitemap: https://www.apple.com/autopush/robots/compare-sitemap.xml
Sitemap: https://www.apple.com/autopush/sitemap/sitemap-index.xml
Sitemap: https://www.apple.com/newsroom/sitemap.xml
Sitemap: https://www.apple.com/retail/sitemap/sitemap.xml
Sitemap: https://www.apple.com/today/sitemap.xml
```

The robots.txt you primarily tells the Sogou inst spider and Sogou spider2 to avoid crawling most parts of the website, except for certain sections meant for Chinese audiences (/cn/ and /cn-k12/). The file also indicates the presence of several sitemaps which provide structured lists of URLs for web crawlers to access.

> Website scraped: www.apple.com
>
> Date:16/SEP/2023

3. Dynamic Websites

- Many modern websites load content dynamically using JavaScript.
- Traditional scraping tools can't capture this dynamic content. Tools like Selenium can automate browsers to capture such content.

In [ ]:
```
# pip install selenium
```

In [ ]:
```
# pip install webdriver_manager
```

In [11]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By  # Importing By
from webdriver_manager.chrome import ChromeDriverManager
import time

# Automatically download and use the latest ChromeDriver
driver = webdriver.Chrome()

# Navigate to the website
driver.get('http://quotes.toscrape.com/js/')

# Give the JavaScript some time to load the dynamic content
time.sleep(2)  # 2 seconds delay

# Extract quotes from the website using the new method
quotes = driver.find_elements(By.CSS_SELECTOR, '.quote .text')

for quote in quotes:
    print(quote.text)

# Close the browser
driver.quit()
```

```
“The world as we have created it is a process of our thinking. It cannot
be changed without changing our thinking.”
“It is our choices, Harry, that show what we truly are, far more than our
abilities.”
“There are only two ways to live your life. One is as though nothing is a
miracle. The other is as though everything is a miracle.”
“The person, be it gentleman or lady, who has not pleasure in a good nove
l, must be intolerably stupid.”
“Imperfection is beauty, madness is genius and it's better to be absolute
ly ridiculous than absolutely boring.”
“Try not to become a man of success. Rather become a man of value.”
“It is better to be hated for what you are than to be loved for what you
are not.”
“I have not failed. I've just found 10,000 ways that won't work.”
“A woman is like a tea bag; you never know how strong it is until it's in
hot water.”
“A day without sunshine is like, you know, night.”
```

> Website scraped: http://quotes.toscrape.com/js/
>
> Date:16/SEP/2023

Key Points:

- Dynamic content is loaded on-the-fly, often after the initial page load.
- Selenium automates a browser, allowing capture of dynamic content.

SQL

- SQL is a domain-specific language designed to manage and query data held in relational databases.
- Users can perform tasks like adding, retrieving, and updating data.

1. Basic Commands

CREATE

> CREATE TABLE table_name (column1 datatype, column2 datatype, ...);

INSERT

> INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);

SELECT

> SELECT column1, column2, ... FROM table_name;
>
> SELECT column1, column2, ... FROM table_name WHERE condition;

UPDATE

> UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

DELETE

> DELETE FROM table_name WHERE condition;

DROP

> DROP TABLE table_name;

Key Points:

- `CREATE` Creates a new table, view, or other database objects.
- `INSERT` Adds new records into a table.
- `SELECT` Retrieves data from a database.
- `WHERE` Filters records based on one or more condition.
- `UPDATE` Modifies existing records in a table.
- `DELETE` Removes records from a table.
- `DROP` Deletes an existing table in a database.

2. Advanced SQL Commands

JOIN

Types include

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN.

Key Points:

- `JOIN` Combines rows from two or more tables based on related columns.
- `GROUP BY` Groups rows with the same values in specified columns.
- `HAVING` Filters the result of a GROUP BY operation.
- `ORDER BY` Sorts the result set based on specified columns.
- `ALTER` Modifies an existing table, such as adding or deleting columns.

3. SQL Injection Attacks

- SQL Injection is a technique where attackers can insert malicious SQL code into a query. It capitalizes on inadequate input validation in applications, leading to unauthorized data access or corruption.

Prevention:

1. Sanitize Inputs: Ensure all user inputs are sanitized before they're processed.

2. Parameterized Queries: Use parameterized queries or prepared statements to separate SQL logic and data, eliminating the risk of malicious data altering the query structure.

3. Least Privilege Principle: Give the minimum required permissions to the database accounts. If a user doesn’t need to drop tables, they shouldn’t have that permission.

4. Regular Audits: Periodically review and audit your code and databases for vulnerabilities.

Testing & Version Control

1. Unit Testing

- Unit testing involves testing individual units or components of a software in isolation.
- The primary aim is to ensure each unit functions correctly.

In [39]:
```python
# Basic example using Python's built-in unittest module.
import unittest

unittest.main(argv=['first-arg-is-ignored'], exit=False)

def add(a, b):
    return a + b

class TestAdditionFunction(unittest.TestCase):
    def test_add(self):
        self.assertEqual(add(2, 3), 5)

if __name__ == "__main__":
    unittest.main(argv=['first-arg-is-ignored'], exit=False)
```

```
----------------------------------------------------------------------
Ran 0 tests in 0.000s

OK
.
----------------------------------------------------------------------
Ran 1 test in 0.001s

OK
```

Key Points:

- Unit tests check the correctness of individual functions or methods.
- They should be isolated from external factors like databases or network services.

2. Git (Version Control)

- Git is a distributed version control system.
- It tracks changes in source code, allowing multiple people to collaborate efficiently.

In [ ]:
```python
# Sample Git Commands
git init      # Initialize a new git repository
git log       # View commit logs
git commit -m "Initial commit" # Commit changes with a message
```

Key Points:

- Version control systems, like Git, track and manage changes in code.
- Distributed systems (e.g., Git) allow every user to have a full copy of the repository, while centralized systems have one central repository.
- Basic git commands: `git init`, `git log`, `git commit`, etc.

Data Visualization

Data visualization is the graphical representation of data. By visualizing data, one can recognize patterns, trends, and correlations that might go unnoticed in text-based data.

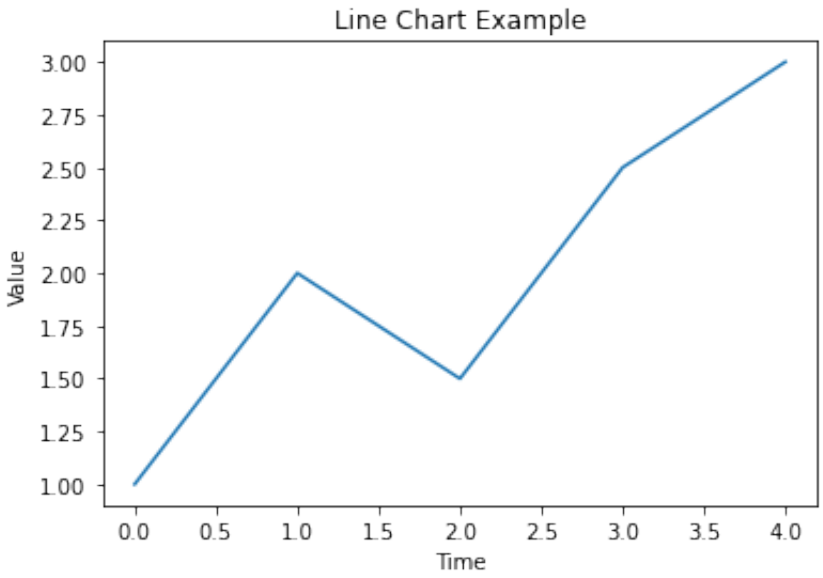
Different Types of Visualizations and Their Use-Cases

- There are numerous types of visualizations, each suitable for representing different kinds of data. Let's briefly touch upon a few:

Line Charts: Ideal for showing trends over time.

> plt.plot(xValues[ ], yValues[ ])

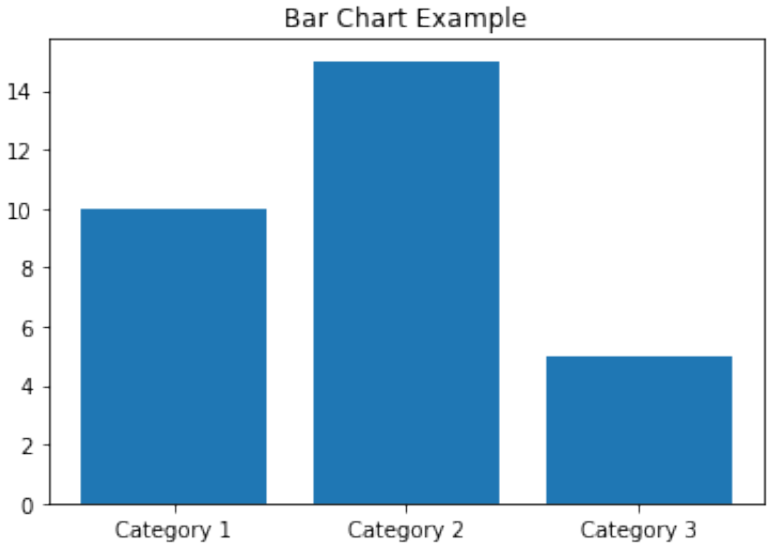
In [4]:
```
# Code example for Line Chart using Python's Matplotlib
import matplotlib.pyplot as plt

time = [0, 1, 2, 3, 4]
values = [1, 2, 1.5, 2.5, 3]
plt.plot(time, values)
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Line Chart Example')
plt.show()
```



Bar Charts: Useful for comparing quantities across categories.

> plt.bar(xValues[ ], yValues[ ])

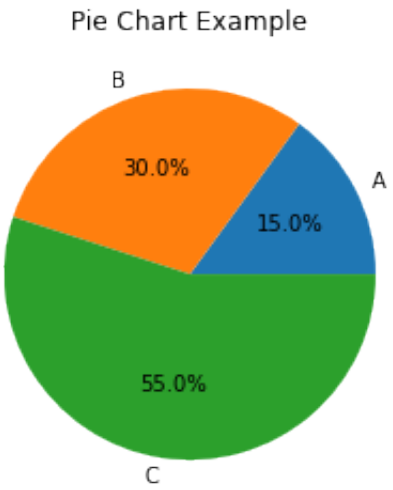
In [5]:
```
# Code example for Bar Chart
categories = ['Category 1', 'Category 2', 'Category 3']
values = [10, 15, 5]
plt.bar(categories, values)
plt.title('Bar Chart Example')
plt.show()
```



Pie Charts: Great for showing proportional data, representing percentages of a whole.

> plt.pie(Values[ ], *labels*=Names[ ],..*optionals*)

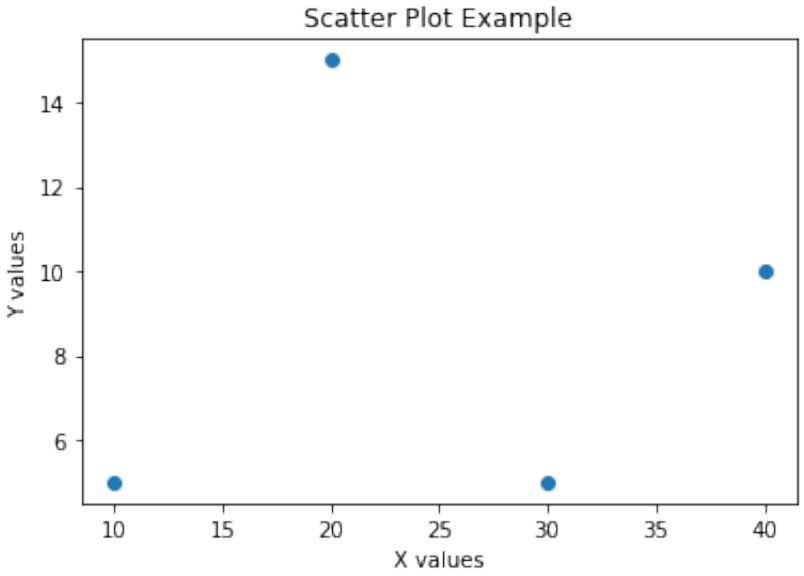
In [6]:
```
# Code example for Pie Chart
labels = ['A', 'B', 'C']
sizes = [15, 30, 55]
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title('Pie Chart Example')
plt.show()
```



Scatter Plots: Useful for showing the relationship between two continuous variables.

> plt.scatter(xValues[ ], yValues[ ])

In [7]:
```
# Code example for Scatter Plot
x = [10, 20, 30, 40]
y = [5, 15, 5, 10]
plt.scatter(x, y)
plt.xlabel('X values')
plt.ylabel('Y values')
plt.title('Scatter Plot Example')
plt.show()
```



When 2D Might Be Preferable to 3D

2D visualizations are often preferable to 3D for several reasons:

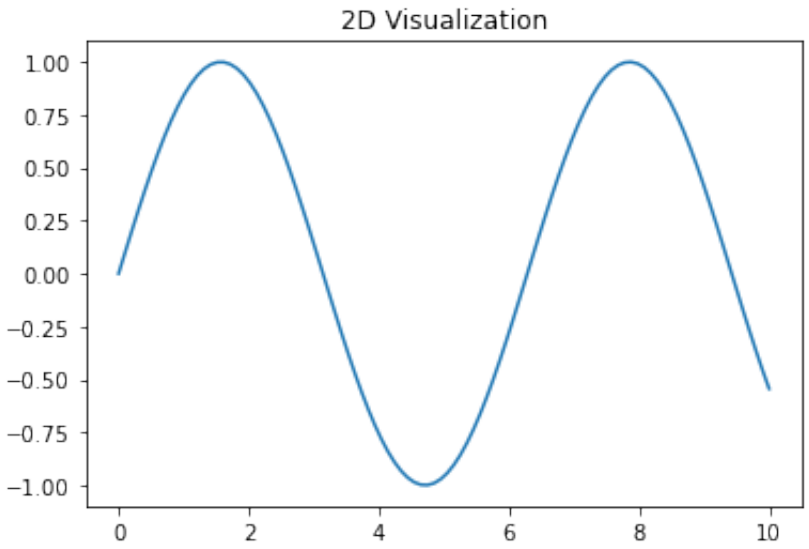
1. Clarity: 2D visualizations tend to be more straightforward and easier to interpret.

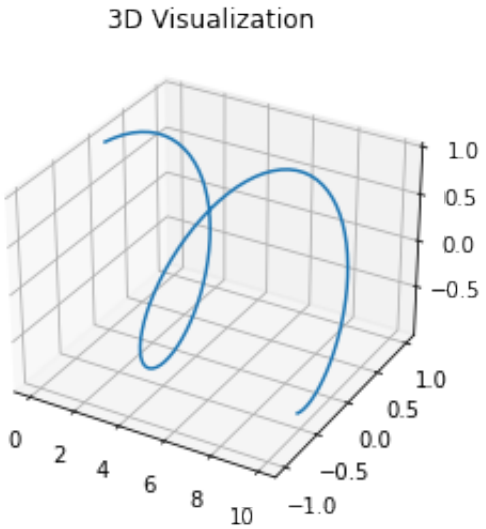
2. Distortion:3D can sometimes distort data, giving a misleading representation.

3.Usability: 2D charts and graphs are generally more mobile-friendly and accessible.

4.Overcomplication: In many cases, adding a third dimension doesn't provide additional clarity and instead makes the graph more difficult to understand.

In [8]:
```
# Compare 2D and 3D visualization using Matplotlib

import numpy as np

# 2D visualization
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.title('2D Visualization')
plt.show()

# 3D visualization
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = np.linspace(0, 10, 100)
y = np.sin(x)
z = np.cos(x)
ax.plot(x, y, z)
ax.set_title('3D Visualization')
plt.show()
```





Key Points:

- Data visualization helps in understanding complex data by representing it graphically.
- Line charts are for trends over time, bar charts for comparing quantities, pie charts show proportions, and scatter plots indicate relationships between variables.
- 2D visualizations are often clearer and more user-friendly than 3D, avoiding potential distortion and overcomplication.

HTTP and Web Technologies

- #### HTTP (Hypertext Transfer Protocol) is the foundation of any data exchange on the Web. It's a protocol that allows the fetching of resources, such as HTML documents. Let's dive into the basics and understand the importance of its various components.

Basics of HTTP

- #### HTTP operates as a request-response protocol. A client sends a request, and the server, in turn, sends a response.

Sample HTTP Request:

> ### GET /index.html HTTP/1.1
>
> ### Host: www.example.com

Sample HTTP Response:

> ### HTTP/1.1 200 OK
>
> Date: Mon, 27 Jul 2009 12:28:53 GMT Server: Apache Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT Content-Length: 88 Content-Type: text/html

Hello, World!

Status Codes

- #### HTTP status codes indicate the outcome of the HTTP request. They fall into classes: informational (1xx), successful (2xx), redirection (3xx), client errors (4xx), and server errors (5xx).

In [9]:
```python
# Python code to make a GET request and check status code
import requests

response = requests.get('https://www.example.com')
print(f"Status Code: {response.status_code}")
```

```
Status Code: 200
```

Understanding Elements like Port Numbers in URLs

- #### Every web server listens on a port, typically 80 for HTTP and 443 for HTTPS. A URL might sometimes include a port number, especially if it's a non-standard port.

Example: http://www.example.com:8080/index.html

Here, the port number is 8080.

Web Content Structures

> Web pages can be static (the same for every user) or dynamic (adapted based on the user, context, or other parameters).

Static Web Pages

- ##### They are fixed-content pages. Their content doesn't change unless it's manually updated by a developer.

Dynamic Web Pages

- ##### Content changes based on user interactions, database interactions, or other parameters.

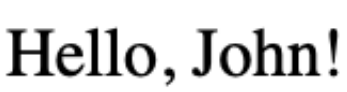
In [ ]:
```python
# A simple dynamic web page using Python's Flask
from flask import Flask

app = Flask(__name__)

@app.route('/greet/<name>')
def greet(name):
    return f"Hello, {name}!"

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

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production d
eployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

```
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Sep/2023 20:17:30] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [16/Sep/2023 20:19:22] "GET /greet/John HTTP/1.1" 200 -
```

Key Points:

- HTTP operates as a request-response protocol.
- HTTP status codes indicate the outcome of a request.
- Web servers listen on ports, typically 80 for HTTP and 443 for HTTPS.
- Web pages can be static or dynamic. Static pages remain the same for every user, while dynamic ones change based on various parameters.



NumPy (Numerical Python)

- ### NumPy is a library for the Python programming language, allowing for efficient numerical operations on large arrays and matrices of numeric data. It provides a high-performance multidimensional array object and tools for working with these arrays. It's fundamental for scientific computing with Python and serves as the foundational package for many other scientific libraries, such as pandas, scikit-learn, and SciPy.

Key Features:

1. ### Multidimensional Arrays: At the core of the NumPy package is the ndarray object which encapsulates n-dimensional arrays of homogeneous data types.
2. ### Broadcasting: A powerful feature that lets you perform arithmetic operations on arrays of different shapes.
3. ### Mathematical Functions: NumPy provides a comprehensive set of mathematical functions to operate on these arrays.
4. ### Linear Algebra: Contains built-in functions for linear algebra calculations.
5. ### Integration with C/C++ and Fortran: NumPy can also be used as a flexible container for generic data to seamlessly integrate with legacy data.

In [3]:

```python
import numpy as np

# Create a simple array
arr = np.array([1, 2, 3, 4, 5])
print("Array:", arr)

# Compute the mean
mean_val = np.mean(arr)
print("Mean:", mean_val)

# Reshape to a 2D array
arr_2d = arr.reshape(5, 1)
print("Reshaped Array:\n", arr_2d)

# Matrix multiplication
mat1 = np.array([[1, 2], [3, 4]])
mat2 = np.array([[2, 2], [2, 2]])
result = np.dot(mat1, mat2)
print("\n")
print("Matrix 1:\n", mat1)
print("Matrix 2:\n", mat2)
print("\n")
print("Matrix Multiplication Result:\n", result)
```

```
Array: [1 2 3 4 5]
Mean: 3.0
Reshaped Array:
 [[1]
 [2]
 [3]
 [4]
 [5]]


Matrix 1:
 [[1 2]
 [3 4]]
Matrix 2:
 [[2 2]
 [2 2]]


Matrix Multiplication Result:
 [[ 6  6]
 [14 14]]
```

Best Wishes!🤍

Yassin Nawar