# 30 days with python

Zainal Fitri Nur Syarafina Noriah

# 30 days with python

Zainal Fitri Nur Syarafina Noriah

### Authors: Zainal Fitri bin Mohd Zolkifli Nur Syarafina binti Abdul Rahman Noriah binti Razali

#### Copyright© 2023 www.psmza.edu.my

Zainal Fitri bin Mohd Zolkifli | Nur Syarafina binti Abdul Rahman | Noriah binti Razali Jabatan Teknologi Maklumat dan Komunikasi Politeknik Sultan Mizan Zainal Abidin Dungun, Terengganu

#### Copyright © 2023

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.



e ISBN: 978-967-0047-46-1

First Edition: Nov, 2023

30 Days with Python

Published by: Politeknik Sultan Mizan Zainal Abidin KM 08, Jalan Paka, 23000 Dungun, Terengganu

### Synopsis

"30 Days with Python" offers readers a delightful and accessible journey through the world of Python programming. This book is not a comprehensive guide, but rather a collection of brief, engaging articles designed to complement your learning experience. Each day presents a new topic, providing a light-hearted exploration that touches the surface of Python's diverse capabilities.

From the fundamentals of variables and data types to more advanced concepts like loops, functions, and Object-Oriented Programming (OOP), this book provides a taste of it all. Whether you're a complete beginner or looking to refresh your Python skills, these bite-sized articles offer a fun and informative way to expand your programming knowledge.

Each day's article is crafted to be a quick, enjoyable read, making it perfect for those with busy schedules or just a few spare moments to spare. While not intended to replace comprehensive guides, "30 Days with Python" serves as a delightful companion, sparking curiosity and encouraging further exploration.

By the end of this 30-day journey, you'll have gained a broad understanding of Python's capabilities, setting the stage for further exploration and deeper learning. So, embark on this light-hearted adventure and let Python's charm and versatility unfold before you. Happy coding!

### Preface

Assalamualaikum and greetings,

Many thanks to Allah SWT, the eBook is completed successfully. The purpose of this e-book is to be one of the source for references to polytechnic students and also for those who want to learn and get a glimpse of Python programming. It is hoped that this eBook will be beneficial to all.

Python has become a favorite among programmers worldwide for its versatility, clean syntax, and extensive libraries. Whether you're a seasoned developer or just starting, this book is designed to be as secondary read to complement other books on Python Programming.

"30 Days with Python" is built on the belief that consistent learning is key to real proficiency. Over the next 30 days, we will take a structured approach to explore Python's fundamentals and progress to more advanced topics, Object-Oriented Programming (OOP).

Each day, you will find a brief yet comprehensive article that will introduce a new concept or technique. You'll tackle practical examples that provide the hands-on experience necessary to solidify your understanding of each topic. By the end of this journey, you'll have a good understanding of Python Programming.

Remember that learning to code is a journey, and we're excited to have you along for the ride. I encourage you to embrace each day's lesson, experiment, and explore on your own. The Python community is vast and supportive, and there are countless resources available to assist you in your learning journey.

Thank you for choosing "30 Days with Python" as your guide. By the end of this journey, you'll have a valuable skillset and a newfound appreciation for the art and science of programming.

Wishing you a fulfilling and enlightening journey ahead. Happy coding!

Sincerely,

Zainal Fitri bin Mohd Zolkifli Author

### Contents

Introduction	2
Day 1	<u>3</u>
Day 2	<u>6</u>
Day 3	<u>8</u>
Day 4	<u>10</u>
Day 5	<u>12</u>
Day 6	<u>13</u>
Day 7	<u>15</u>
Day 8	<u>16</u>
Day 9	<u>17</u>
Day 10	<u>19</u>
Day 11	<u>20</u>
Day 12	<u>22</u>
Day 13	<u>23</u>
Day 14	<u>27</u>
Day 15	<u>28</u>
Day 16	<u>29</u>
Day 17	<u>31</u>
Day 18	<u>32</u>
Day 19	<u>33</u>
Day 20	<u>34</u>
Day 21	<u>36</u>
Day 22	<u>39</u>
Day 23	<u>40</u>
Day 24	<u>41</u>
Day 25	<u>42</u>
Day 26	<u>44</u>
Day 27	<u>46</u>
Day 28	<u>48</u>
Day 29	<u>51</u>
Day 30	<u>55</u>

### INTRODUCTION

"30 Days with Python" is a comprehensive guide to learning Python, suitable for beginners and experienced coders. The book contains 30 concise yet comprehensive articles that cover everything from the basics of Python programming to the intricacies of Object-Oriented Programming (OOP).

Each day builds on the knowledge gained from the previous, ensuring steady progression towards better understanding of Python and brings a new opportunity to hone your skills and broaden your horizons.

By the end of the 30-day journey, readers will have a good understanding of Python and the confidence to start Python projects and explore the endless potential of this remarkable language.

Get ready to code, learn, and transform your programming prowess. Let's begin "30 Days with Python" and set off on a transformative journey through the world of Python programming!



WHAT IS PYTHON AND

Python - a popular programming language in the world of programming. According to the Stack Overflow survey in 2021, Python ranked 4th in the category of the most popular programming languages among software developers and 3rd in the most popular programming language among all respondents. [1]

Why is Python important to learn in the programming world?

Python is used in various applications such as web development, software, data analysis, and artificial intelligence. This means that there are vast job opportunities for those who are proficient in Python.

Many technology companies such as Google, Facebook, Amazon, and Netflix use Python in the development of their products and services.

Here are some examples of technology companies that use Python in the development of their products and services:

- Google Python is used in various Google applications such as Google Search, Google Maps, and YouTube. Google also has an open-source project called Python Fire, which is a Python library that facilitates the creation of command-line interfaces (CLI) for Python. [2]
- Facebook Python is used in the development of Facebook applications and products such as Instagram. Facebook also has an open-source project such as PyTorch which is used in the development of artificial intelligence (AI) and machine learning. [3]
- Amazon Python is used in the development of Amazon products such as Amazon Web Services (AWS) and in data processing. [4]
- Netflix Python is used in the development of Netflix applications and in processing data to generate content backups for users. [5]

In addition, Python has a syntax that is easy to understand and concise code to allow readability, making it a programming language that is easy to learn and use by those who are new to the programming world. [6]

Python's easy-to-read and easy-tounderstand syntax allows programmers to build and develop applications more easily and quickly.

Furthermore, Python has many libraries and modules that have been developed by a large Python community, making it easier for programmers to develop applications more quickly and easily.

In a short period of time, you can learn this programming language and start developing your own applications and projects.

If you want to learn Python, there are many learning resources that you can find online, such as tutorials and courses. You can also join active Python communities to network and get support in your learning.

I hope you will be inspired to learn Python and develop your skills in the field of IT.

Let's do it!

#python #programming

#programminglanguage #technology

#learnpython #itworld #career #dataanalysis

#artificialintelligence #letsdoit

## 



HISTORY AND
EVOLUTION OF PYTHON:
HOW DID IT EVOLVE
FROM THE FIRST
VERSION TO THE
PRESENT?

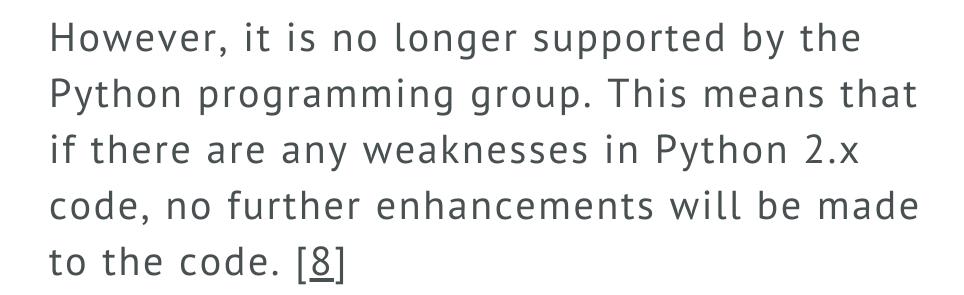
Python - a popular programming language widely used in various applications in the IT world. But do you know how it evolved from its first version to what it is today?

Python was first introduced in 1991 by Guido van Rossum and has gone through several developments since then. The first version of Python was known as Python 0.9.0 and had weaknesses in performance and security. However, with each new version, Python continued to evolve and overcome existing problems.

In 2000, Python 2.0 was released with some performance improvements and new features such as support for Unicode, garbage collection, and improved exception handling. Python 2.7.18 is the last version of the Python 2.x series. [7]

In 2008, Python 3.0 was released with the introduction of a more consistent programming language and improvements to some weaknesses in previous versions.

However, there were some changes that were not compatible with the existing Python programming language. Therefore, some organizations still use Python 2.x.



Today, Python 3.10 is the latest version and has some new enhancements in programming language and performance optimization. It is used in various applications such as web development, artificial intelligence, and data analysis. [9]

In the world of programming, Python continues to be a popular and evolving programming language. Wide career opportunities and improvements in performance and usability in every new version make Python an important programming language to learn and master.

So, let's learn Python and follow its development in the world of programming!

Let's do it.

#python #programming

#programminglanguage #technology

#learnpython #itworld #career #dataanalysis

#artificialintelligence #letsdoit



WHAT IS THE
DIFFERENCE BETWEEN
PYTHON AND OTHER
PROGRAMMING
LANGUAGES LIKE JAVA
AND C++?

Today, we will discuss the differences between Python and other programming languages like Java and C++. Python has a syntax that is easy to read and write, making it one of the most popular programming languages among programmers and data scientists.

Unlike Java and C++, Python does not require specific data type declarations before use. This makes Python easier and more flexible to use, especially for new users. [10]

Python also has many libraries that can be used, such as numpy [11], pandas [12], and matplotlib[13], which allow for easy and fast use in data analysis and scientific software development.

However, the biggest difference between Python and Java/C++ is that Python is an interpreted programming language, while Java/C++ are compiled programming languages. [14]

This means that Python requires interpretation every time the program is run, while Java/C++ needs to be compiled into machine code first.



Do you have experience using Python or other programming languages? Please share your thoughts in the comments below!

Let's do it together.

#PythonVsJavaVsCPlusPlus
#ProgrammingLanguages #DataScience
#ScientificSoftware #MachineCode
#Interpretation #PythonLibraries
#learnpython #let'sdoittogether



WHO USES PYTHON: EXAMPLES OF PYTHON USE IN INDUSTRY AND ITS IMPACT.

Have you ever wondered who uses the Python programming language and what its impact is on the industry?

Today, let's delve into this issue! Python is increasingly gaining popularity in various industries, including technology, scientific software, artificial intelligence, and many more. Many renowned companies such as Netflix [15], Spotify [16], and Dropbox [17] use Python to develop their software.

Python is used in web application development such as Instagram [18] and Pinterest [19], as well as in natural language processing and information modeling at Airbnb [20]. It is also used for data analysis and scientific software development in the science and technology sector, such as NASA [21] and CERN [22].

The advantages of Python are programming speed, ease of understanding, and simple and flexible syntax. Additionally, many libraries and tools are available openly, such as numpy, pandas, and matplotlib, which make data analysis and usage easier.



Let's do it! 🔐

Join the Python community and explore new opportunities waiting for you!

#Python #Technology #DataScience
#ArtificialIntelligence #Netflix #Spotify
#Dropbox #letsdoit

# 



HOW TO INSTALL
PYTHON ON WINDOWS,
MACOS, AND LINUX
COMPUTERS?

Today, we will discuss how to install Python on Windows, MacOS, and Linux operating systems.

- To install Python on Windows, you can go to the official Python website and download the installation package that corresponds to your Windows version.
- For MacOS, you can download the Python installation from the Python website or install it via Homebrew or MacPorts.
- \* For Linux users, you can install Python through the Terminal using a suitable package manager.

Remember to download the latest version and check the minimum system requirements before installing Python.

It's easy to install Python on Windows, MacOS, and Linux operating systems. By understanding how to install Python, you can start programming more easily. Let's give it a try!

#Python #Technology #Windows #MacOS
#Linux #letsdoit



#### DIFFERENCES BETWEEN PYTHON 2 AND PYTHON 3

Perhaps you are wondering, between Python 2 and Python 3, which one should you choose?

Python 2 was introduced in 2000, while Python 3 was introduced in 2008. Although both Python versions are still in use today, Python 3 is the recommended version by Python developers.

Python 2's support ended in 2020, meaning it will no longer receive security patches or new features after that.

So, if you want to learn Python, choose Python 3. You are also advised to use Python 3 in developing new software because in the long run, using Python 3 will allow you to have access to the latest features in the Python programming language and make it easier to use new features that are available in Python 3 for your software in the future.

Don't worry, there are many resources available for learning Python 3 out there. I'm sure you can learn Python 3 easily and quickly!



Let's do it!

#Python #Python2 #Python3
#ProgrammingLanguage
#SoftwareDevelopment #letsdoit

DAY 6



WHAT IS A VIRTUAL ENVIRONMENT AND HOW CAN IT HELP IN PYTHON APPLICATION DEVELOPMENT?

Today, let's talk about Virtual Environment in Python application development.

What is Virtual Environment and how can it help in Python application development?

Nirtual Environment is a tool that helps in logically separating your Python project from other projects using Python. By using Virtual Environment, you can have different versions of Python and libraries for each project without interfering with other projects on your system.

Why is this important? In Python application development, the version of Python and libraries used may differ between projects. Therefore, using Virtual Environment allows you to have a separate development environment for each of your projects to avoid any conflicts.

If you're interested in learning more about Virtual Environment and how it can help in Python application development, you can refer to the links below.

Let's do it together.

- #Python #VirtualEnvironment
- #SoftwareDevelopment #Programming
- #Technology #let'sdoittogether



WHAT IS INDENTATION AND HOW DOES IT AFFECT THE STRUCTURE OF PYTHON CODE?

Let's discuss the concept of Indentation in Python and how it affects the structure of Python code. Indentation refers to the number of spaces or tabs used to denote a certain level in Python code.

In other programming languages, curly braces ({}) are usually used to indicate code blocks, but in Python, indentation is used instead.

This provides cleaner and more readable code. However, it also requires additional efficiency in the written code because errors in indentation can cause errors in the code.

Therefore, make sure that your indentation is consistent and correct!

Let's Do It!

#PythonApplication #IndentationConcept

#CodeStructure #CleanCode

#ProgrammingTips #let'sdoit



UNDERSTANDING PEP8:
HOW TO FOLLOW THE
PEP8 GUIDE IN CLEAN
AND EASY-TO-READ
PYTHON CODE WRITING?

Today, let's talk about PEP8 and how we can use it in Python code writing so that the code is more structured and easy to read.

- PEP8 is an acronym for Python Enhancement Proposal 8, a document outlining the Python code writing guide agreed by the Python community.
- This document provides guidance on naming patterns, code conventions, and the use of free space in codes.
- To write consistent and easy-to-read Python code, it is very important to follow the PEP8 guide. Some important PEP8 guides to follow include:
- Use four spaces for each stage of indentation
- 2 One line of code not exceeding 79 characters
- 3 Use words without underscore (underscore) for package names, modules, and functions



Avoid the use of single characters such as uppercase "O", lowercase "l", or the number "1" in variable names

By following the PEP8 guide, our code will become more readable and understandable to others.

So, don't forget to read the PEP8 document and follow its guide when writing Python code.

Let's do it!

#PEP8 #Python #Python2 #CleanCode
#CodeConvention #Indentation #let'sdoit

## 



USE OF COMMENTS IN
PYTHON CODE: WHAT IS
THE PURPOSE OF
COMMENTS AND HOW
TO USE COMMENTS IN
PYTHON CODE?

Comment in Python code! 🔊

Comments are used to provide descriptions or posts in code. This facilitates the reading and understanding of the code we write. In Python, comments are marked by using the hashtag (#) at the beginning of the comment line.

Comments are an important part of code writing that allows programmers to provide additional information about the written code. It will also be ignored by the interpreter during the implementation of the code.

As a good programmer, be sure to use comments to help others understand your code. However, don't use too many unnecessary comments and interrupt the entire code.

Let's do it!

#PythonProgramming #Comment #CleanCode
#let'sdoit



IDENTIFYING DATA
TYPES IN PYTHON: WHAT
TYPES OF DATA ARE
THERE IN PYTHON?

Hi all! 🔊

Let's talk about the types of data in Python!

In Python, there are several basic types of data. The types of data are:

- Numeric this class consists of three types, namely int, float, and complex. This class is used to store the values of numbers in Python.
- 2 String this class consists of only one type which is str. This class is used to store a sequence of characters or words in Python.
- 3 Sequence this class is of three types, namely list, tuple, and range. This class is used to store groups of items in a specific order.
- 4 Mapping this class consists of only one type, that is, dict. This class is used to store data in the form of a key-value pair.
- Boolean this class consists of only one type which is bool. This class is used to store values that can only be true or false.



NoneType is a specific data type in Python that represents empty or valueless objects. It is used to denote a situation where there is no value available or when an object needs to be defined but its actual value is not required.

Each of these types of data has its own characteristics and uses in Python programming. Therefore, understanding these types of data is very important for anyone who wants to learn Python.

Let's do it!

#Python #DataType #Integer #Float #Boolean
#String #List #Tuple #Set #Dictionary
#let'sdoit



A GUIDE TO USING
NUMBERS IN PYTHON:
UNDERSTANDING
INTEGERS VS FLOATS

Python uses two primary number types: integer and float. Integers are whole numbers that do not include fractions or decimals, while floats are numbers that have fractions or decimals.

Mathematical operations such as addition, subtraction, multiplication, and division can be performed on both integer and float numbers. However, it is important to note that the result of an integer operation will always yield an integer value, whereas the result of a float operation will yield a float value. So, when selecting the number type for your operation, be mindful of this distinction!

Ready to dive into the world of Python programming? Let's do it!

#PythonProgramming #Integer #Float
#MathematicalOperations #ProgrammingTips
#Let'sDoThis



USING STRING: HOW TO PRODUCE AND MANIPULATE STRINGS IN PYTHON.

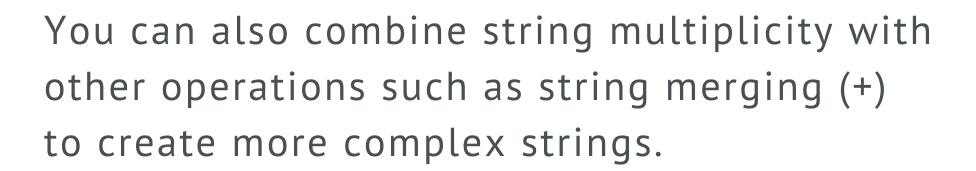
Let's discuss how to produce and manipulate strings in Python!

A string is a type of data that stores a sequence of characters. In Python, we can use a single quote (' ') or double (" ") to denote a string.

Among the operations that can be performed include connecting two strings (concatenation), doubling the string (string replication), and cutting the string (string slicing).

Connecting two strings or concatenation in Python can be done by using the "+" operator to combine two strings together. For example, if we have two strings "Hello" and "World", we can combine them into "Hello World".

Multiplication of strings or string replication is the process of making a copy of a string by copying it several times. In Python, you can use the \* operator to double the string. For example, if you want to double the string "Hello" 3 times, you can write "Hello" \* 3, which will result in the output of "HelloHelloHello".



String cutting or in Python is known as slicing is another way to manipulate strings in Python. The string cut is used to acquire part of the original string. In the cut, we need to specify the starting index and the final index to get the desired subset of strings.

It is worth noting that the index in Python starts from 0 and the final index is not included in the cut-off result.

Let's learn more about using strings in Python!

Let's go! We made him.

#Python #StringManipulation #CodingTips
#BelajarPython #jomkitabuatdia

#### TUTORIAL / EXAMPLE

### data types

Python has several built-in data types that are used to represent different kinds of data. These data types include numbers, strings, lists, tuples, dictionaries, and more. In this tutorial, we will explore each data type and learn how to work with them.

Numbers in Python can be integers
(whole numbers) or floating-point
numbers (numbers with decimal points).
To declare a variable as a number,
simply assign a value to it. For example:

```
x = 10  # integer
y = 3.14  # float
```

You can perform arithmetic operations like addition, subtraction, multiplication, and division on numbers using operators such as +, -, \*, and /. Here's an example:

```
a = 5
b = 2

print(a + b)  # Output: 7
print(a - b)  # Output: 3
print(a * b)  # Output: 10
print(a / b)  # Output: 2.5
```

Strings are sequences of characters enclosed in single quotes (') or double quotes ("). You can perform various operations on strings, such as concatenation (joining strings), indexing (accessing individual characters), and slicing (extracting portions of a string). Here's an example:

Lists are ordered collections of items, enclosed in square brackets ([]). You can store different types of data in a list, and you can modify, access, and iterate over its elements. Here's an example:

#### TUTORIAL / EXAMPLE

Tuples are similar to lists, but they are immutable, meaning you cannot modify their elements after creation. Tuples are defined using parentheses (()). Here's an example:

```
point = (2, 3)
print(point) # Output: (2, 3)

# Accessing tuple elements
print(point[0]) # Output: 2
print(point[1]) # Output: 3
```

Dictionaries are unordered collections of key-value pairs, enclosed in curly braces ({}). Each element in a dictionary consists of a key and its corresponding value, separated by a colon (:). Here's an example:

```
person = {"name": "John", "age": 25, "city":
    "New York"}
print(person) # Output: {'name': 'John',
    'age': 25, 'city': 'New York'}

# Accessing dictionary values
print(person["name"]) # Output: John
print(person["age"]) # Output: 25

# Modifying dictionary values
person["age"] = 30
print(person) # Output: {'name': 'John',
    'age': 30, 'city': 'New York'}
```

In this tutorial, you've learned about some of the fundamental data types in Python, including numbers, strings, lists, tuples, and dictionaries. Understanding these data types is essential for working with data and performing various operations in Python.



**USING BOOLEAN: WHAT** 

What is Boolean?

Boolean refers to a type of data that has only two values: true (True) or wrong (False). In programming, it is used to determine the logical state in the code.

Boolean can be used in flow control statements such as if-else, while, and for.

You can also use logical operators such as and, or, and not to combine several boolean statements.

In Python, the value of True is every value other than 0 and False is 0. Make sure you understand how to use boolean in Python to write more effective code.

Let's do it!

#Python #Boolean #CodingTips #let'sdoit



USING LIST: HOW TO ORGANIZE DATA IN A LIST AND ACCESS ELEMENTS

Let's learn how to organize data in list form and access its elements in Python!

Lists are a type of data that allows us to store a collection of items in one variable. We can compile a list by writing its elements in parentheses (bracket []) and separated by commas.

To access the elements in the list, we'll use the index. The index will start from 0 for the first element in the list and so on.

We can also access the elements in the list using slicing. In Python, list is a frequently used data structures and have many useful functions for manipulating data in lists.

Let's do it!

#Python #List #CodingTips #LearnPython
#let'sdoit



LIST: HOW TO TAKE A
SUBSET OF ELEMENTS IN
A LIST AND MODIFY
THEM.

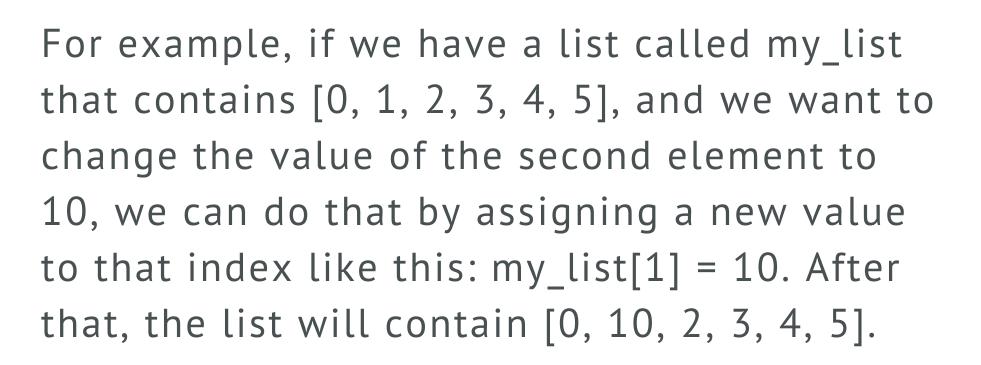
Let's talk more about list.

To take a part of an element or a subset of an element in a list, we can use a technique called slicing. Slicing can take a certain part of the list by compiling the initial and final indexes in brackets.

To accomplish this, you can use the colon (:) to specify the initial and final indexes of the slice you want to take. The initial index indicates the first element to be taken, and the last element to be taken is the final index minus 1.

For example, if we have a list called my\_list that contains [0, 1, 2, 3, 4, 5], we can take a subset of elements with the initial index 1 and the final index 4 by using my\_list[1:4]. This will restore the new list [1, 2, 3].

In addition, we can also modify elements in the list by using an index. We can access the elements in the list by using the index, and then change that value by assigning a new value to that index.



In Python programming, list is one of the most commonly used types of data. Therefore, it is very important to understand how to take a subset of elements in the list and how to manipulate it.

Let's do it!.

#Python #List #CodingTips #LearnPython
#let'sdoit



TUPLE: WHAT IS A TUPLE AND HOW IS IT DIFFERENT FROM A LIST?

Tuple is a type of data in Python that is similar to a list but cannot be changed once it is created.

Tuple is commonly used to store related datasets but doesn't need to be changed such as coordinates, month names, or syllables in a word.

In contrast to the list that uses elbow brackets [], tuple uses regular brackets ().

It also has the ability to store elements with different data types such as integers, floats, and strings.

Although tuple has disadvantages because it cannot be changed once made, but it provides an advantage in performance and memory over a list that requires more space.

Therefore, if you need a dataset that does not need to be changed, tuple is a better option.

Let's do it!

#Python #List #CodingTips #LearnPython
#let'sdoit



SET: HOW TO USE A SET IN PYTHON AND HOW IS IT DIFFERENT FROM LIST AND TUPLE?

Set is one of the types of data in Python, which is used to store a collection of objects non-sequentially and uniquely. In the set, the objects are not in a certain order, and each object appears only once.

In Python, the set is represented by a curly bracket sign {}. One of the differences between a set with a list and a tuple is that the set has no fixed order and the items inside the set are not accessible using the index.

You can use sets to perform mathematical operations such as unions, intersections, or differences between two sets. Make sure you understand how to use sets in Python to manipulate data more effectively.

Let's do it!.

#Python #Set #List #Tuple #DataTypes
#CodingTips #LearnPython #let'sdoit



DICTIONARY: HOW TO USE DICTIONARY IN PYTHON AND HOW IS IT DIFFERENT FROM LIST AND TUPLE?

Dictionary is a type of data collection in Python consisting of pairs of keys and values. It is used to store data in the form of key pairs and values that allow us to quickly access and manipulate them.

Dictionary differs from lists and tuples in that it uses keys to access values, not use indices. In Python, keys and values can be anything like integers, floats, strings, tuples, and so on, but key values must be unique.

Dictionary is very useful for manipulating complex data and can be used in many scenarios such as mapping names to age, mapping products to prices, and so on. Make sure you understand how to use the dictionary in Python to improve your programming capabilities.

Let's do it!

#Python #Dictionary #List #Tuple #DataTypes
#CodingTips #LearnPython #let'sdoit



IF-ELSE STATEMENTS:
HOW ARE IF-ELSE
STATEMENTS USED IN
PYTHON AND HOW DO
THEY ENABLE YOUR
CODE TO MAKE
DECISIONS?

If-else statements is a flow-controlling structure in Python that allows your code to make decisions based on a logical state. If-else statements test the logical state and, based on the results, will execute the relevant code block.

In programming, if-else statements are used to make decisions based on a number of conditions. For example, if the user's input value is greater than 10, we want to execute a block of code, and if not, we want to execute a different block of code. In situations like this, we can use if-else statements to test logical conditions and make decisions based on test results.

If-else statements are created in Python by writing the keyword "if", followed by the test conditions, followed by the code block to be executed if the condition is true. If the condition is not true, then Python will execute the existing code block under the "else" statement.

For example, here is a code that uses if-else statements to make decisions based on user input values:

x = int(input("Please enter a number: "))
if x > 10:
 print("The entered number is greater than
10.")
Else:
 print("The entered number is smaller or

print("The entered number is smaller or equal to 10.")

In the example above, we test the value of user input with if-else statements. If the entered number is greater than 10, then Python will print "The entered number is greater than 10.". Otherwise, Python will print "The entered number is smaller or equal to 10.".

In programming, if-else statements are a very useful tool for making decisions based on logical circumstances. For example, we can use it to test user input, detect errors in code, or make decisions in larger programming applications.

Let's go! We made him.

#Python IfElseStatements #CodingTips
#LearnPython #let'sdoit



LOOPS: HOW ARE FOR LOOPS AND WHILE LOOPS USED IN PYTHON TO REPEAT CODE?

For Loops and While Loops are two types of loops in Python that allow you to repeat the code multiple times without having to rewrite the same code. For loops are usually used to repeat something with a known amount, while while loops are used to repeat something until a certain condition is met.

In programming, loops are used to perform tasks such as reading and processing data from external sources, carrying out tasks that require some iteration, and making complex calculations. You can also combine loops with other flow control structures such as ifelse statements to make decisions inside the loop.

In Python, you can use for loops by repeating through each element in a data structure such as a list, tuple, or dictionary. You can also use temporary loops to repeat code until a condition is met, such as user input that meets certain criteria. Make sure you understand how to use for loops and while loops in Python to write more effective code.

Let's do it!

#Python #Looping #ForLoops #WhileLoops
#CodingTips #LearnPython #let'sdoit

# TUTORIAL / EXAMPLE I 0 0 p S

Loops are used in programming to repeat a block of code multiple times. Python provides two types of loops: "for" loops and "while" loops. In this tutorial, we will explore how to use both types of loops in Python.

The "for" loop in Python is used to iterate over a sequence (such as a list, tuple, or string) or other iterable objects. Here's the basic syntax:

```
for item in sequence:
    # Code block to be repeated
```

Let's see an example of using a "for" loop to iterate over a list:

```
fruits = ["apple", "banana", "orange"]

for fruit in fruits:
    print(fruit)
```

Output:

apple banana orange The range() function is often used in conjunction with the "for" loop to generate a sequence of numbers. It creates a range object that represents a sequence of numbers. Here's an example:

```
for num in range(1, 6):
    print(num)
```

Output:

The "while" loop is used to repeatedly execute a block of code as long as a certain condition is true. Here's the basic syntax:

```
while condition:
    # Code block to be repeated
```

Let's see an example of using a "while" loop to print numbers from 1 to 5:

```
num = 1
while num <= 5:
    print(num)
    num += 1</pre>
```

#### TUTORIAL / EXAMPLE

Output:

Python provides loop control statements to modify the behavior of loops. The three main control statements are break, continue, and pass.

- The break statement terminates the loop prematurely.
- The continue statement skips the rest of the code in the current iteration and moves to the next iteration.
- The pass statement is used as a placeholder when you want an empty block of code.

Python allows you to have loops within loops, known as nested loops. This is useful when you need to iterate over multiple sequences or perform complex iterations.

In this tutorial, you've learned how to use "for" and "while" loops in Python.

Loops are powerful constructs that allow you to repeat code and automate repetitive tasks.



FUNCTIONS: HOW TO DEFINE AND CALL FUNCTIONS IN PYTHON AND HOW DO THEY FACILITATE CODE REPETITION?

A function is a block of code that can be called upon to perform a specific task. You are able to create your own functions in Python and can call them repeatedly in your program. This is called code reuse (code reusability).

To create a function in Python, use the keyword "def" followed by the name of the function and the parameter you want. Then, in the body function, you can determine the instructions you want to carry out.

To call a function, use only the name of a previously created function followed by the required parameter (if any). In python, parameters and return statements are not mandatory. A return statement can also return more than one value and be returned in the form of a tuple.

Make sure you are well versed in creating and manipulating functions so that your code is more modular and structured and avoids data repetition.

Let's do it!

#Python #Function #CodingTips
#LearnPython #let'sdoit



How to read user input and output output in Python?

You can use the input() function to read the user input and print() function to display the output. Here is an example of its use:

```
name = input("Please enter your name: ")
age = input("Please enter your age: ")
print(f"Hi, my name is {name} and I'm {year}.")
```

In this example, we use the input function() to read the user's input in text form and store it in two variables, namely name and age. Then, we use the print() function to output that contains the values entered by the user.

Now you already know how to read user input and output output in Python!

Let's go! We made him.

#Python #InputOutput #CodingTips #LearnPython #let'sdoit



How to read and output from a file in Python? Using Python, we can easily read and write data into files.

To read data from a file, we can use the open() function to open the file, then use the read() function to read its contents. Once done, don't forget to close the file using the close() function.

To write data into a file anyway, the steps are almost the same. We open the file by using open() and write its contents using the write() function. Finally, don't forget to close the file using close().

In Python programming, we can also use file handling concepts such as "context manager" by using the with statement. This allows files to close automatically after use, and keeps our code cleaner and safer.

So, let's take advantage of using inputoutput files in Python to read and write data easily and efficiently!

Let's do it!

#Python #InputOutput #FilesIO #CodingTips #LearnPython #let'sdoit



USING PYTHON MODULE: HOW TO USE MODULES AVAILABLE IN PYTHON SUCH AS MATH AND TIME?

Let's talk about using modules in Python such as math and time. These modules provide useful functions to facilitate programming work.

The math module provides various mathematical functions such as rounding, square root, logarithm, and more. Using the math module, we can easily carry out complex mathematical operations.

While the time module provides time-related functions such as getting the current date and time, measuring the time distance, and controlling time intervals and pauses. This module is very useful in handling tasks that involve time and schedule.

To use these modules, we need to import them into our code using the 'import' statement. For example, to use the math module, we can use 'import math' and then use the functions provided in the module.



Apart from the above modules, there are many more modules that come with Python. You can see more at the https://docs.python.org/3/py-modindex.html link

Let's do it!

#Python #PythonModule #Math #Time
#CodingTips #LearnPython #let'sdoit



IMPORTING MODULES: HOW TO IMPORT SELF-WRITTEN MODULES IN PYTHON CODE?

In some larger Python projects, you may need to split your code into several different files to make it easier to maintain and manage code. One way to do this is to write your own modules and import them into your main code.

To import your modules, you just need to use the 'import' statement.

Let's say you have a file named 'mymodule.py' that contains the 'my\_function' function. To import this function into your other Python files, you can add the following statement at the beginning of your file:

• • •

import mymodule

mymodule.my\_function()



If you only need certain functions from your module, you can import only those functions by using the following statement:

. . .

from mymodule import my\_function

my\_function()

• • •

Let's do it!

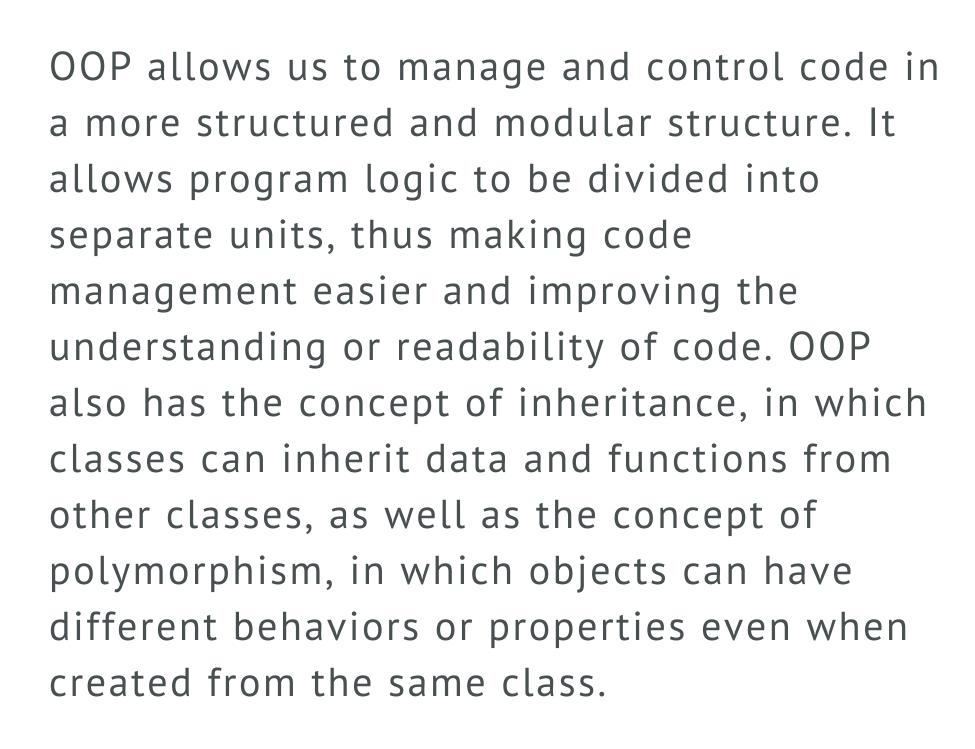
#Python #ImportModule #CodingTips
#LearnPython #let'sdoit



WHAT IS OOP AND HOW IS IT USED IN PYTHON?

OOP (Object-Oriented Programming) or Object-Oriented Programming is a programming paradigm that uses objects as the main unit for managing and handling data and related functions. In OOP, an object is the embodiment of a class, where a class is a blueprint or template that defines the properties (atribute) and behavior (behaviour) of an object.

In Python, OOP is used using the concepts of classes and objects. We can define classes by combining data (properties) and functions (behavior) in one entity. Objects can then be created from that class, and each object will have the same properties and behavior as defined in the class.



Let's do it!

#Python #OOP #CodingTips #LearnPython
#let'sdoit



CLASS DEFINITION IN PYTHON: HOW TO CREATE A CLASS IN PYTHON AND HOW IT HELPS IN OOP.

Mhat is Class Definition in Python? 🔊

In Python, the definition of a class is a way to create an entity that possesses certain properties (atribute) and behavior (behaviour). Classes serve as blueprints or templates used to create objects. In OOP (Object-Oriented Programming), classes are an important part of managing and controlling code in a more structured and structured manner.

→ How to Create a Class in Python? 🔁

To create a class in Python, use the keyword 'class' followed by the class name we want. See the following example:

Car class:
 def \_\_init\_\_(self, brand, year):
 self.brand = brand
 self.year = year

def info(self):
 print(f"Car {self.brand} year {self.year}")

In the example above, we create a 'Car' class that has two attributes, namely 'brand' and 'year'. We also define the 'info' function that prints information about the Car.

How Do Classes Help in OOP?

Classes help in OOP by allowing us to create objects based on a predetermined blueprint. When we create an object from a class, it will have attributes and functions that have been defined in that class.

Using classes, we can group data and functions that have a specific relationship into one structured entity. This makes the code more structured, manageable, and reusable.

For example, with the 'Cars' class above, we can create several different Car objects with different attributes. We can also use the functions that have been defined in the class to perform certain actions on those objects.



By learning the definition of classes in Python, you will be able to master OOP and leverage it to develop more structured, modular, and manageable code. Happy study!

Let's go! We made him.

#Python #Kelas #Class #OOP #CodingTips
#BelajarPython #jomkitabuatdia



ENCAPSULATION AND ABSTRACTION: HOW ENCAPSULATION AND ABSTRACTION USED IN PYTHON'S OOP?

Encapsulation and Abstraction in OOP for Python!

In Object-Oriented Programming (OOP), encapsulation and abstraction are important concepts that help in building structured, organized, and easy-to-read code. Let's see how these two concepts are used in OOP for Python.

#### Encapsulation:

Encapsulation is a concept in which attributes and functions in a class are grouped together and accessed through an interface (not referring to a GUI or Graphical User Interface) defined by a programmer.

In Python, encapsulation can be achieved by using the identifier naming convention by adding a double underscore (double underscore) to the name of the attribute or function.

```
Example of capsulation in Python:
Car class:
  def __init__(self, color, year):
     self.__year = color
     self.__year = year
  def get_color(self):
     return self. color
  def get_year(self):
     return self.__year
  def set_color(self, color):
     self.__color = color
car = Car("Red", 2022)
print(car.get_warna()) # Output: Red
print(car.get_year()) # Output: 2022
kereta.set_color("Blue")
print(kereta.get_color()) # Output: Blue
, , ,
```

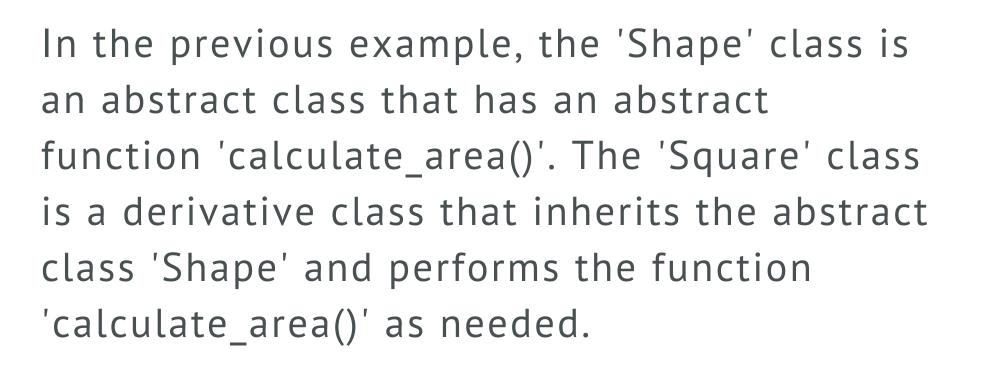
In the example above, the 'color' and 'year' attributes are encapsulated in the 'Cars' class using a double underline. To access and change those attributes, we use the 'get\_warna()' and 'set\_warna()' functions.



Abstraction is a concept in which we hide the internal details of an object and display only relevant and useful features. In Python, abstraction can be achieved by using abstract classes and abstract functions.

### DAY 29

111



Let's do it!

#Python #Kelas #Encapsulation #Abstraction
#OOP #CodingTips #LearnPython #let'sdoit



INHERITANCE AND
POLYMORPHISM: HOW IS
INHERITANCE AND
POLYMORPHISM USED IN
PYTHON'S OOP?

Inheritance and Polymorphism in OOP for Python!

In Object-Oriented Programming (OOP), inheritance and polymorphism are powerful concepts for managing and controlling code more efficiently. Let's see how these two concepts are used in OOP for Python.

#### Inheritance:

Inheritance is a concept in which a class can inherit attributes and functions from another class.

The inheriting class is called a child class or a subclass, whereas the one that gives the inheritance is called a master class or a superclass.

Using inheritance, we can gather and group traits and behaviors from the parent class to the child's class.

A simple example of inheritance in Python:

```
Animal class:

def __init__(self, name):

self.name = name

def sound(self):

pass

Class Cat(Animals):

def sound(self):

print("Meow!")

Animal = Cat("Kitty")

print(Animals.name) # Output: Kitty

Animal.sound() # Output: Meow!
```

In the example above, the 'Cat' class inherits the 'name' attribute and the 'voice' function from the 'Animals' class. We can use the attributes and functions inherited from the 'Animals' class on the 'Cat' object.



Polymorphism is a concept in which objects from different classes can be accessed through the same interface.

In polymorphism, we can use the same function in different ways on different objects.

This allows us to handle objects in general without having to know the specific type of object.

A simple example of polymorphism in Python:

class Animal:

def sound(self):

pass

Class Cat(Animal):

def sound(self):

print("Meow!")

Class Dog(Animal):

def sound(self):

print("Woof!")

def animal\_sound(Animal):
 Animal.sound()

### DAY 30

```
cat = Cat()
dog = Dog()
```

animal\_sound(cat) # Output: Meow!
animal\_sound(dog) # Output: Woof!

In the example above, the 'animal\_sound()' function accepts objects of any class that have the function 'sound()'. We can call this function with different 'Cat' and 'Dog' objects, and the 'sound()' function will be called in accordance with that type of object.

Let's do it!

#Python #Class #Inheritance #Polymorphism
#OOP #CodingTips #LearnPython #let'sdoit

#### TUTORIAL / EXAMPLE

### 0 O P

A class is a blueprint that defines the structure and behavior of an object. An object is an instance of a class.

Inheritance allows one class to inherit the attributes and methods of another class. This facilitates code reuse and helps in creating more specialized classes.

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def introduce(self):
        return f"My name is {self.name} and I am {self.age} years old."

# Creating an object of class Person
john = Person("John Doe", 30)

# Accessing attributes
print(john.name) # Output: John Doe

# Calling a method
print(john.introduce()) # Output: My name is John Doe and I am 30 years old.
```

The \_\_\_init\_\_\_ method is a special method (also known as the constructor) that is called when a new object is created. It initializes the object's attributes.

Methods are functions defined within a class. They define the behavior of objects created from the class.

Encapsulation is the concept of bundling data (attributes) and methods that operate on the data within a single unit (a class).

Polymorphism allows objects to be treated as instances of their superclass. This means that a method defined in the superclass can be overridden in a subclass to provide specific behavior.

## REFERENCES

Stack overflow developer survey 2021. Stack Overflow. (n.d.). Retrieved April 13, 2023, from https://insights.stackoverflow.com/survey/2021 #section-most-popular-technologies-programming-scripting-and-markup-languages

Google. (n.d.). Google/Python-Fire: Python Fire is a library for automatically generating command line interfaces (clis) from absolutely any python object. GitHub. Retrieved April 13, 2023, from

https://opensource.google/projects/python-fire

Python 2.7.18 documentation. (n.d.). https://docs.python.org/2/index.html

What's New In Python 3.0. (n.d.). Python Documentation.

https://docs.python.org/3/whatsnew/3.0.html

Medium. (n.d.). Medium.

https://towardsdatascience.com/5-industries-being-transformed-by-python-83333a09e11f

Medium. (n.d.). Medium.

https://towardsdatascience.com/the-importance-of-following-pep-8-style-guide-for-python-code-c2cacb8c7f96



## REFERENCES

Python Data Types (With Examples). (n.d.). https://www.programiz.com/python-programming/variables-datatypes

Python, R. (2023, September 7). Conditional Statements in Python. https://realpython.com/python-conditional-statements/

Pal, S. (2017, October 18). Python Loops Tutorial.

https://www.datacamp.com/community/tutorials/loops-python-tutorial

Python Functions. (2023, May 4). GeeksforGeeks. https://www.geeksforgeeks.org/functions-in-python/

9. Classes. (n.d.). Python Documentation. https://docs.python.org/3/tutorial/classes.html

