The Chinese University of Hong Kong

Department of Information Engineering

Summer Workshop – Fun with Information Engineering and Security

Lab 1 – Python programming

Introduction

Programming is the process of creating instructions or code that a computer can understand. It involves writing a series of instructions in a programming language to solve a specific problem or perform a particular task. These instructions, known as code, are written using a syntax that follows the rules and conventions of the chosen programming language.

Among hundreds of programming languages, Python is one of the most popular languages. Python is a general-purpose programming language that is well-known for having friendly syntax for beginners. The strong community of third-party library support also makes it widely used in the industry such as data analysis, machine learning, and web development. In the area of cybersecurity, python is often used to automate repetitive tasks and create tools for ethical hacking.

In this lab, we will learn some basic concepts of programming. Including basic syntax, variables, function and control flow.

Section 1: Introduction to Python Programming

Section 1.1: Hello Python

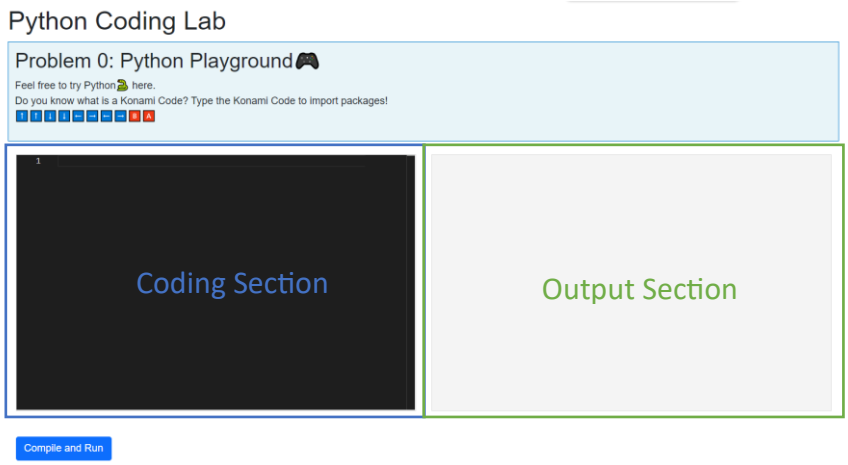
Say Hello to Python.

Try it!

Copy the below code snippet and paste it into the coding section of https://iesummerworkshop.github.io/py-lab.html, click the "Compile and Run" button to check the result

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

"print" is a built-in function of Python. It can help us to write something to the output section. We can declare the output inside the parentheses (). If you want to declare text (文字), a single quote (' ') or double quote (" ") is needed. You can put any text within the quote.



The web page should look like the below picture after you click on the 'compile and run' button:

Congratulations! You have completed your first Python program!

Try it!

Copy the following code snippet into https://iesummerworkshop.github.io/py-lab.html.

```
Print("Hello World")
```

Does the program work?

Python is a case-sensitive language. Uppercase Characters (e.g. "A") and lowercase Characters (e.g. 'a') consider different. For example, "Apple" is different from ''apple". In the above case, "print" is recognizable, but "Print" is not.

Section 1.3: Arithmetic Operation

Python operators can help us to achieve mathematical calculations in programming. The following are some of the arithmetic operators in Python:

Operator	Description	Example	Result?
+	Adds 2 operands	998 + 3	
-	Subtract 2 operands	998 – 3	
*	Multiply 2 operands	998 * 3	
/	Divide the first operand by the second operand	998 / 3	
//	Divide the first operand by the second operand and return the quotient only	998 // 3	
**	Exponent	998 ** 3	
%	Modulus	998 % 3	

Exponent: The exponent operator is used to raise the first operand to the power of the second, 10 ** 3 = 10^3 = 10×10×10 = 1000

Modulus: The modulus operator is used to find the remainder(餘數) when the first operand is divided by the second operand, e.g. 10 % 3 = 1

Try it!

You may go to https://iesummerworkshop.github.io/pyodide.html. What are the results of the example? Test the operation on the website by copying the example columns one by one and pressing "enter" to get the result.

```
Welcome to the Pyodide 0.26.0 terminal emulator
Python 3.12.1 (main, May 27 2024 13:56:13) on WebAssembly/Emscripten
Type "help", "copyright", "credits" or "license" for more information.
>>> 100 + 3
103
>>>
```

Section 1.4: Variables

Variables are named storage for data in a program. The program can access and change the value afterward. The following are the rules for naming a variable:

1. A variable name must start with a letter or underscore character
2. A variable name must not start with a number
3. A variable name can only contain alpha-numeric characters (e.g. A-Z, 0-9) and underscores (_)
4. Python has 35 keywords (aka. reserved words), they cannot be used as variable name. E.g. “if”, “else”, ”def”, “import”, “True”

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
a = 10
b = 20
c = a + b
print(c)
```

We declared 3 variables named a, b and c.

- a stores a value of 10
- b stores a value of 20
- c stores a value of a + b, where a = 10 and b = 20

What is the result of c?

Section 1.5: List

List is a named storage that can store multiple values. It is one of the built-in data types in Python. The following code snippet shows how to define a list in Python:

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

We declared one list named `fruits`

- The list length of `fruits` is 3
- There are 3 items stored in fruits: `"apple"`, `"banana"` and `"cherry"`

To access the item inside the list, we can refer to the index number. The index number starts from 0.

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
fruits = ["apple", "banana", "cherry"]
print(fruits[1])
```

What is the output?

We can also assign value to a specific index. For example, if we want to update "banana" to "orange":

```
fruits = ["apple", "banana", "cherry"]
fruits[1] = "orange"
```

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

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

What should you do to change the first item from "apple" to "berry"?

To get the length of the list, we can use the `len()` function by passing the list as the argument:

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
fruits = ["apple", "orange", "cherry"]
print(len(fruits))
```

What is the output?

To add a new item in Python, we can use the `append()` method. Using the append() method adds a new item at the end of the list:

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
fruits = ["berry", "orange", "cherry"]
fruits.append("melon")

print(len(fruits))
print(fruits)
```

1. What is the length of the list?
2. What is the index value of "`melon`"?

To remove an item from a list, we can use the `remove()` method. We can remove the specific item by its value:

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
fruits = ["berry", "orange", "cherry", "melon"]
fruits.remove("orange")

print(len(fruits))
print(fruits[1])
```

1. What is the length of the list?
2. What is inside fruits[1] now?

Section 1.6: Logical Condition

Boolean is a data type that has only two values: "True" and "False"

Python supports logical conditions which will return a Boolean value. The following are the logical conditions in Python:

Operator	Description	Example	Result?
==	Equal to	10 == 21	
!=	Not equal to	10 != 21	
<	Smaller than	10 < 21	
		10 < 10	
>	Greater than	10 > 21	
		10 > 10	
<=	Smaller than or equal to	10 <= 21	
		10 <= 10	
>=	Greater than or equal to	10 >= 21	
		10 >= 10	

Try it!

You may go to https://iesummerworkshop.github.io/pyodide.html. What are the results of the example? Test the operation on the website by copying the example columns one by one and pressing "enter" to get the result.

```
Welcome to the Pyodide 0.26.0 terminal emulator
Python 3.12.1 (main, May 27 2024 13:56:13) on WebAssembly/Emscripten
Type "help", "copyright", "credits" or "license" for more information.
>>> 1000 == 1000
True
```

Besides the basic operators, we can combine multiple logic conditions by using "and" and "or". The following shows the truth table of "AND" and "OR" operations:

AND

Statement 1	Statement 2	Output
True	True	True
True	False	False
False	True	False
False	False	False

OR

Statement 1	Statement 2	Output
True	True	True
True	False	True
False	True	True
False	False	False

Try it!

You may go to https://iesummerworkshop.github.io/pyodide.html. What are the results of the following table? Test the operation on the website and form an appropriate logic statement using Python to get the result

Statement 1	Output of statement 1	Operator	Statement 2	Output of statement 2	Output
3 > 1	True	AND	10 > 3	True	True
2 < 4		AND	100 > 2		
3 > 1		OR	10 > 3		
2 < 4		OR	100 > 2		
10 < 2		OR	27 > 103		

```
Welcome to the Pyodide 0.26.0 terminal emulator
Python 3.12.1 (main, May 27 2024 13:56:13) on WebAssembly/Emscripten
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 > 1 and 10 > 3
True
```

Section 1.7: Control Flow

Control flow refers to the order in which the statements or instructions of a program are executed. It determines the path that the program takes based on certain conditions and decisions. Control flow allows programmers to specify the flow of execution within their code, enabling them to make choices, repeat actions, and handle different scenarios.

if-else statement

The if statement allows us to execute a block of code conditionally. The program can make decisions and perform different actions based on the condition. The following code snippet shows how an if-else statement can be used in Python:

```
if first_condition:
    do_something_1
elif second_condition:
    do_something_2
else:
    do_something_3
```

- The keyword `if` initiates the conditional statement
- The keyword `elif` extends the `if` statement with additional clauses
- The keyword `else` handles the case that does not match any condition in `if` or `elif`
- The colon `:` indicates the start of a code block associated with the "if" statement
- The indented block is executed only if the condition is True

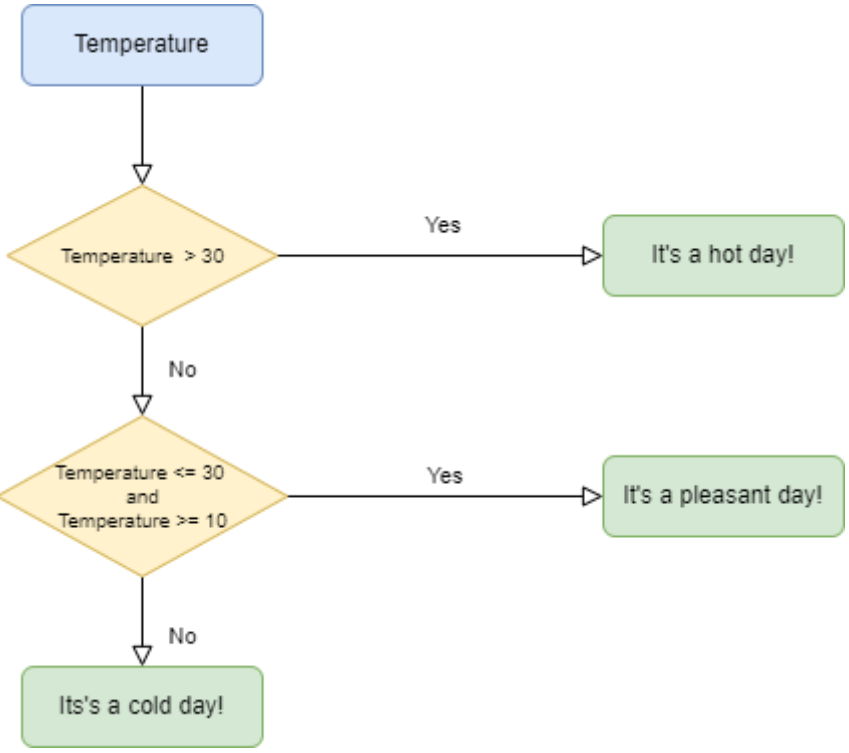
Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section. Change the value of the variable `temperature`:

```
temperature = 30

if temperature > 30:
    print("It's a hot day!")
elif temperature <= 30 and temperature >= 10:
    print("It's a pleasant day!")
else:
    print("It's a cold day!")
```

The flowchart of the program:



1. What is the output of `temperature = 36` ?
2. What is the output of `temperature = 24` ?
3. What is the output of `temperature = 5` ?

As you may noticed, we added some space before each print statement, it is indentation. Indentation is one of the syntax in Python. Syntax refers to the set of rules that define the structure and format of valid statements or expressions in a programming language. The syntax is just like Grammar in English, each programming language will have its own set of syntax.

Indentation is one of the rules in Python. Indentation refers to the space at the beginning, where Python uses indent to indicate a block of code.

Try it!

Try to remove the spaces before the print function. Copy the following code snippet into https://iesummerworkshop.github.io/py-lab.html.

```
temperature = 30

if temperature > 30:
print("It's a hot day!")
elif temperature <= 30 and temperature >= 10:
print("It's a pleasant day!")
else:
print("It's a cold day!")
```

What is the result?

for-loop

A "for" loop is a control flow statement that allows you to iterate over a sequence of elements and perform a specific action for each element. The following code snippet shows how an if-else statement can be used in Python:

```
for i in range(0, 10):
    do_something
```

- The keyword `for` initiates the for loop
- The program iterates over a sequence of numbers from 0 – 9 (i.e. 10 times)
- A variable `i` is defined. It represents the current value in each iteration. You can define a different variable name
- The colon `:` indicates the start of a code block associated with the "for" statement

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
for i in range(0, 10):
    print(i * 2)
```

What is the output?

1st round: i = 0, i * 2 = 0 * 2 = 0

2nd round: i = 1, i * 2 = 1 * 2 = 2

3rd round: i = 2, i * 2 = 2 * 2 = 4

…

10th round: i = 9, i * 2 = 9 * 2 = 18

Besides using a for loop with numbers, we can also use it to iterate over a sequence of elements. The following code snippet shows how an if-else statement can be used in Python:

```
for item in sequence:
    # Code block to execute for each item
```

- The keyword `for` indicates the start of the loop.
- A variable `item` is defined. It represents the current element being processed in each iteration. You can choose any variable name you like.
- The sequence is an object contains multiple objects, e.g. list

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

What is the output? Is there any alternative way to achieve the same result?

```
fruits = ["apple", "banana", "cherry"]
for i in range(0, len(fruits)):
    print(fruits[i])
```

Section 1.8 Functions

Function is a block of code. The code only runs when the function is called. Function helps break down large and complicated problems into chunks, and also increases code reusability. The following code snippet shows how a Python function is defined:

```
def my_function():
  print("You called the function!")
```

- function is defined by a keyword `def`.
- `my_function` is the function name
- The code inside the function should start with an indent

To call a function, we can use the function name followed by a parathesis:

```
1   def my_function():
2     print("You called the function!")
3
4   my_function()
5   my_function()
```

The program is executed in the following order:

line 4 → line 1 → line 2 → line 5 → line 1 → line 2

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
def myfunction():
    print("You called the function!")

myfunction()
myfunction()
```

What is the output of this program?

We can also pass information to the function using arguments. Arguments are defined inside the parathesis.

```
def function_with_arg(value1, value2):
  print("You called the function!")
  print("the value you passed are: ", value1, value2)

function_with_arg('a','b')
function_with_arg('1','abc')
```

- function is defined by a keyword `def`.
- `function_with_arg` is the function name
- There are 2 arguments in the function: `value1`, `value2`
- The argument can be used within the function
- The information can be passed when the function is called

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

```
def function_with_arg(value1, value2):
  print("You called the function!")
  print("the value you passed are: ", value1, value2)

function_with_arg('a','b')
function_with_arg('1','abc')
```

What is the output of this program?

A function can have a return statement that specifies the value or values to be returned from the function when it is called. The return statement allows a function to compute a result and provide it back to the caller. The following code snippet shows how a Python function is defined:

```
def function_with_return():
    return 10

result = function_with_return()
print(result)
```

- The function `function_with_return` is defined, it returns a value of `10`
- The function `function_with_return` is called, and the return value of the function is stored in a variable `result`
- The variable `result` stores the value `10` returned from the function

Try it!

Copy the following code snippet to https://iesummerworkshop.github.io/py-lab.html coding section.

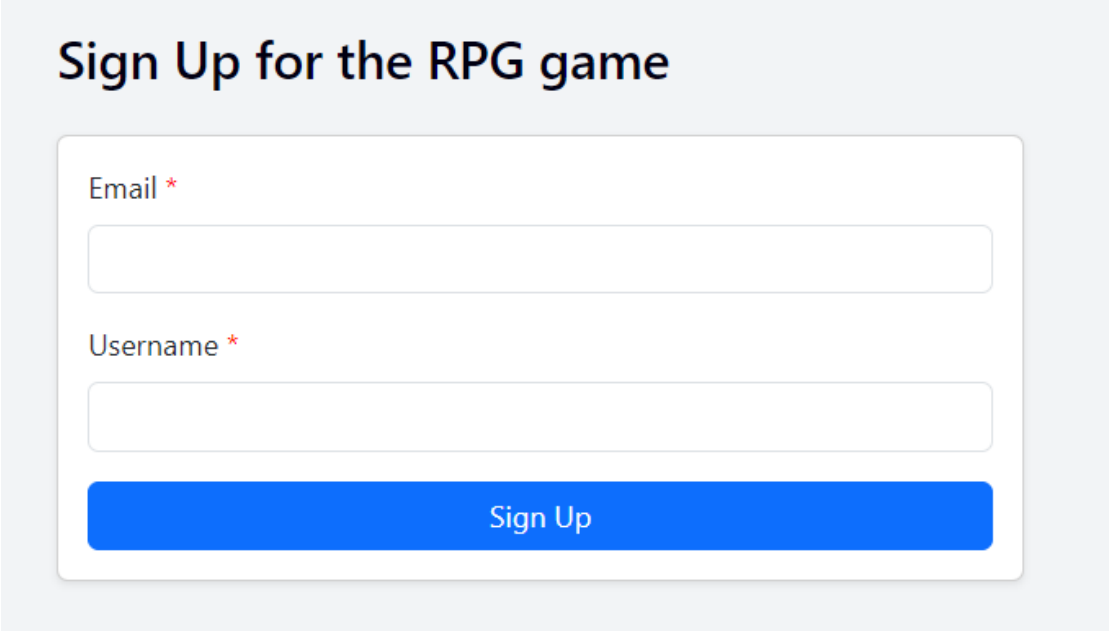
```
def sum_of_two(value1, value2):
    value = value1 + value2
    return value

value = sum_of_two(10, 20)
print(value)
```

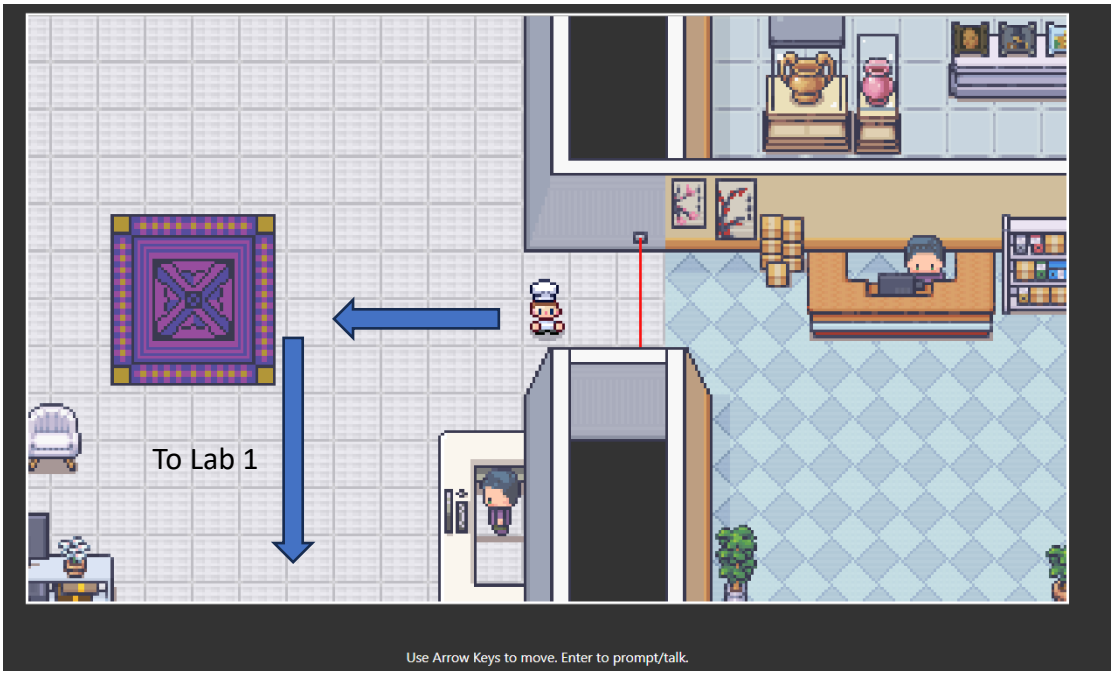
What is the output of this program?

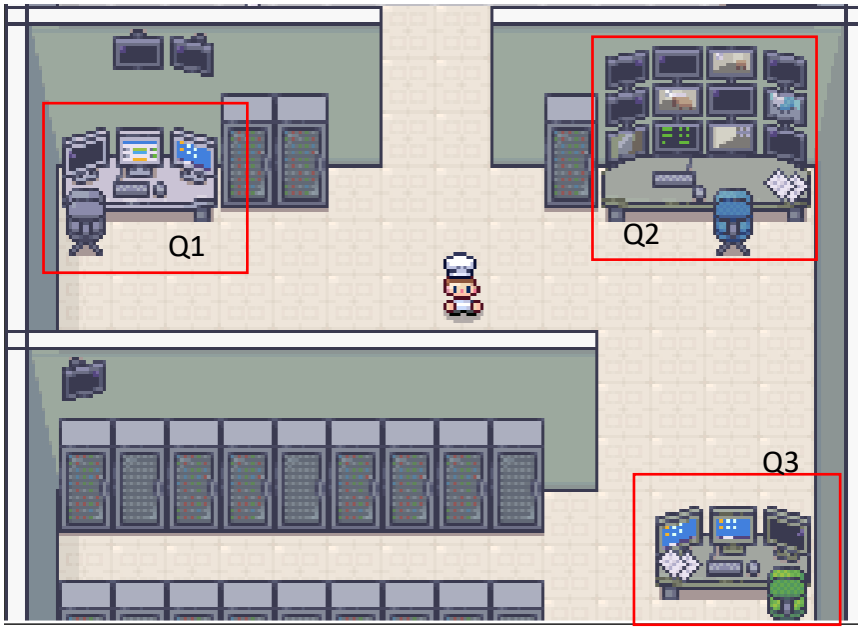
Section 3: Challenge Time

Before we start the challenge, please sign up at https://iesummerworkshop.github.io/sign-up.html



After sign-up, you should be redirected to the RPG page. Or else please go to https://iesummerworkshop.github.io/rpg/page.html to enjoy your challenge. Walk in front of the computer and press enter to interact with it. Mark down the key you get!





1. [☆] Area of circle

Complete a function area_of_circle(r) that calculates the area of the circle.

Given that radius r = 3 is passed to the function, the function should return 28.
Please use 3.14 for pi and round off the value to the nearest integer

2. [☆] Sphinx's riddle

The Sphinx asked: What goes on four feet in the morning, two feet in midday, and three feet in the evening?

Oedipus answered: 'Man: as an infant, he crawls on all fours; as an adult, he walks on two legs and; in old age, he uses a walking stick

Complete the function man(feet) that returns 'infant'/'adult'/'elderly' depending on the number of feet. Return 'unknown' if the number of feet is not 2, 3 or 4.

Given that feet = 2 is passed to the function, the function should return adult.

3. [☆☆] Prime number

Do you know cryptography leverages large prime numbers? A prime number is a natural number greater than 1 that has only 2 factors: 1 and itself

Complete the function prime(max) to return a list of prime numbers smaller or equal to max, where 3 <= max <= 100

Given that max = 30, the function should return [2, 3, 5, 7, 11, 13, 17, 19, 23, 29].

Interact with this counter to submit the key for lab 1 and lab 2