

หลักสูตร

# วิศวกรความมั่นคงปลอดภัย Security Engineer

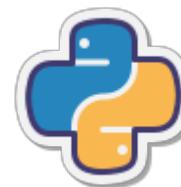
ภายใต้โครงการเส้นทางสู่วิศวกรความมั่นคงปลอดภัย สำหรับนักศึกษาจบใหม่สู่การทำงานในภาคอุตสาหกรรม



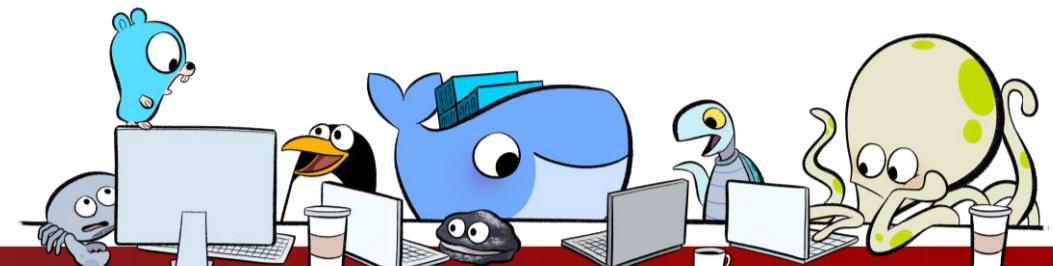
Designed by [catalyststuff](#)  
on Freepik

# The Code Awakens

Hello, Human. Let's Code.



**Maykin W., PhD**  
(Tue) November 4, 2025



## អាគ្នេត ធម្មតា និងកម្រិតសិក្សា Security Engineer

ការណានីករដ្ឋបាលការងារសៀវភៅកម្រិតសិក្សា Security Engineer ដែលរួចរាល់នូវការងារសៀវភៅកម្រិតសិក្សា និងកម្រិតសិក្សា និងកម្រិតសិក្សា

slido

# Coding & U ?

How do you feel ?

# Tracking your status

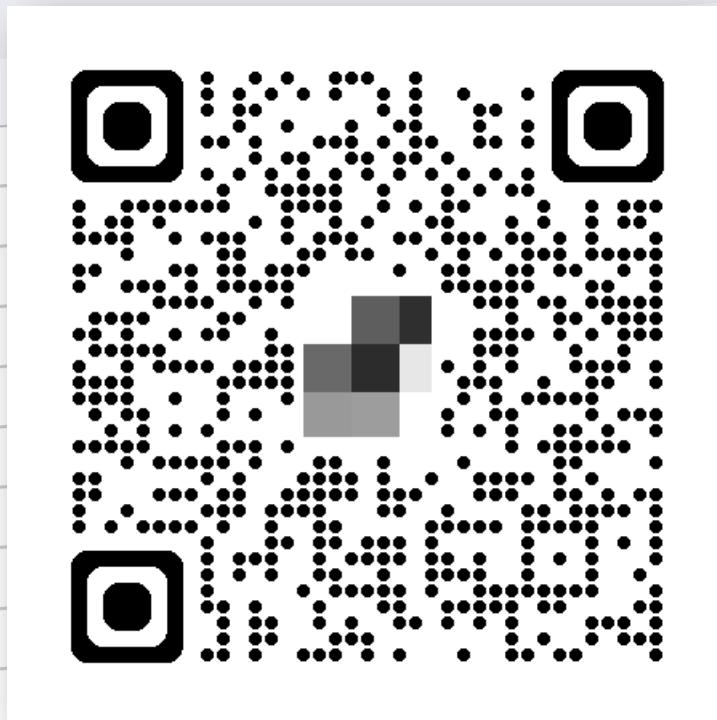


01:54

## Assignment Tracking

X View only

A	B	C	D
No.		Status	Remark
1	1		
2	2		
3	3		
4	4		
5	5		
6	6		
7	7		
8	8		
9	9		
10	10		
11			



# Submit your assignment



Let's Code! - Nov 4, 2025

Simple assignment for security engineers.

maykin@owasp.org [Switch account](#)



The name, email, and photo associated with your Google account will be recorded when you upload files and submit this form

Any files that are uploaded will be shared outside of the organization they belong to.

\* Indicates required question

No. (เลขที่) \*

Your answer



Full name (Eng) \*

Your answer

ชื่อ-สกุล (ไทย) \*

Your answer

Anything new you discovered outside the course? \*

Your answer

Let's see it. (URL) \*

Your answer

Screenshot \*

Upload 1 supported file: image. Max 10 MB.

[Add file](#)

[Submit](#)

[Clear form](#)

Never submit passwords through Google Forms.

This form was created inside of IEEE. - [Contact form owner](#)

Does this form look suspicious? [Report](#)

Google Forms

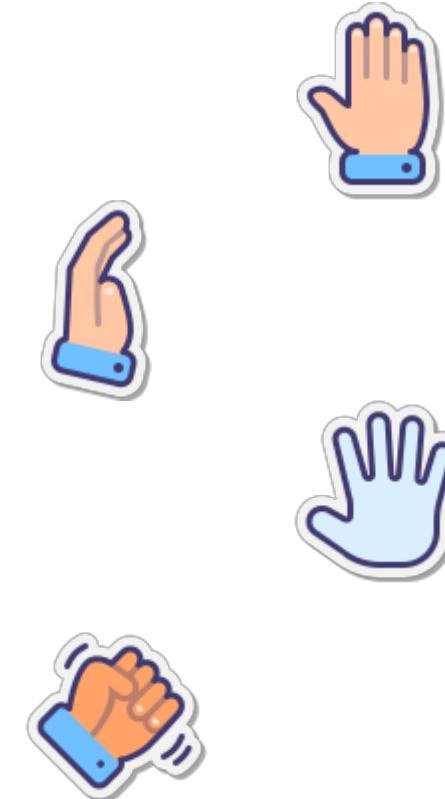
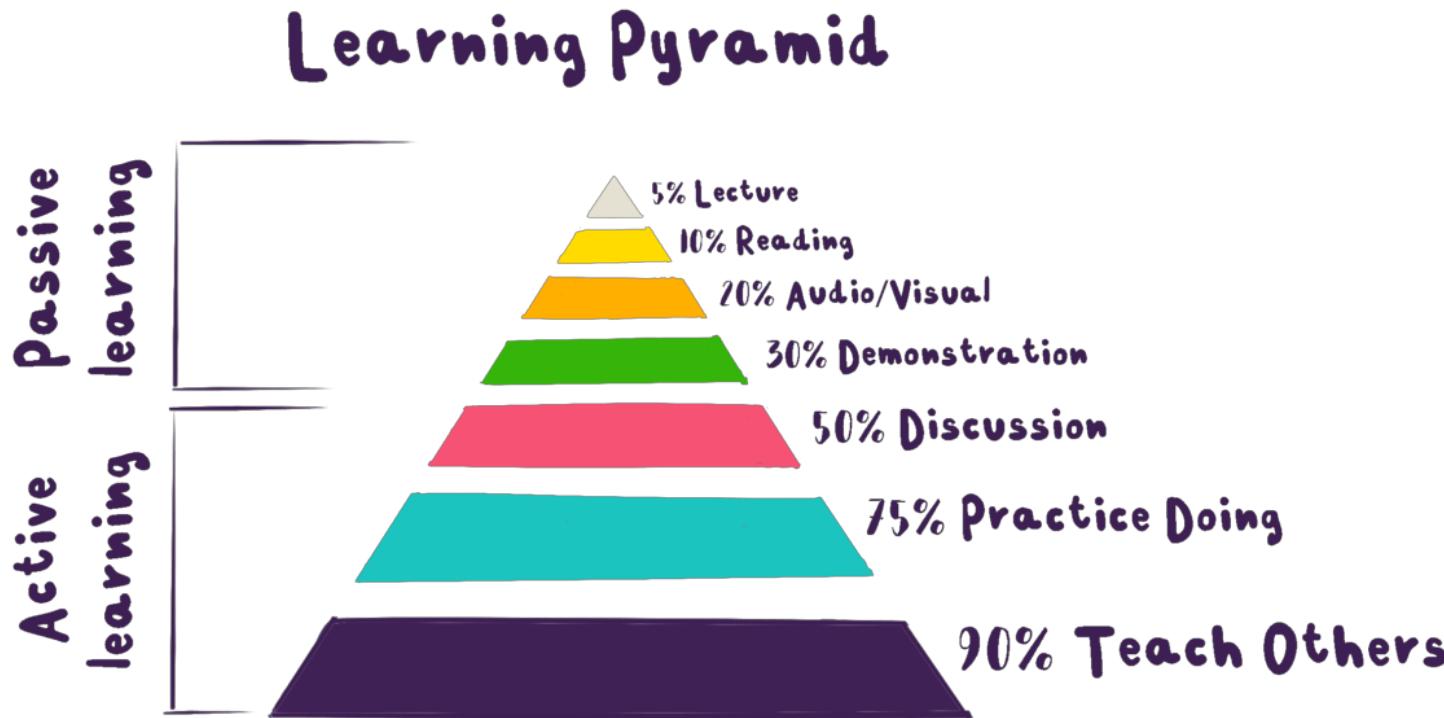


# Course Outline

- Print Objects and Data Types
- Input
- Input and Operations
- Strings and Operations
- Arithmetic Operations
- Comparisons and Logical Operations
- If Statements
- Loops
- Data Structures (List, Dictionary)
- Python void Function
- Return function
- Exceptions
- Classes, Objects and Methods
- Inheritance
- Modules

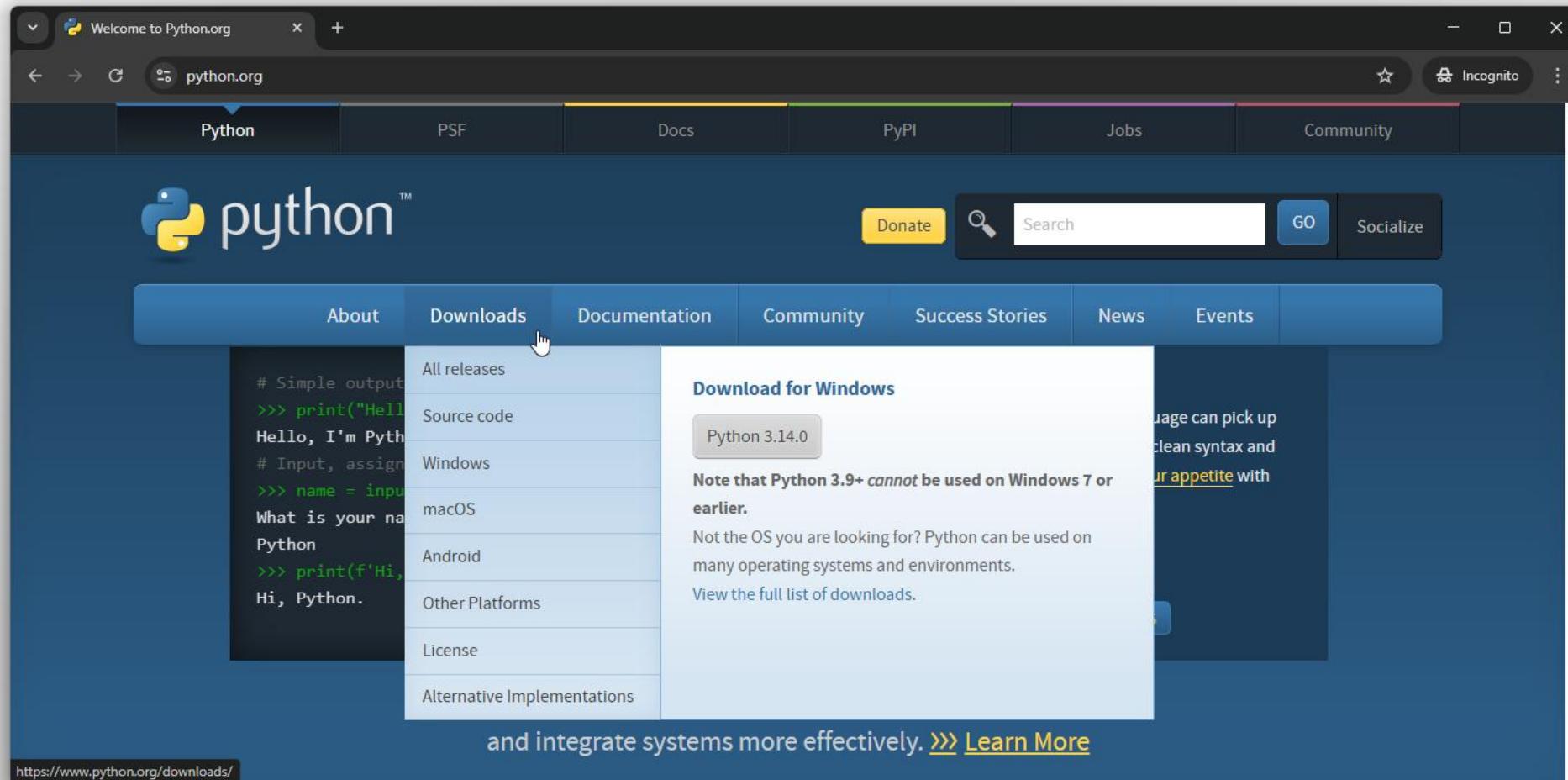


# Housekeeping



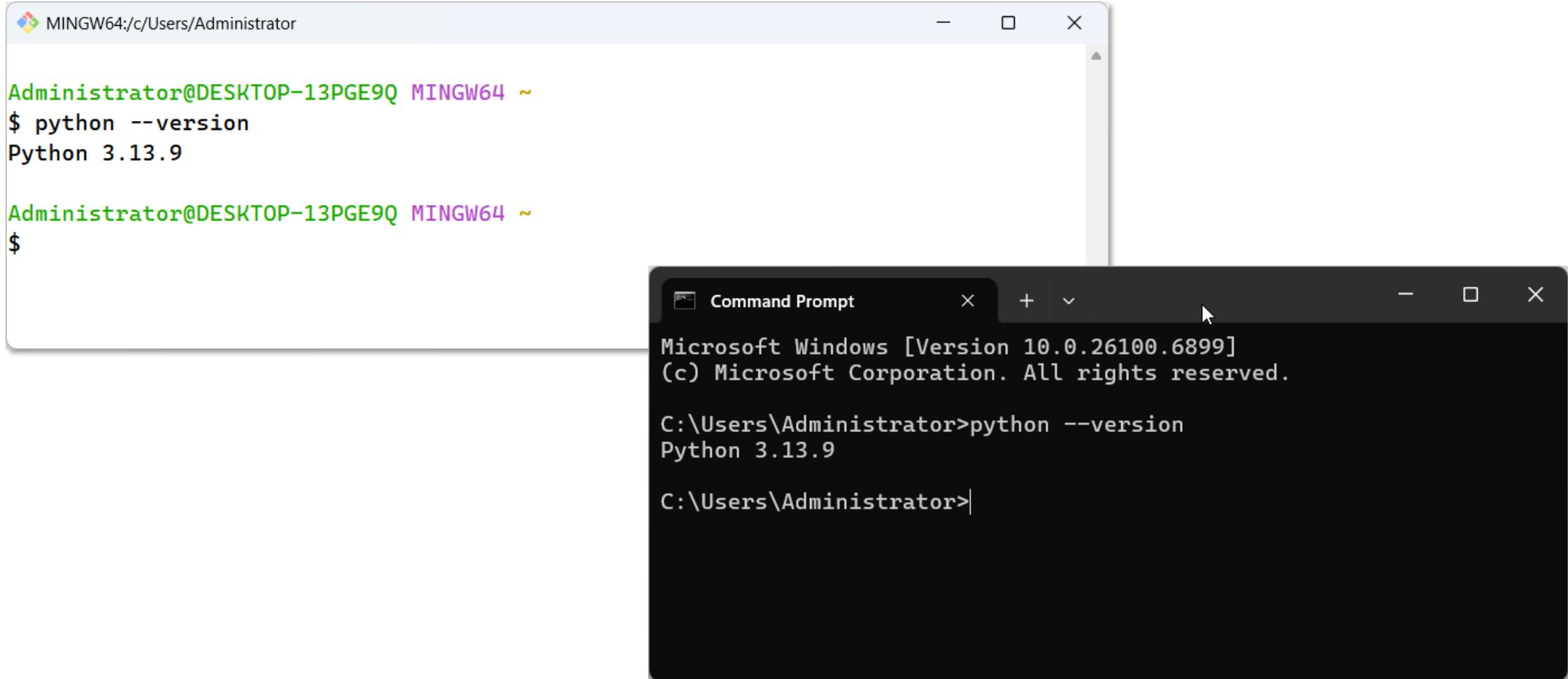
## អាគកសូត ធម្មតរគារមំបែកង់លួចកាយ Security Engineer

ការណាំក្នុងការសេវាការមំបែកង់លួចកាយ ដោយរបៀបកើតការងារ និងការរំលែកការងារ នៃការងារក្នុងការអនុវត្តន៍យោង



## หลักสูตร วิศวกรความมั่นคงปลอดภัย Security Engineer

ภายใต้โครงการเส้นทางสู่วิศวกรความมั่นคงปลอดภัย สำหรับบัณฑิตจบใหม่สู่การทำงานในภาคอุตสาหกรรม



The image displays two side-by-side command-line windows. The top window is a terminal session titled 'MINGW64:/c/Users/Administrator' running on a Windows system. It shows the command '\$ python --version' being run, followed by the output 'Python 3.13.9'. The bottom window is a 'Command Prompt' window titled 'Command Prompt' running on a Windows system. It shows the command 'C:\Users\Administrator>python --version' being run, followed by the output 'Python 3.13.9'. Both windows are dark-themed.

```
Administrator@DESKTOP-13PGE9Q MINGW64 ~
$ python --version
Python 3.13.9

Administrator@DESKTOP-13PGE9Q MINGW64 ~
$
```

```
Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>python --version
Python 3.13.9

C:\Users\Administrator>
```

## หลักสูตร วิศวกรความมั่นคงปลอดภัย Security Engineer

ภายใต้โครงการเส้นทางสู่วิศวกรความมั่นคงปลอดภัย สำหรับบุคคลที่สนใจในการทำงานในภาคอุตสาหกรรม

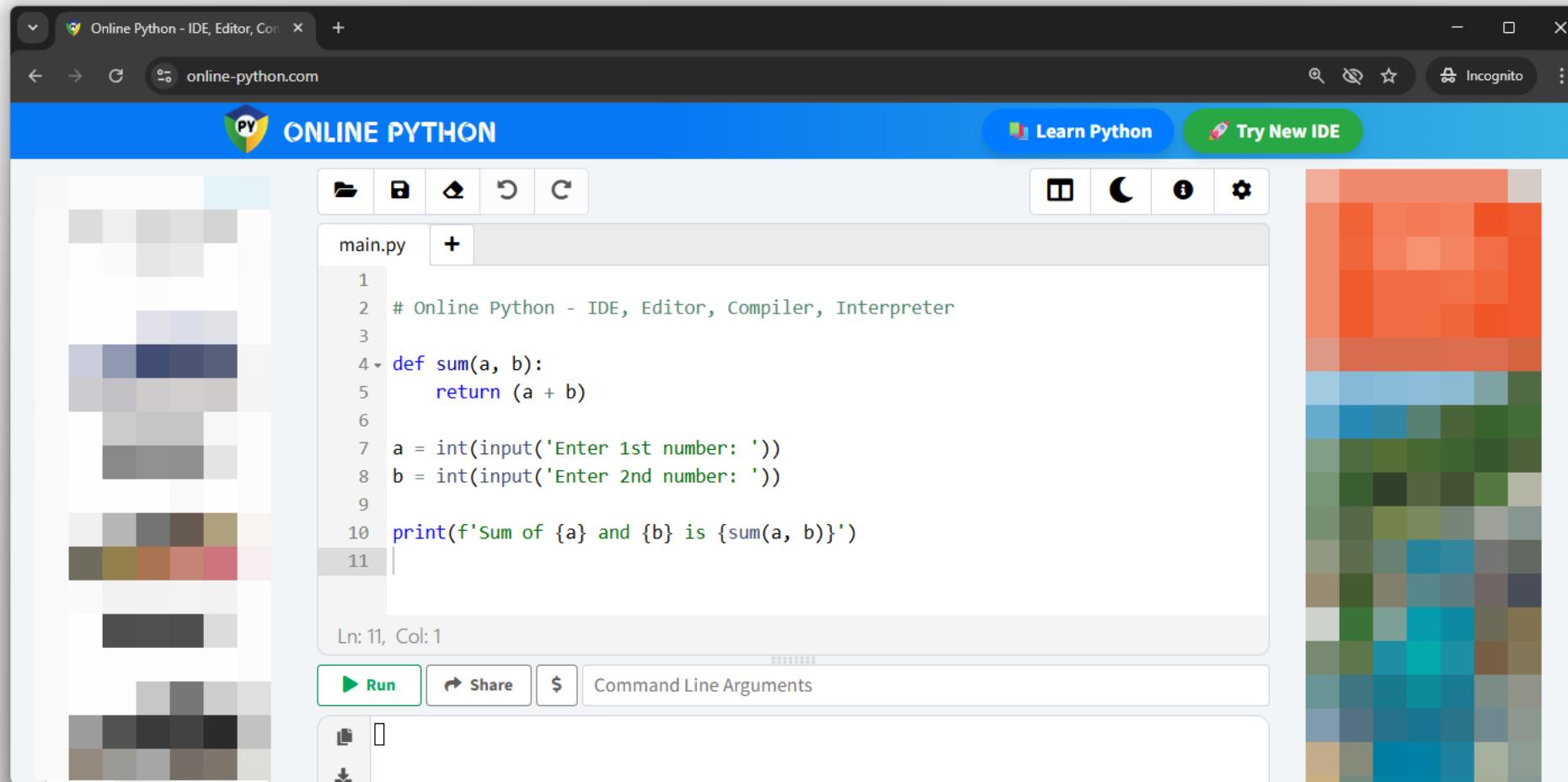
The screenshot shows the official download page for Visual Studio Code at [code.visualstudio.com/download](https://code.visualstudio.com/download). The page features a large "Download Visual Studio Code" heading and a subtext stating "Free and built on open source. Integrated Git, debugging and extensions." Below this, there are three main sections for different operating systems:

- Windows:** Shows the Windows logo and a "Windows" button with "Windows 10, 11" underneath. Below it, there are links for "User Installer" (x64, Arm64), "System Installer" (x64, Arm64), ".zip" (x64, Arm64), and "CLI" (x64, Arm64).
- Linux:** Shows the Tux logo and buttons for ".deb" (Debian, Ubuntu) and ".rpm" (Red Hat, Fedora, SUSE). Below it, there are links for ".deb" (x64, Arm32, Arm64), ".rpm" (x64, Arm32, Arm64), ".tar.gz" (x64, Arm32, Arm64), "Snap" (Snap Store), and "CLI" (x64, Arm32, Arm64).
- Mac:** Shows the Apple logo and a "Mac" button with "macOS 11.0+" underneath. Below it, there are links for ".zip" (Intel chip, Apple silicon, Universal), "CLI" (Intel chip, Apple silicon), and "CLI" (x64, Arm32, Arm64).

At the bottom left, there is a link to the full URL: <https://code.visualstudio.com/Download>. The top right corner of the browser window shows a "Download" button with a mouse cursor hovering over it.

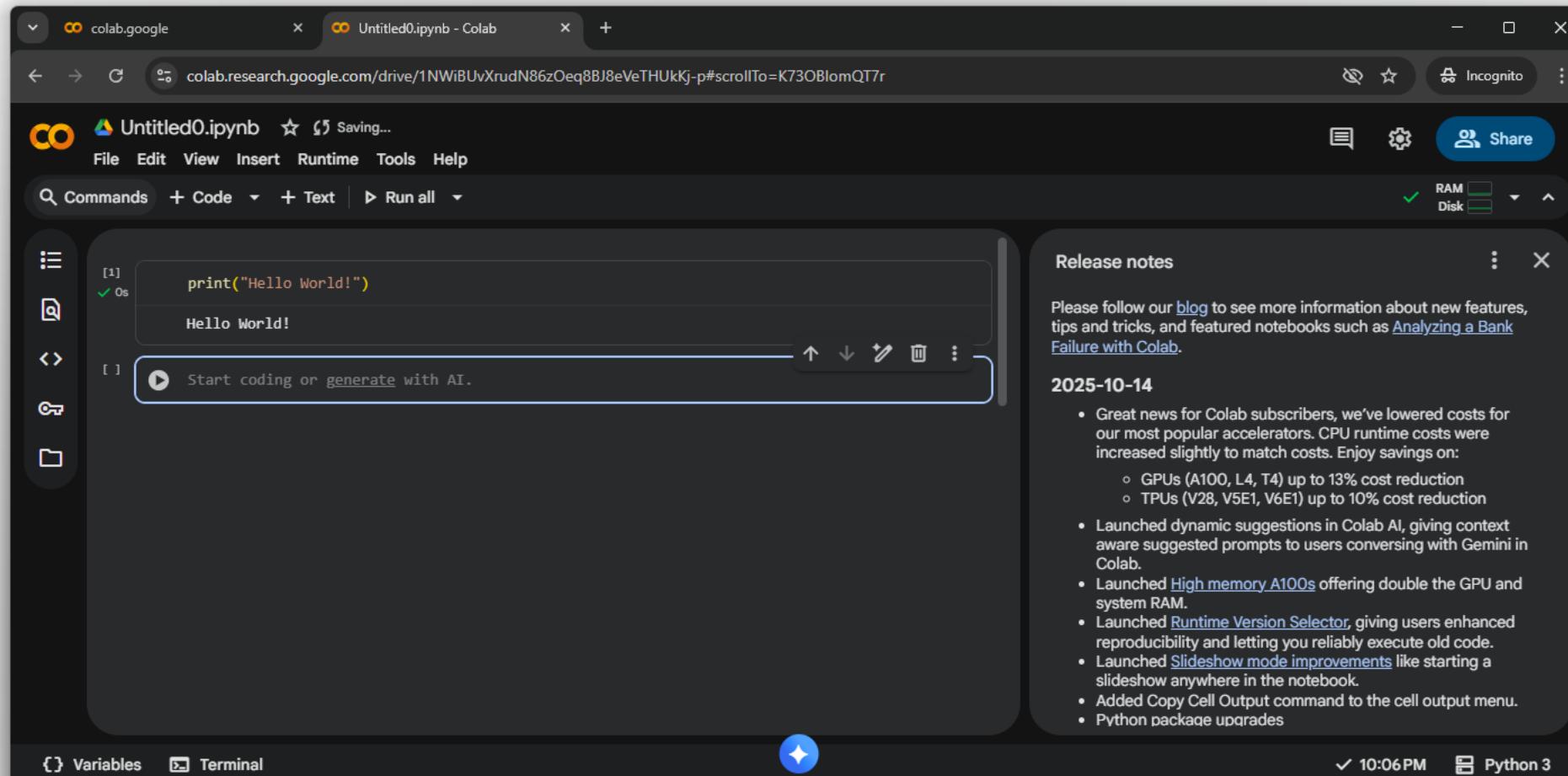
## អាគកសូត្រ ធម្មតាអនុញ្ញន៍កម្រិតខ្ពស់ Security Engineer

ការបង្កើតគម្រោងនូវការរាយការណ៍របស់អ្នកសូត្រ ដែលមានភាពខ្ពស់នៅក្នុងការងាររបស់អ្នក។



## អាគកសូត្រ ធម្មតាអនុញ្ញាតកម្មភ័យ Security Engineer

ការងារក្នុងកម្មវិធីសូត្រ ធម្មតាអនុញ្ញាតកម្មភ័យ ដែលរួចរាល់បានការងារក្នុងកម្មភ័យ ដែលត្រូវបានរាយការណ៍នា



# Print Objects and Data Types

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



# Print Objects and Data Types

- **myPortNo = 22**
- **print(myPortNo)**



Each word, except the first,  
starts with a capital letter

- **MyPortNo = 22**
- **print(MyPortNo)**



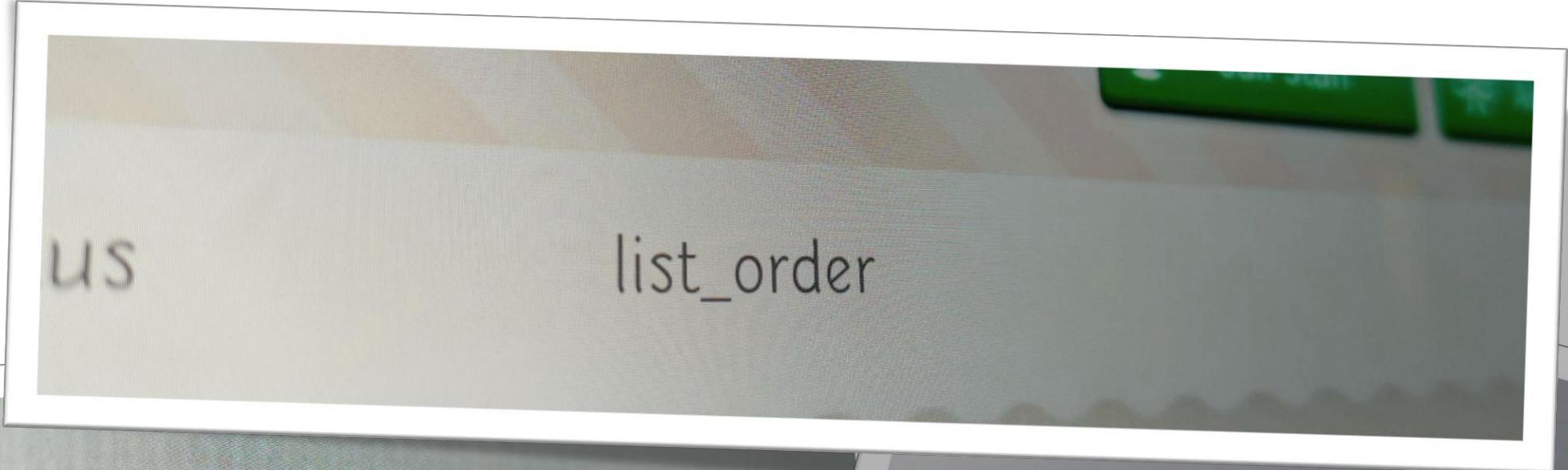
Each word starts  
with a capital letter

- **my\_port\_no = 22**
- **print(my\_port\_no)**



Each word is separated by  
an underscore character

# Print Objects and Data Types



btn\_back

Bill

alertCheckbill

# Print Objects and Data Types

- **port\_no = 22**
- **print(type(port\_no))**



- **port\_no = 22.0**
- **print(type(port\_no))**



- **port\_no = "22"**
- **print(type(port\_no))**



# Print Objects and Data Types



# Print Objects and Data Types

- **logic** = true
- print(type(logic))

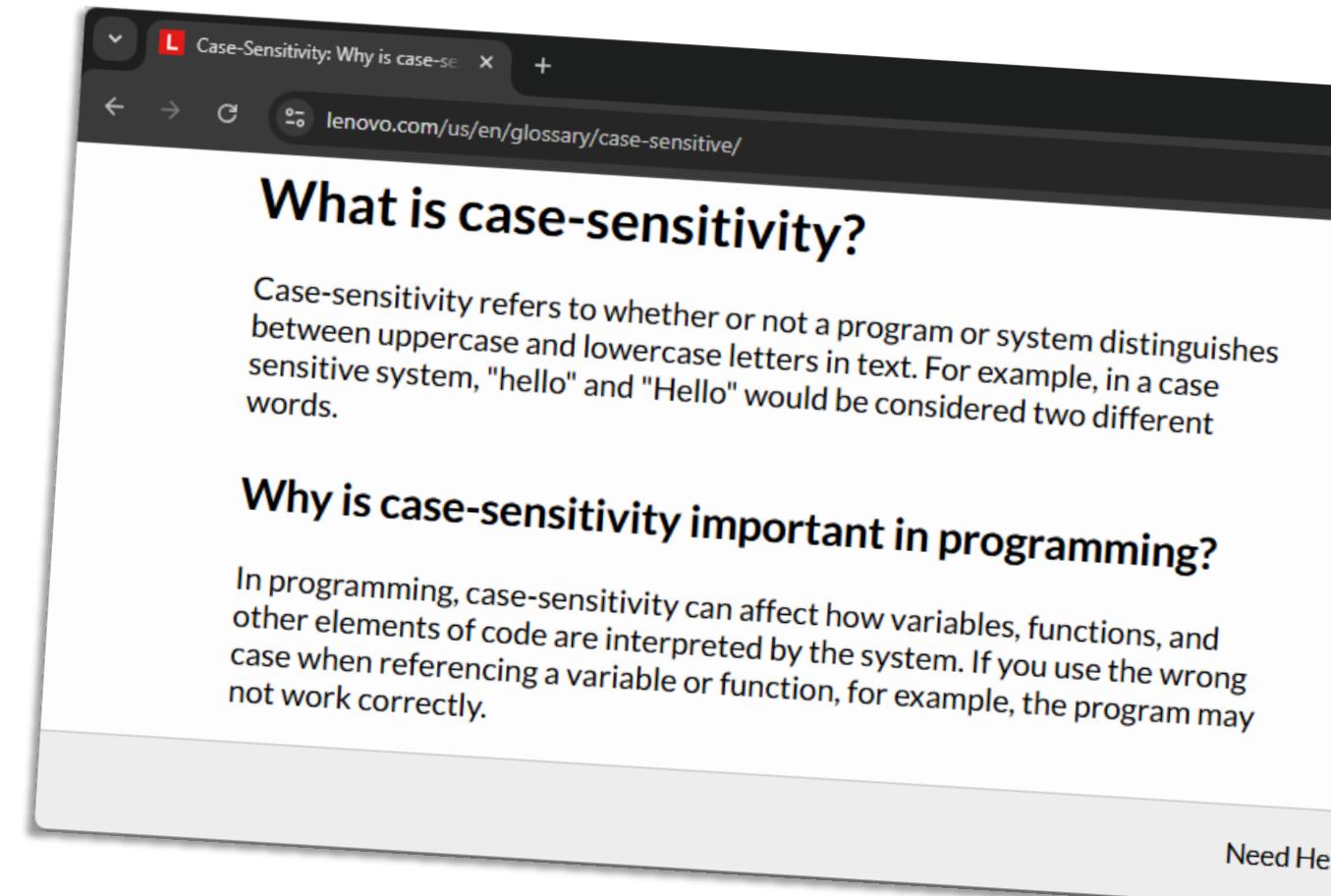


- **logic** = True
- print(type(logic))



# Print Objects and Data Types

- **logic** = true
- print(type(logic))
- **logic** = True
- print(type(logic))



Need He

# Print Objects and Data Types

- **logic** = true
- print(type(logic))
  
- **logic** = True
- print(type(logic))
  
- **logic** = "False"
- print(type(logic))



# Print Objects and Data Types

- `logic = True`
- `print(logic)`



- `print(not logic)`



# Print Objects and Data Types

- `first_name = "Maykin"`
- `last_name = "Warasart"`
- `print(first_name + last_name)`



Python Strings

- Python Strings
- Slicing Strings
- Modify Strings
- Concatenate Strings

## String Concatenation

To concatenate, or combine, two strings you can use the `+` operator.

[https://www.w3schools.com/python/python\\_strings\\_concatenate.asp](https://www.w3schools.com/python/python_strings_concatenate.asp)

# Print Objects and Data Types

- `full_name = "Maykin W."`
- `lucky_number = 4`
- `print(full_name + lucky_number)`



# Input

- `input()`

- `ip = input()`
- `print(ip)`



- `ip = input("Enter IP Address : ")`
- `print("IP Address : " + ip)`



# Input and Operations

- `your_age = input("How old are you? : ")`
- `year_of_birth = 2025 - int(your_age)`
- `print("Year of Birth : " + str(year_of_birth))`

- How old are you? : Twenty



# Strings and Operations

- `last_name = "Warasart"`
- `print(last_name.upper())`
- `print(last_name.find("s"))`
- `print(last_name.find("a"))`



# Strings and Operations

- `full_name = "Maykin Warazart"`
- `print(full_name)`
- `print(full_name.replace("z", "s"))`
- `print(full_name.find("art"))`
- `print(full_name.find("sart"))`



# Arithmetic Operations

• <b>Addition</b>	<code>+</code>	$x + y$
• <b>Subtraction</b>	<code>-</code>	$x - y$
• <b>Multiplication</b>	<code>*</code>	$x * y$
• <b>Division</b>	<code>/</code>	$x / y$
• <b>Floor Division</b>	<code>//</code>	$x // y$
• <b>Modulo</b>	<code>%</code>	$x \% y$
• <b>Power</b>	<code>**</code>	$x ** y$

# Arithmetic Operations

- `x = 7`
- `y = 3`
- `print("x + y = ", x + y)`
- `print("x - y = ", x - y)`
- `print("x * y = ", x * y)`
- `print("x / y = ", x / y)`
- `print("x // y = ", x // y)`
- `print("x % y = ", x % y)`
- `print("x ** y = ", x ** y)`



# Comparisons and Logical Operations

• Less than	<	x < y
• Less than or equal	<=	x <= y
• Greater than	>	x > y
• Greater than or equal	>=	x >= y
• Equal	==	x == y
• Not equal	!=	x != y
• Object identity	is	x is y
• Negated object identity	is not	x is not y

# Comparisons and Logical Operations

- `print("4 == 4 : ", 4 == 4)`
- `print("2 < 4 : ", 2 < 4)`
- `x = 10`
- `y = 8`
- `print("x != y : ", x != y)`
- `print("x - y == 2 : ", x - y == 2)`



# Comparisons and Logical Operations

- **and**       $x \text{ and } y$       **True** if  $x$  and  $y$  are true, else **False**
- **or**         $x \text{ or } y$       **True** if  $x$  or  $y$  are true, else **False**
- **not**        $\text{not } x$       **True** if  $x$  is False, else **True**

# Comparisons and Logical Operations

- `print(2 == 2 or 2 < 4)`
- `print(2 == 2 and 2 > 4)`
- `x = True`
- `print(not x)`



# if Statement

- `x = 5`
- `if x > 4:`  
 `print("x > 4")`
- `x = 5`
- `if x > 4:`  
 `print("x > 4")`
- `else:`  
 `print("x <= 4")`

# if Statement

- `number_1 = float(input("1st Number : "))`
  - `number_2 = float(input("2nd Number : "))`
  - `sub = number_1 - number_2`
  - `print("Result : " + str(sub))`
  
  - `if sub < 0:`  
    `print("Negative Number")`
  - `else:`  
    `print("Positive Number")`
- “Zero is the **boundary** between positive and negative numbers.”

# if Statement

- `number_1 = float(input("1st Number : "))`
- `number_2 = float(input("2nd Number : "))`
- `sub = number_1 - number_2`
- `print("Result : " + str(sub))`
  
- `if sub < 0:`  
    `print("Negative Number")`
- `else:`  
    `print("Non-Negative Number")`

# if Statement

- `x = 5`
- `if x == 5:`  
 `print("x = 5")`
- `elif x == 10:`  
 `print("x = 10")`
- `else:`  
 `print("x is not 5, 10")`

# Loops (while)

- `i = 0`
- `while i < 5:`  
 `print(i)`  
 `i=i+1`

# Loops (for)

- temperatures = [5, 10, 15, 20]
- **for** item in temperatures:  
    print(item)
  
- numbers = range(5)
- **for** num in numbers:  
    print(num)
- numbers = range(2,15,3)
- **for** num in numbers:  
    print(num)

# Data Structures (list)

- `dataList = ["Security", 500, 123.45, False]`
- `print(dataList)`
- `print(dataList[0])`
- `print(dataList[0:2])`
- `print(len(dataList))`
- `print(dataList[:2])`



# Data Structures (list)

- `dataList = ["Security", 500, 123.45, False]`
- `dataList.remove("Security")`
- `print(dataList)`



- `dataList.insert(1, "Cybersecurity")`
- `print(dataList)`



# Data Structures (list)

- `dataList = ["Security", 500, 123.45, False]`
- `print("Security" in dataList)`
- `print("Securities" in dataList)`



- `dataList.clear()`
- `print(dataList)`



# Data Structures (Dictionary)

- users = {"Alice": 40, "Bob": 30, "Eve": 25}
- print(users)
- print(users['Bob'])



- users = {"Alice": 40, "Bob": 30, "Eve": 25}
- print(users)
- users['Alice'] = 41
- print(users)
- users['Tom'] = 20
- print(users)



# void Function

- `def my_function():  
 print("Hi, my function.")`
  - `my_function()`
- 
- `def sub_func(num1, num2):  
 print("Result = " + str(num1-num2))`
  - `sub_func(5, 4)`

# Return Functions

- ```
def sub_func(num1, num2):  
    return num1 - num2
```
  - ```
print(sub_func(4, 5))
```
- 
- ```
def add_func(num1, num2):  
    return num1 + num2
```
  - ```
print(add_func(4, 5))
```

# Exceptions

- **try:** Run this code
- **except:** Execute this code  
when there is an exception
- **else:** No exceptions?  
Run this code.
- **finally:** Always run this code

# Exceptions

- **try:**

```
num = int(input("Enter a number : "))  
print("You entered:", num)
```

- **except ValueError:**

```
    print("That's not a valid number!")
```

- **else:**

```
    print("Conversion successful!")
```

- **finally:**

```
    print("Program finished.")
```

Enter a number: **123**  
You entered: **123**  
Conversion successful!  
Program finished.

Enter a number: **Ten**  
That's not a valid number!  
Program finished.

# Exceptions

- **try:**

```
x = int(input("x = "))
y = int(input("y = "))
result = x / y
```

x = 2

y = 1

**Result = 2.0**

Program finished

- **except ValueError:**

```
print("Please enter numbers only .")
```

x = ten

**Please enter numbers only.**

Program finished

- **except ZeroDivisionError:**

```
print("Division by zero is not allowed.")
```

- **else:**

```
print("Result = ", result)
```

x = 2

y = 0

**Division by zero is not allowed.**

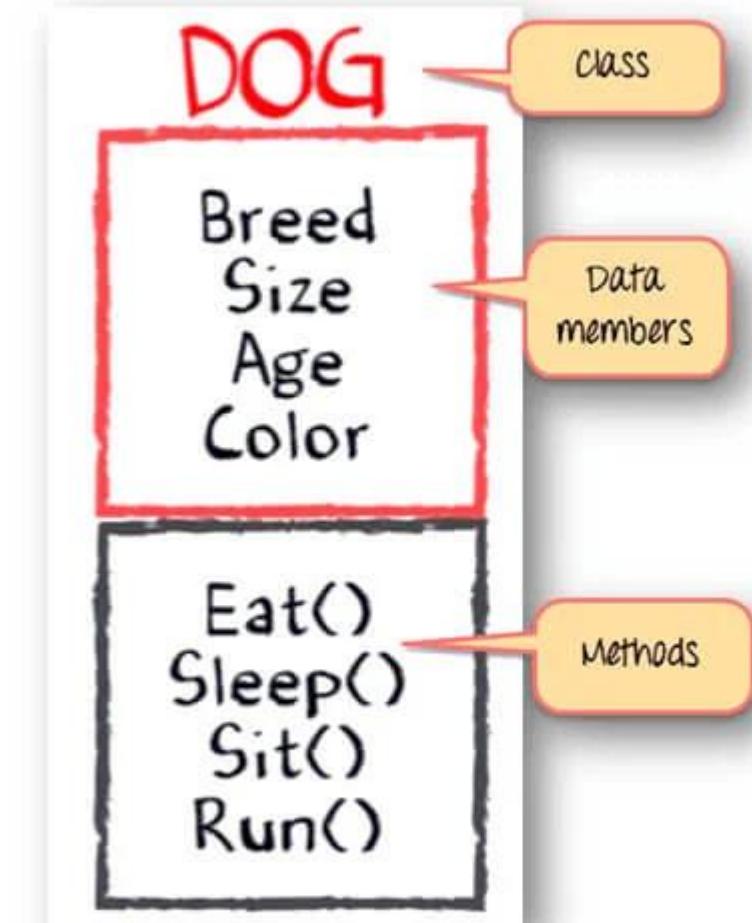
Program finished

- **finally:**

```
print("Program finished")
```

# Classes, Objects and Methods

- class MyClass:  
    x = 10
- myObject = MyClass()
- print(myObject.x)



# Classes, Objects and Methods

- class Person:

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

- myPerson = Person("Maykin W.", 25)
- print(myPerson.name)
- print(myPerson.age)

# Classes, Objects and Methods

- class Person:

```
def __init__(self, name, age):
    self.name = name
    self.age = age
def reverse_name(self):
    return self.name[::-1]
```
- myPerson = Person("Maykin W.", 25)
- print(myPerson.name)
- print(myPerson.age)
- print(myPerson.reverse\_name())

# Inheritance

- class Person:  
    def \_\_init\_\_(self, name, age):  
        self.name = name  
        self.age = age  
    def Fullname(self):  
        print(self.name, self.age)
- class Student(Person):  
    pass
  - myStudent = Student("Maykin W.", 25)
  - myStudent.Fullname()

# Modules

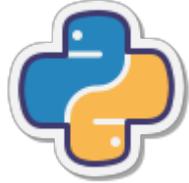
```
• def say_hello(name): (my_hello.py)
    print("Hello, ", name)
```

```
• import my_hello
• my_hello.say_hello("Maykin W. ")
```

# Modules

```
• def factorial(n):          (Fact.py)
    if n == 0 or n == 1:
        return 1
    else:
        return n*factorial(n-1)
```

- import Fact
- print("4! = ", Fact.factorial(4))



# The Code Awakens

Hello, Human. Let's Code.



**Maykin W., PhD**  
(Tue) November 4, 2025



maykin@owasp.org  
maykin@ieee.org  
maykin@acm.org

LINE @maykin



# Maykin W.

ເມືຂັນທົ່ງ ວິຊາສາຫະກຣນ

**Verisette | MKS & UBSL | CSC LAO | MAND**  
Chief Information/Technology Officer/Evangelist

**T-NET IT Solution | INSEC**  
Cybersecurity Advisor

**ARIT | ACinfotec | Training DD**  
Invited Instructor

**PPlus Network**  
SOC Manager (MOL)  
**iKNEX (Thailand)**  
Founding President



Ph.D. (ICT for Education), KMUTNB  
M.Sc. (Information Systems Security), MUT  
M.Sc. (Computer Science), NIDA  
B.Sc. (Computer Science), KMITNB

**Former Microsoft MVP**

Cloud and Datacenter Management  
Security

**Ex-IBMer (KBank, TBank)**

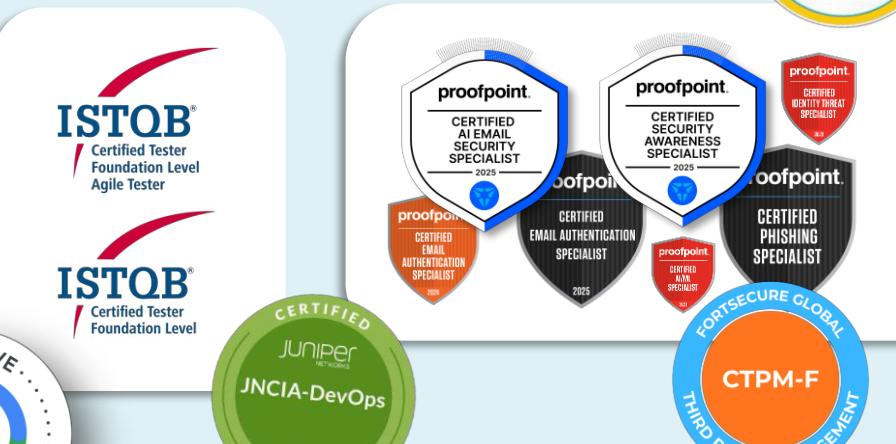
Defect Manager (K-Transformation)  
Test Conductor

**Ex-DST (Output)**

Offshore Software Developer

**Ex-EGA, Ex-DGA (Public Organization)**

Sr. Innovation Prototype Development Engineer  
Sr. Digital Technologist





Maykin Warasart (មេខិន វរសាត់)

(មេខិន វរសាត់, ខេត្ត ខេត្ត, ខេត្ត) #iKNEX #VolunteXTH #MiSSConf  
Verified email at ieee.org - [Homepage](#)

Cybersecurity Applied Cryptography Service Innovation Quality Attributes  
Process Improvement

FOLLOW

TITLE

Paper-based Document Authentication using Digital Signature and QR code  
M Warasart, P Kuacharoen  
4TH International Conference on Computer Engineering and Technology (ICCET 2012)

CITED BY  
121  
YEAR  
2012

Design and accountability analysis of a secure SMS-based mobile payment protocol  
S Duangphasuk, M Warasart, S Kungpisdan  
The 8th Electrical Engineering/Electronics, Computer, Telecommunications and ...

16  
2011

Coin Recovery from Inaccessible Cryptocurrency Wallet Using Unspent Transaction  
Output  
P Rakdej, N Janpitak, M Warasart, W Lilakiatsakun  
2019 4th International Conference on Information Technology (InCIT), 99-103

9  
2019

Synthesis of Gamified Social Collaboration via Mesh Community of Practice to Enhance  
Cybersecurity Awareness  
M Warasart, P Piriyasurawong  
2022 Joint International Conference on Digital Arts, Media and Technology ...

8  
2022

Design and analysis of a secure agent-based mobile bill payment protocol for bulk  
transactions  
P Limpittaya, M Warasart, S Kungpisdan  
2012 Ninth International Conference on Computer Science and Software ...

6  
2012

Somp: An sms-based operator-assisted mobile payment protocol  
S Kungpisdan, M Warasart  
Proceedings of the 2010 National Computer Science and Engineering Conference

3  
2021

Low Cost Automated OS Security Audit Platform Using Robot Framework  
H Dabthong, M Warasart, P Duma, P Rakdej, N Majaroen, ...  
2021 Research, Invention, and Innovation Congress: Innovation Electricals ...  
The 5th RMUTT CI-CI

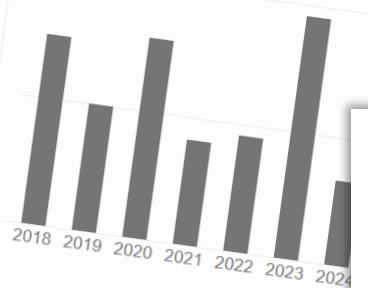
GET MY OWN PROFILE

SIGN IN

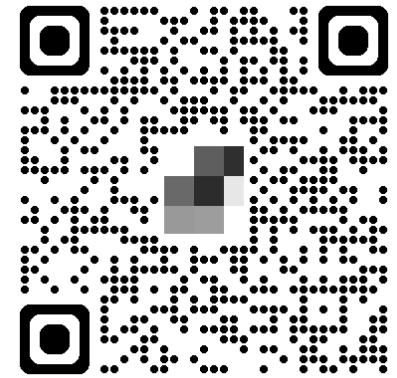
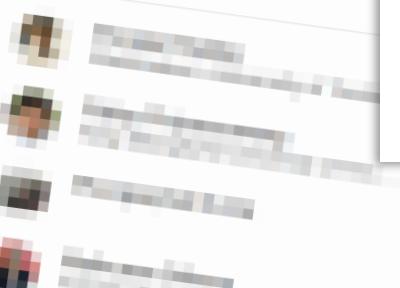
Cited by

VIEW ALL

	All	Since 2020
Citations	171	80
h-index	6	3
i10-index	2	1



Co-authors



security Engineers in Industry

