

## Quiz

### 1. Benefits of File I/O in Python:

- Data Persistence: File I/O អនុញ្ញាតឱ្យរក្សាទុកទិន្នន័យជាប់លាប់នៅលើថាស ដូច្នេះវាអាចត្រូវបានទាញយកនៅពេលក្រោយ ទោះបីជាកម្មវិធីបានបញ្ចប់ការប្រតិបត្តិក៏ដោយ។
- Data Sharing: Files ផ្តល់មធ្យោបាយទូទៅក្នុងការចែករំលែកទិន្នន័យរវាងកម្មវិធីផ្សេងៗគ្នា ឬសូម្បីតែករណីផ្សេងគ្នានៃកម្មវិធីដូចគ្នា។
- Configuration Storage: Settings and configurations អាចត្រូវបានរក្សាទុកក្នុងឯកសារ ដែលធ្វើឱ្យកាន់តែងាយស្រួលក្នុងការកែប្រែឥរិយាបថកម្មវិធីដោយមិនចាំបាច់ផ្លាស់ប្តូរកូដប្រភព។
- Data Analysis: Reading data ពីឯកសារគឺជាវិធីធម្មតានៅក្នុងកិច្ចការវិភាគទិន្នន័យ ដែលសំណុំទិន្នន័យធំត្រូវបានរក្សាទុកក្នុងឯកសារខាងក្រៅ។

### 2. Built-in Function for File I/O in Python:

- The built-in `'open ()'` function ប្រើជាទូទៅសម្រាប់បង្កើតវត្ថុដើម្បីអាន/សរសេរមាតិកាទៅឯកសារ។ វាត្រឡប់វត្ថុឯកសារដែលអាចត្រូវបានប្រើដើម្បីអនុវត្តប្រតិបត្តិការឯកសារផ្សេងៗ។

### 3. File Access Modes:

- File access modes specify the purpose for which you open the file. They are represented by a combination of characters:
- 'r': Read (default mode). Opens the file for reading.
- 'w': Write. Opens the file for writing. Creates a new file or truncates the existing file to zero length.
- 'a': Append. Opens the file for writing, appending to the end of the file if it exists.
- 'b': Binary mode. Reads or writes the file in binary mode (e.g., 'rb' or 'wb').
- 't': Text mode (default mode). Reads or writes the file as a text file (e.g., 'rt' or 'wt').
- '+': Update. Opens the file for both reading and writing.

### 4. Difference between Bytes Text and Plain Text:

- Bytes Text: Data ត្រូវបានចាត់ទុកជាលំដាប់នៃBytes។ ប្រើរបៀបគោលពីរ ('b') នៅពេលបើកឯកសារ។ សាកសមសម្រាប់ឯកសារដែលមិនមែនជាអត្ថបទ (e.g., images, videos).
- Plain Text: Data ត្រូវបានចាត់ទុកជាលំដាប់នៃអក្សរ។ ប្រើមុខងារអត្ថបទ ('t') នៅពេលបើកឯកសារ។ សាកសមសម្រាប់ឯកសារអត្ថបទ (e.g., .txt files).

### 5. Code Example for Reading and Holding Content from a File:

Code:

```
python
```

```
# Open a file in read mode ('r')
with open('example.txt', 'r') as file:
    # Read the entire content of the file into a string
    content = file.read()

    # Alternatively, read line by line into a list
    # lines = file.readlines()

# Display the content
print(content)
```