Sequential Programming Plan

Topic 03 Key concept

1. Sequential Programming

1.1 Sequential Structure

- Sequential refers to the process of executing commands created by coding in order.
- Although programs appear to perform tasks simultaneously due to their high speed, computers actually process commands in the order specified in the code.
- Naturally, even if the commands are the same, completely different results will occur if the order is changed.

1. Sequential Programming

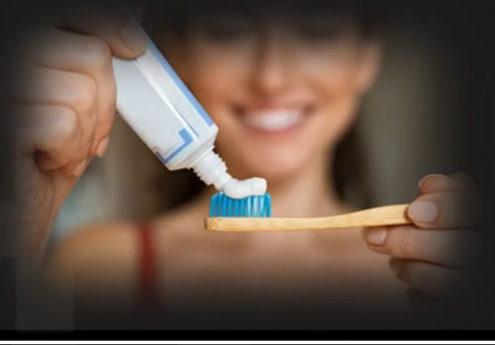
1.1 Sequential Structure

Therefore, when learning and coding Python in the future, it is important to determine the order based on the importance and purpose of the commands to create a program.

1. Sequential Programming

1.1 Sequential Structure

There are many cases in daily life where order is important.



put toothpaste on toothbrush

↓

use toothbrush to clean teeth

↓

rinse toothbrush

1. Sequential Programming

1.1 Sequential Structure

What would happen if the order of the same command was changed?

put toothpaste on toothbrush

↓

use toothbrush to clean teeth

↓

rinse toothbrush

→

rinse toothbrush

↓

put toothpaste on toothbrush

↓

use toothbrush to clean teeth

1. Sequential Programming

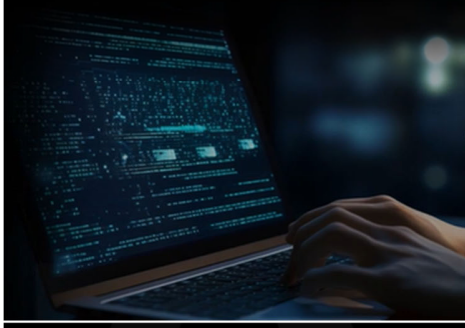
1.1 Sequential Structure

What would happen if the order of the same command was changed?

Similarly, if the order of commands changes, the results will be different and it will be difficult to achieve the desired outcome.

1. Sequential Programming

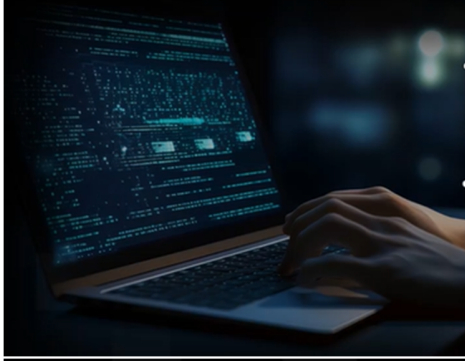
1.2 Pseudocode



- Pseudocode is the process of writing commands for the computer to execute in order.
- It is a good way to start coding with limited knowledge of programming because it uses everyday language.
- It can also serve as a bridge when completing flowcharts and transitioning to actual coding.

1. Sequential Programming

1.2 Pseudocode



- It can be used as a basic guide document among programmers during the implementation phase of the actual code.
- The main goal is to clearly describe and plan what needs to be done for each line of code.

1. Sequential Programming

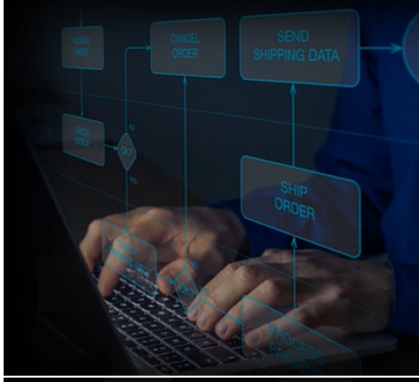
1.2 Pseudocode

```
This program will allow the user to
check the number whether it's even or
odd

If “4”
  print result
  “This number is even”

If “3”
  print result
  “This number is odd”
```

- Start with a sentence that explains the purpose of the implementation.
- Write concisely.
- Use simple and distinguishable names, following rules to create them.
- Indentation and spacing are crucial.
- Express sequential structure, conditional structure, and repetitive structure (using if, for, while).

1. Sequential Programming

1.3 Flowchart



- A flowchart is a representation of commands using standardized shapes and arrows to depict the flow of operations.
- The shapes and symbols used adhere to the standards set by the International Organization for Standardization (ISO).
- The flowchart should be written without overlapping, with the direction from top to bottom and left to right.

1. Sequential Programming

1.3 Flowchart

Start , End

Indicates the start and end of a flowchart.

Preparation

Initializing data.

Action or Process

A single calculation or operation.

Decision

Comparing and judging conditions, branching the flow based on the results.

Input, Output

Input and output of data.

Document

Indicates what will be printed as a document.

On-page Connector

Indicates the connection to another flowchart with this symbol.

Flow Arrow

Indicates the flow of processing.

1. Sequential Programming

1.4 Three Structures Represented by Flowcharts for Program Execution

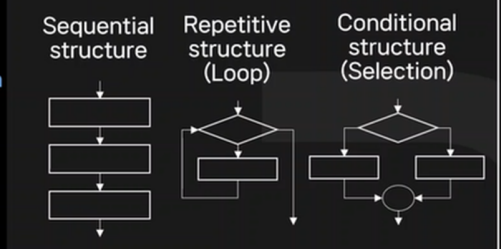
- Flowcharts represent sequential, conditional, and repetitive structures.
  These three structures can be applied not only in coding but also in real life.

In daily life

- ➡ Go straight west at the intersection.
- ➡ If you see a post office, turn right at the post office.
- ➡ Continue straight.
- ➡ If you see an Italian restaurant, the third building is the accommodation you are looking for.

Application ➜

Three structures of a program



1. Sequential Programming

1.4 Three Structures Represented by Flowcharts for Program Execution

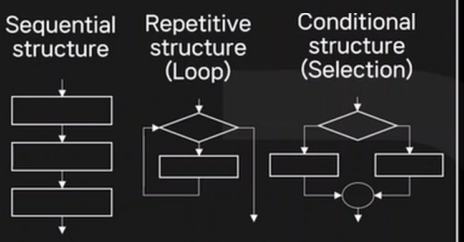
- First, they need to be written in sequential structure. Then, some parts of the order can be developed into repetitive or conditional structures.

In daily life

- Go straight west at the intersection.
- If you see a post office, turn right at the post office.
- Continue straight.
- If you see an Italian restaurant, the third building is the accommodation you are looking for.

Application →

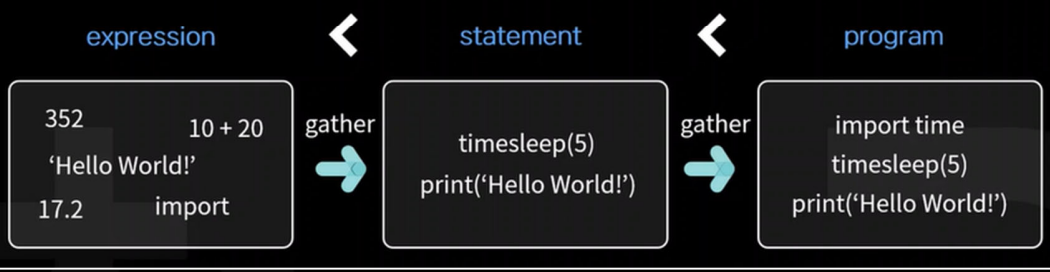
Three structures of a program



1. Sequential Programming

1.5 Expressions, Statements, Programs

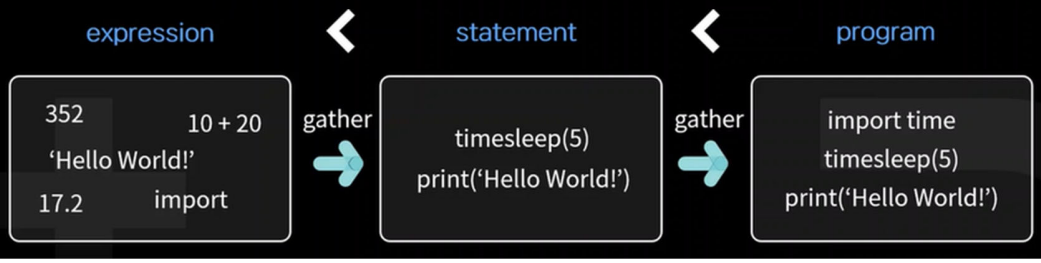
- Plans created with pseudocode and flowcharts need to be developed into programming by coding expressions.



1. Sequential Programming

1.5 Expressions, Statements, Programs

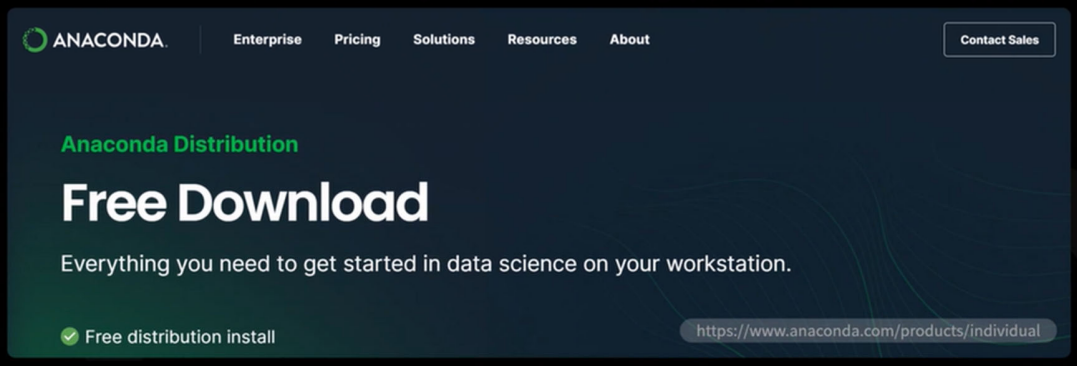
- An expression is a simple code that produces a value, such as numbers, formulas, and strings. A collection of expressions is called a statement. Multiple statements form a program.



2. Setting up the Practical Environment

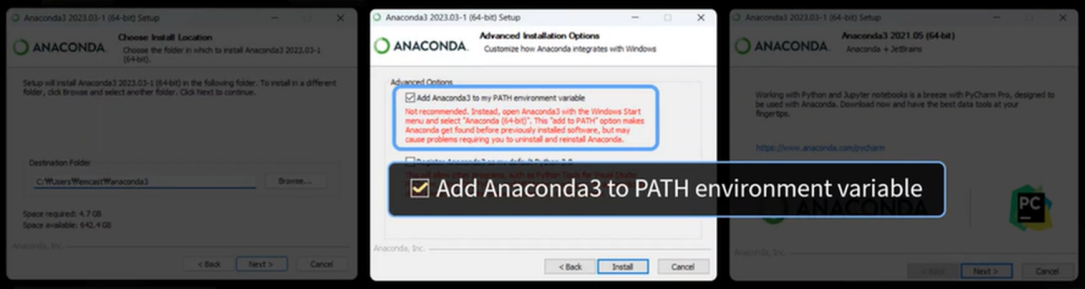
2.1 Installing Anaconda

- We will set up the development environment using Anaconda for coding.



2. Setting up the Practical Environment

2.1 Installing Anaconda



- If you have previously installed a separate Python program, you need to uncheck “Add Anaconda3 to my PATH environment variable” in Advanced Options because it will conflict with python.exe in Anaconda.

2. Setting up the Practical Environment

2.2 Creating Virtual Environments

- When programming with Python, you often use external packages.
- When managing multiple projects that use external packages or starting a new project, you may need to use the latest version of Python or different types/versions of packages.
- In such cases, it becomes very difficult to manage modules and versions, and problems can occur if they are not compatible between versions.

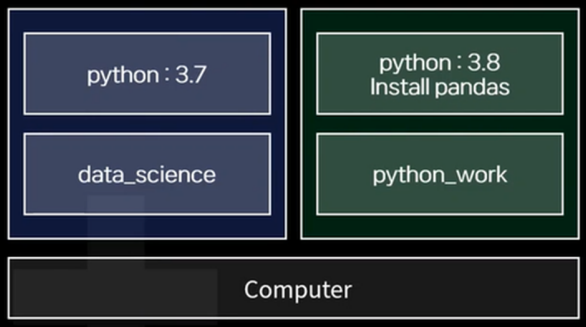
2. Setting up the Practical Environment

2.2 Creating Virtual Environments

- For this reason, we will learn how to create virtual environments.
- A Python virtual environment provides an independent space where you can manage modules and versions for each project. It allows you to build various virtual environments on a single server/computer.

2. Setting up the Practical Environment

2.2 Creating Virtual Environments

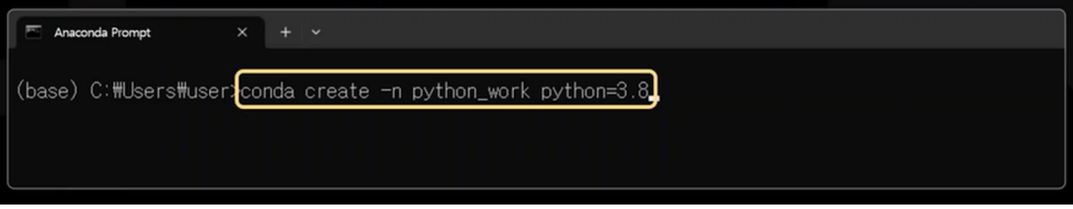


- For example, you can work with python 3.7 in the data_science virtual environment and work with python 3.8 and pandas in the python_work virtual environment, which gives you the effect of using two computers.

2. Setting up the Practical Environment

2.2 Creating Virtual Environments

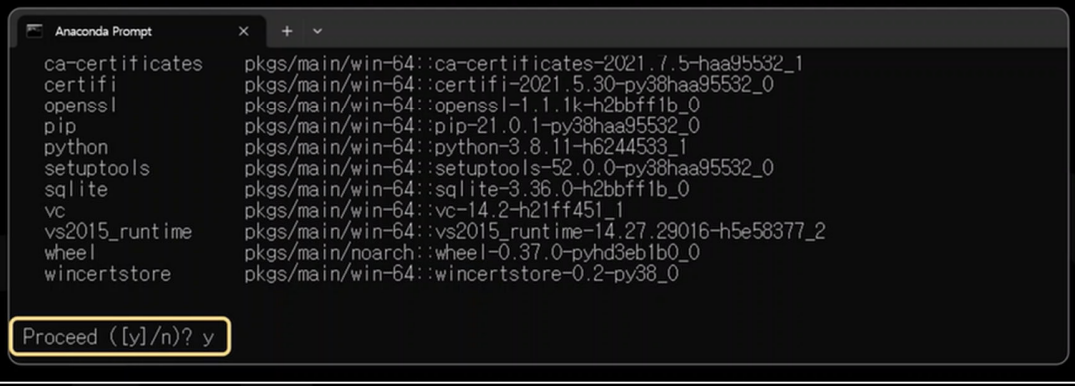
- To create a Python virtual environment, run Terminal on macOS or Anaconda Prompt on Windows.
- In the Terminal or Anaconda Prompt window, enter the command "conda create -n python_work python=3.8" to create a virtual environment compatible with Python 3.8. ‘python_work’ is the name of the virtual workspace and can be defined with a different name.



2. Setting up the Practical Environment

2.2 Creating Virtual Environments

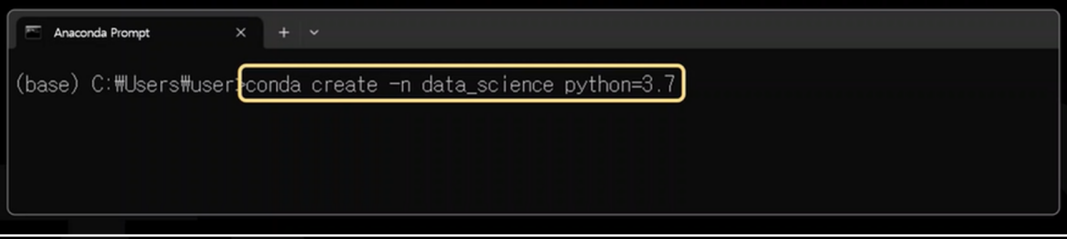
- When the prompt displays "Proceed ([y]/n)?", enter 'y' to continue the process.



2. Setting up the Practical Environment

2.2 Creating Virtual Environments

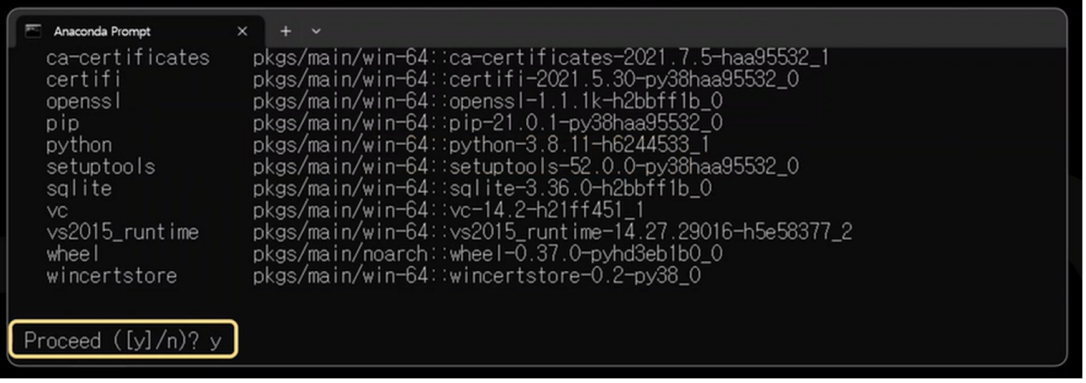
- Let's try entering the command "conda create -n data_science python=3.7" in the Terminal or Anaconda Prompt. This will create a new virtual workspace named 'data_science' compatible with Python 3.7.



2. Setting up the Practical Environment

2.2 Creating Virtual Environments

- Again, when the prompt displays "Proceed ([y]/n)?", enter 'y' to continue the process.



2. Setting up the Practical Environment

2.3 Anaconda Commands for Managing Virtual Environments

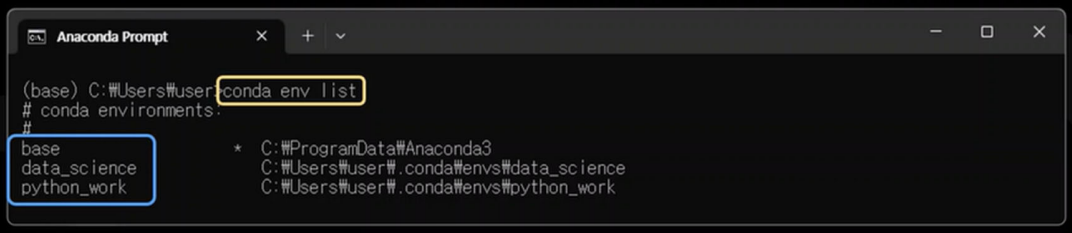
- You can perform various operations related to virtual environments by entering the following virtual environment commands in Terminal or Anaconda Prompt.

Action	Command
List virtual environments	conda env list
Create a virtual environment	conda create –n [environment name]
Clone a virtual environment	conda create --clone [source environment] --n [new environment]
Activate a virtual environment	conda activate [environment name]
Deactivate the current virtual environment	conda deactivate
Remove a virtual environment	conda remove --name [environment name] --all

2. Setting up the Practical Environment

2.3 Anaconda Commands for Managing Virtual Environments

- Run "conda env list" command in Terminal or Anaconda Prompt to see the list of virtual environments in Python.
- The default value is 'base', and you should select the virtual environment you want to install. Previous virtual environments such as ‘data_science’ and ‘python_work’ should be listed.



2. Setting up the Practical Environment

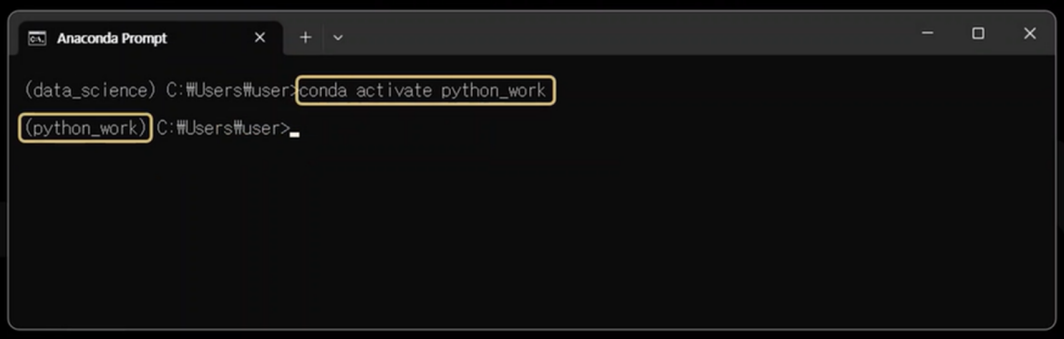
2.3 Anaconda Commands for Managing Virtual Environments

- Use the command "conda activate data_science" to switch to a new working environment. The prompt will change from 'base' to 'data_science' to indicate the changed working environment.

2. Setting up the Practical Environment

2.3 Anaconda Commands for Managing Virtual Environments

- This time, try switching to another working environment named ‘python_work’ by using the command "conda activate python_work".



2. Setting up the Practical Environment

2.4 Installing Jupyter Notebook

- Once you have completed the Anaconda installation, you can install Jupyter Notebook.

3. Writing Output Statements in Jupyter Notebook

3.1 Basic Usage of the print Statement

- Let's write a code to print "Hello World!" in Jupyter Notebook.
- The print() statement is used to display the value inside the parentheses.

```
print("Hello World!")
```

You can use either single or double quotation marks to define a string, but they should match at the beginning and the end.

The content between the opening and the closing quotation marks is called a string.

TIP

Unlike other languages, Python automatically adds a line break after printing a string with a single print statement.

3. Writing Output Statements in Jupyter Notebook

3.2 Printing Strings and Values

- To print a string using the print statement, enclose the string in quotation marks.

```
1 print('Hello World!')
2 print("Hello World!")
```

```
Hello World!
Hello World!
```

- To print a numeric value, simply enter the number without quotation marks.

```
1 print(10)
2 print(7.5)
```

```
10
7.5
```

- If you include quotation marks when printing a number, it will be treated as a string.

3. Writing Output Statements in Jupyter Notebook

3.2 Printing Strings and Values

- You can use a comma (,) within the print statement to print multiple values at once.

```
1 print('The radius of a circle', 4.0)
```

```
The radius of a circle 4.0
```

3. Writing Output Statements in Jupyter Notebook

3.3 Creating a Self-Introduction

- Let's create a self-introduction using the print statement.

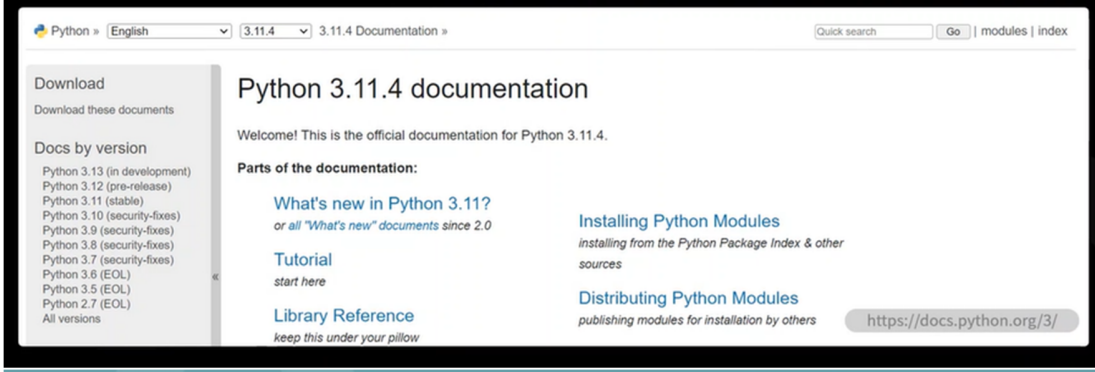
```
print('Hello! I will introduce myself.')
print('Name: David Doe')
print('Age: 23')
print('Job: Data Scientist')
print('Address: Seoul, South Korea')
print('Place of Birth: Southern California, USA')
```

- You can decorate the self-introduction by repeating asterisks (*) or hyphens (-).

```
print('**************************************************')
print('--------------------------------------------------')
```

4. Basic Terminology for Learning

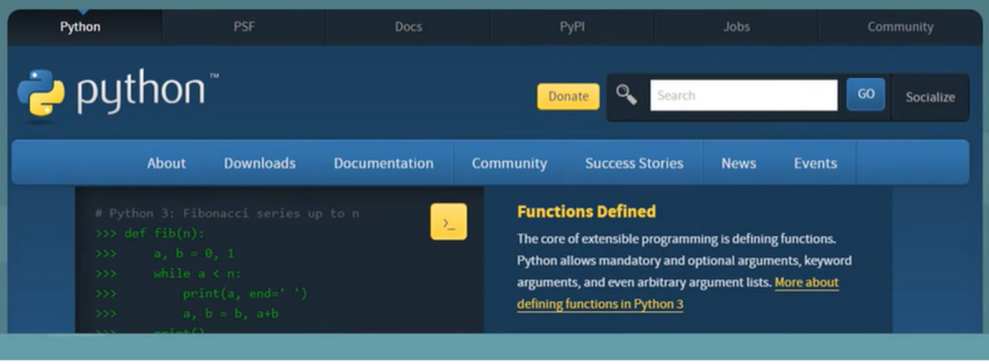
- When encountering new terms during the learning process, it's important to make a note of them and verify their exact definitions.
- It's also a good idea to refer to tutorials provided by the official Python documentation.



One Step Further

1. Introduction to Python Learning References

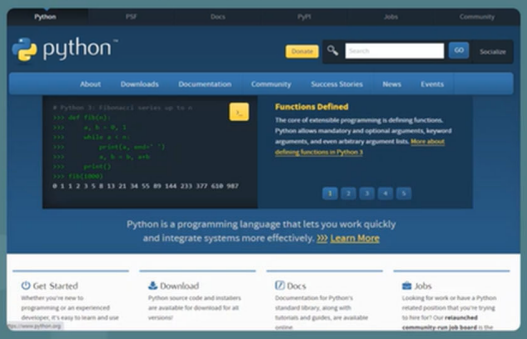
▶ The following learning references are useful for studying Python and can be referred to when you need help or want to learn more about Python.



One Step Further

1. Introduction to Python Learning References

▶ Python Software Foundation



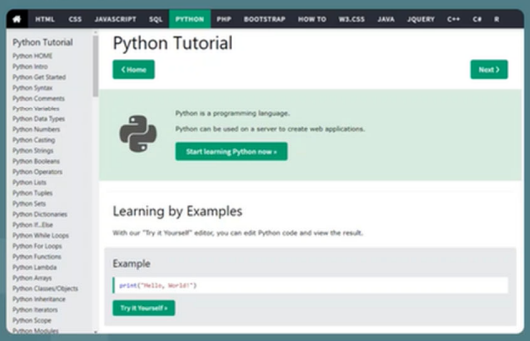
- A non-profit organization responsible for the development and distribution of Python.
- Provides free downloads of the latest Python version.
- Offers tutorials and documentation for learning Python.

https://docs.python.org/3/

One Step Further

1. Introduction to Python Learning References

▶ W3Schools



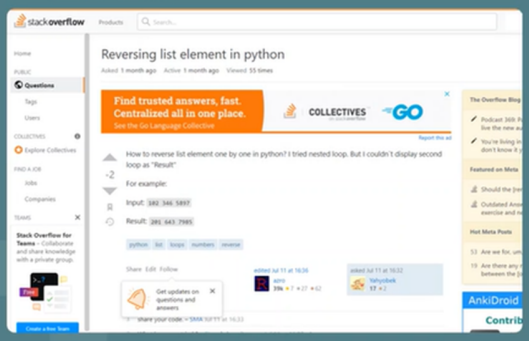
- To expand and apply Python further, where can you find the necessary resources?
- W3Schools offers tutorials on various programming languages, including Python.
- It provides the advantage of being able to run Python code examples directly in a web environment.

https://www.w3schools.com

One Step Further

1. Introduction to Python Learning References

▶ Stack Overflow



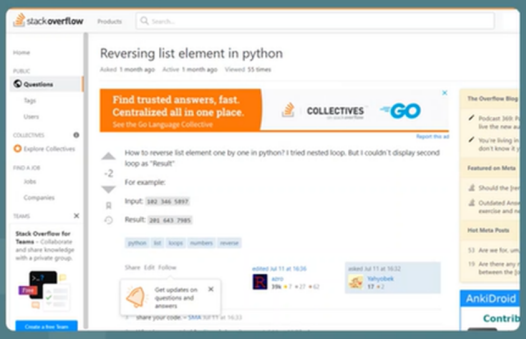
- When encountering problems or difficult issues during programming, how can you solve them?
- Stack Overflow is a globally renowned site for sharing developer information.

https://stackoverflow.com/

One Step Further

1. Introduction to Python Learning References

▶ Stack Overflow



- Try searching for "reverse list element in python" on this site. You will find many questions and discussions related to lists, which you will learn about later.

https://stackoverflow.com/