Programming, Computer and Data Types

Introduction to Computer Programming (Python)

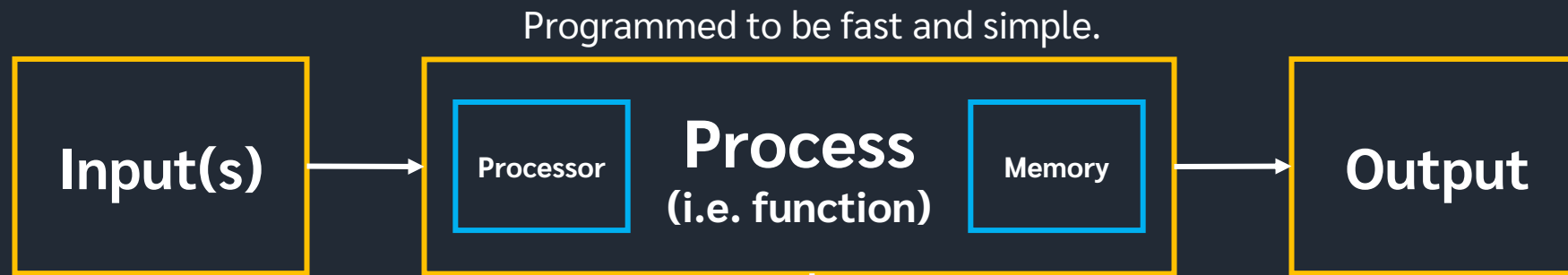**Week 1**

Vivatsathorn Thitasirivit

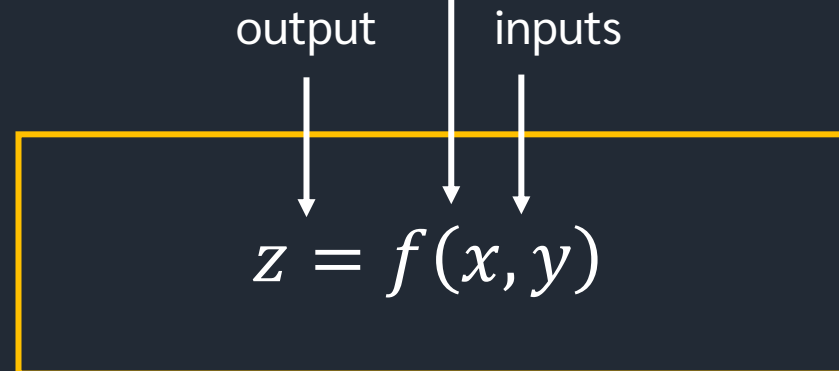*Rev. 1.0 (Course 1/2023)*

https://vtneil.com

What is Computer Programming?

Simple (Linear) Process

Programmed to be fast and simple.

| Input(s) | → | **Process**
(i.e. function)
Processor   Memory | → | Output |

Mathematical Equivalence
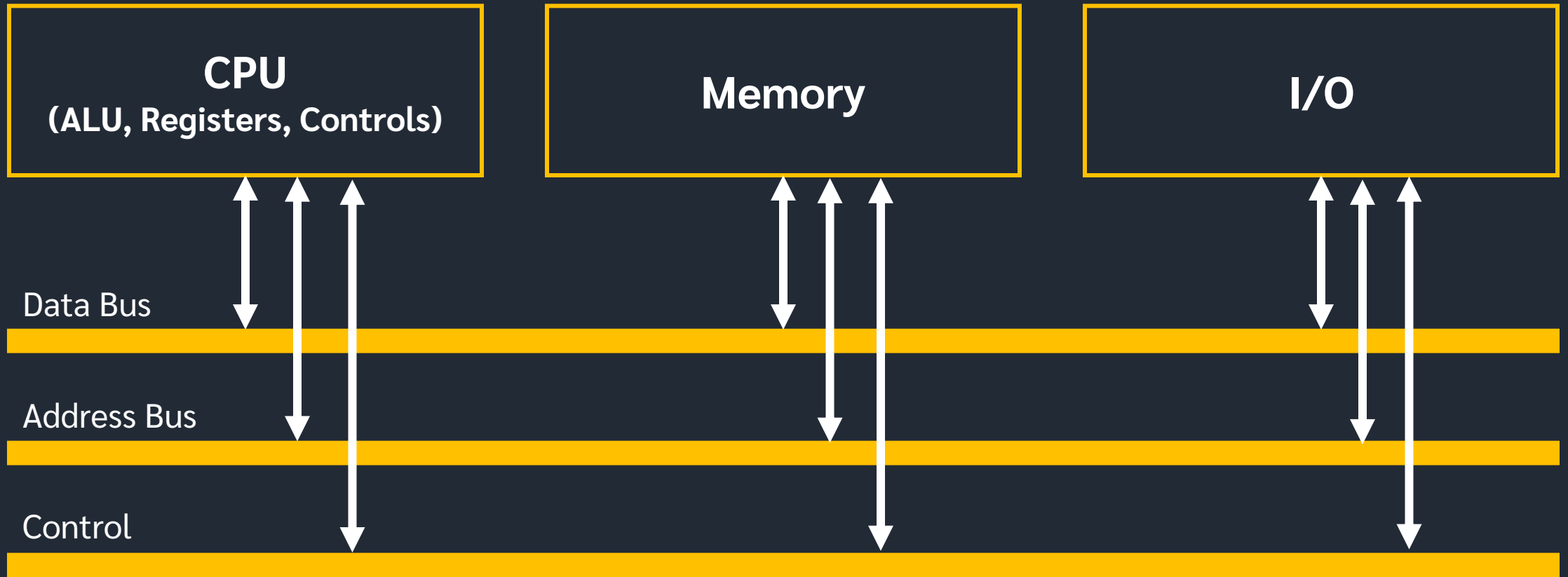
output   inputs

$$z = f(x, y)$$
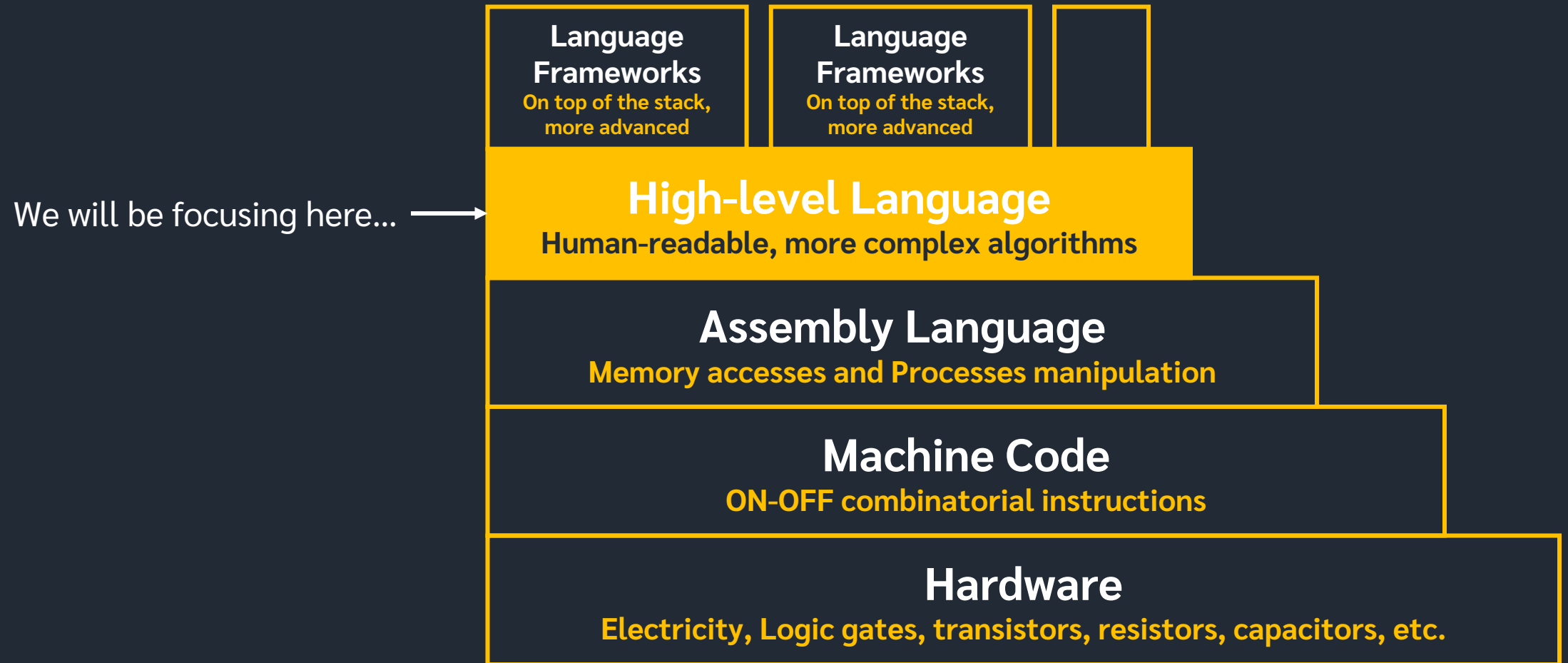
What is Computer Programming?
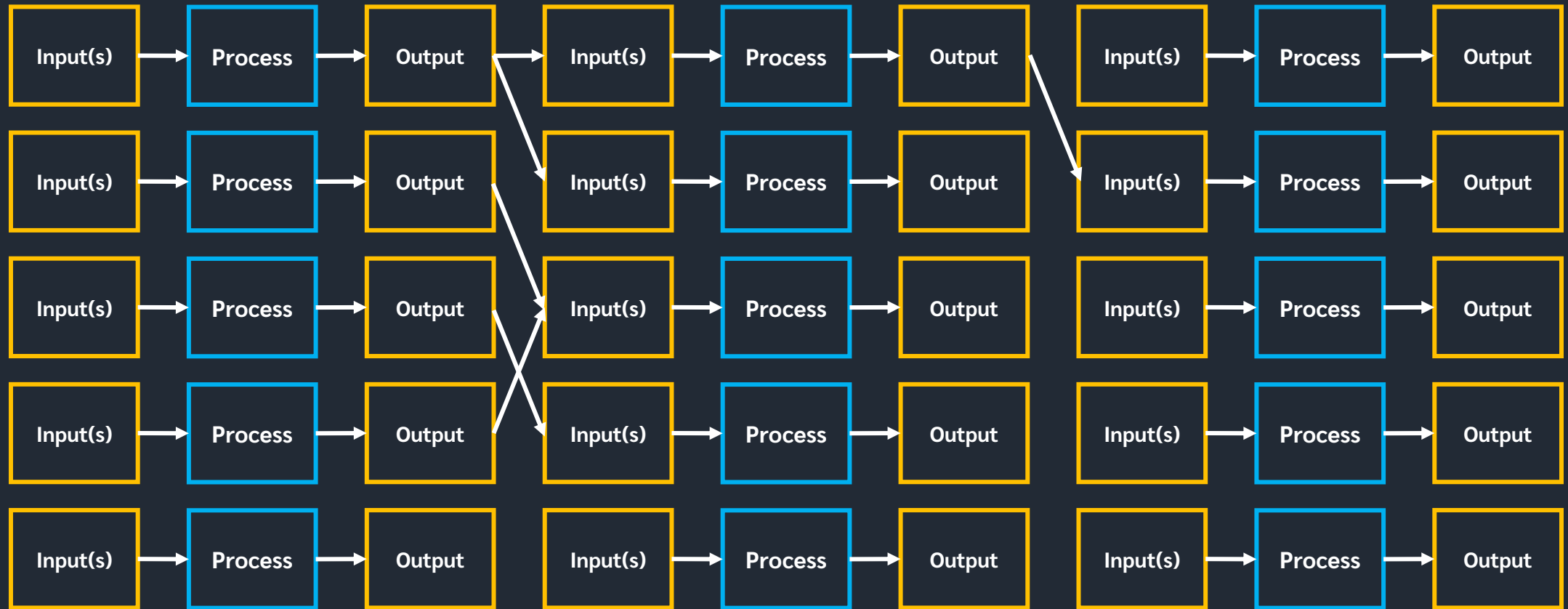
A computer may contain a lot of these individuals and chains of processes.

Computer Communication (Simplified)

Layers of Computer Programming Languages

Language Frameworks
On top of the stack, more advanced

Language Frameworks
On top of the stack, more advanced

We will be focusing here... →

High-level Language
Human-readable, more complex algorithms

Assembly Language
Memory accesses and Processes manipulation

Machine Code
ON-OFF combinatorial instructions

Hardware
Electricity, Logic gates, transistors, resistors, capacitors, etc.

A Program Flow

Input(s) → Process 1 → Process 2 → Output

Operations
How to manipulate a representation/data.

**Objects**
A representation of data.

Algorithm

Definition

A set or sequences of processes and/or operations to solve a specific **problem**.

"Procedure through which we obtain the solution of a problem."

Characteristics of an algorithm

1. **Non-ambiguity**   Unique interpretation for each input (Deterministic mapping).

2. **Executability**   Must be possible to execute each statement in a finite amount of space and time.

3. **Finiteness**   After executed, must terminate within a finite amount of time.
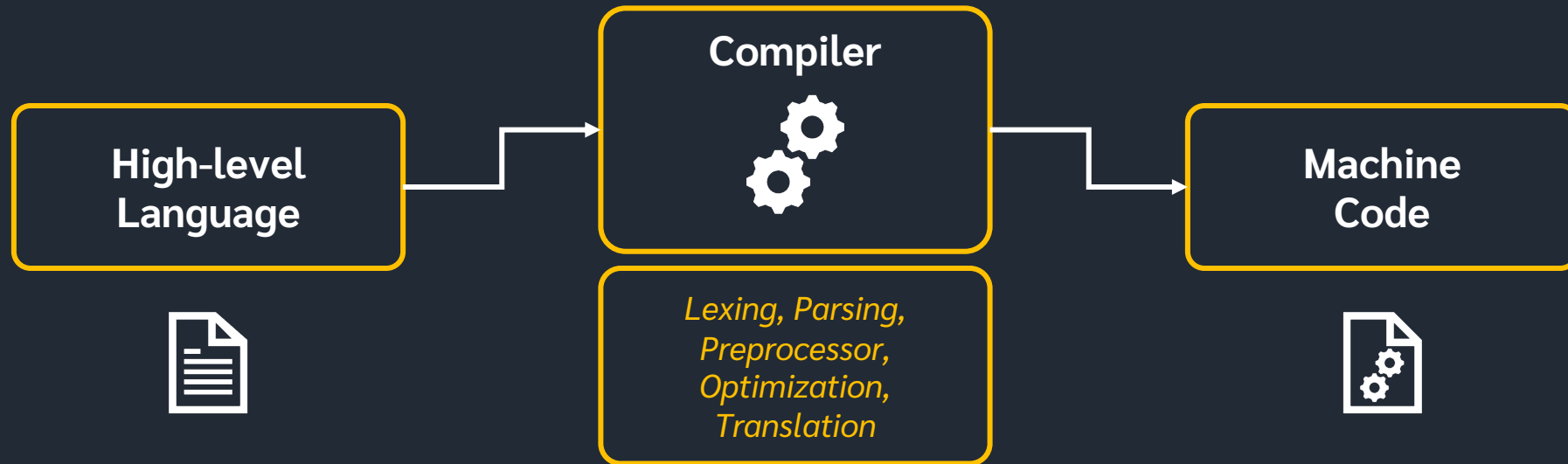
Programming Paradigms
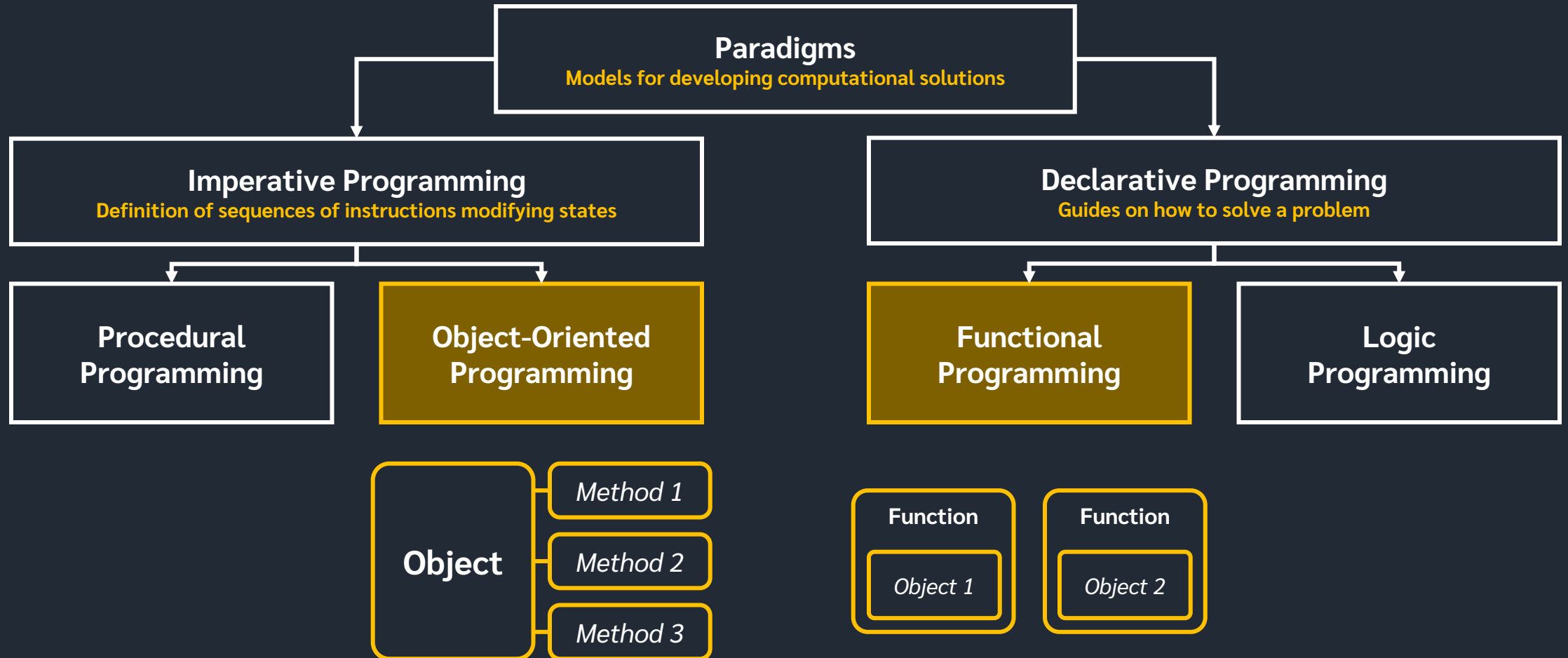
How to translate *High-Level Language* to *Low-Level Language*

1. Compiler
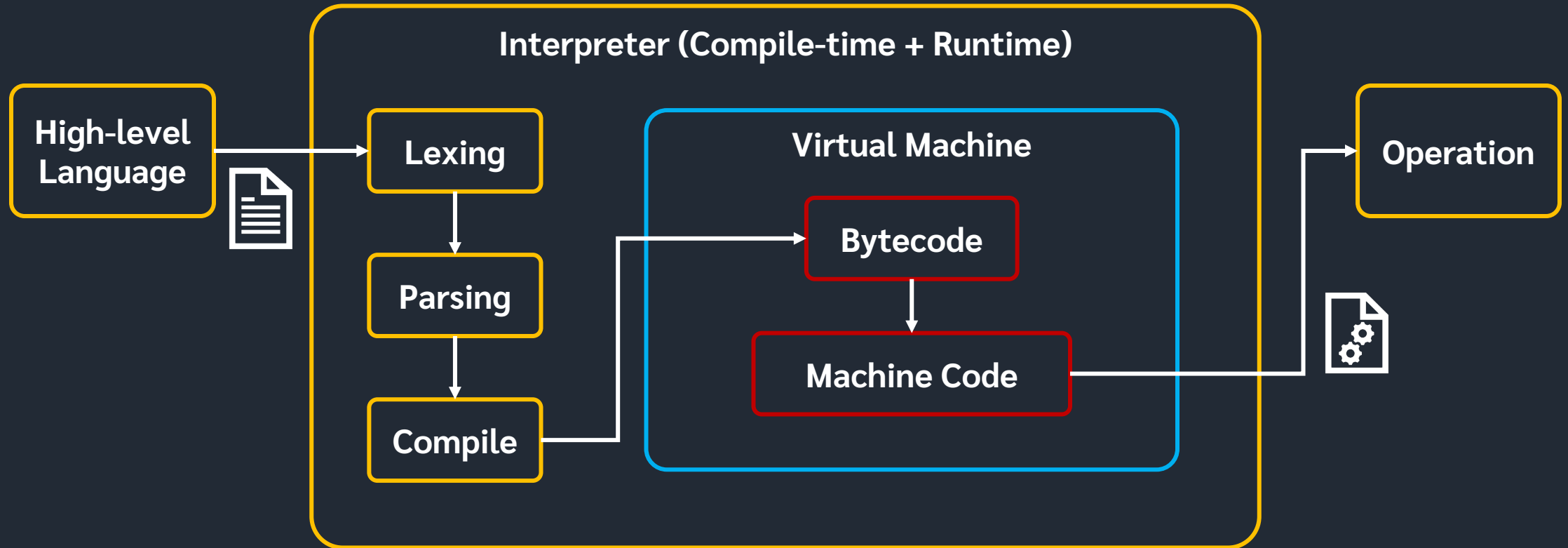*Example: C, C++*

How to translate *High-Level Language* to *Low-Level Language*

2. Interpreter
Example: Python, Java

Interpreter (Compile-time + Runtime)

High-level Language → Lexing → Parsing → Compile

Virtual Machine
Bytecode → Machine Code

→ Operation

Python
Showing texts on the screen

You can either use single quotes or double quotes.

In Python, both are treated the same way. Other languages may not, so be careful.

```python
# Printing a message

print("Hi Thailand!")

# Alternative way of printing a string

print('Hi Thailand!')
```

Python
Escape Characters

In most programming languages, "backslash" or '\' are used as macros for typing special characters which are not on the keyboard.

Backslashes are followed by a sequence of characters.

\"   *Double quote*
\'   *Single quote*
\\   *Backslash*
\n   *New line*
\r   *Carriage return*
\t   *Tab*
\b   *Backspace*

```python
# Example

print("Hello,\nUser")

# Shows:

Hello,
User
```

Python
Arithmetic Binary Operations

```python
# Printing a number
print(99)

# Addition
print(10 + 10)

# Subtraction
print(20 - 5.8)

# Multiplication
print(5 * 20)

# Division
print(3 / 4)

# Floor Division
print(3 // 4)

# Modulo
print(45 % 6)

# Exponentiation
print(3 ** 2)
```

In Python, these are several binary operations you can use on a pair of "numbers."

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Floor Division
6. Modulo
7. Exponentiation

Note: There is operator precedence in Python, just like math.

Try: (5 + 9 * 3) – 5 ** 2 / 5

Python
Variables

Variables, just like in math, are elements subjected to be changed.

A variable can be "declared," "initialize," "accessed," or "assigned to some value."

Other than variables, there are "constants" which are not subjected to be changed, but in Python, there is no way to declare a constant.

Side notes:
In <u>statically typed languages</u>, a variable can have only one type of data.

In <u>dynamically typed languages</u>, like Python, a variable can be declared without specifying its data type.

```python
# Variable Declaration
x = 8
y = 10.5

print('x is', x)

print('x + y is', x + y)
```

Python *behind the curtain*
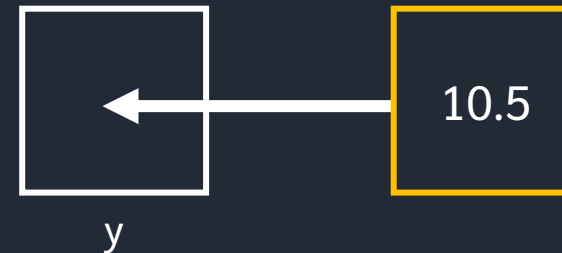# Variables

```
# Variable Declaration
y = 10.5
```

Address	Value
...	...
0x0000A200	10.5
...	...

Rough representation of program's memory

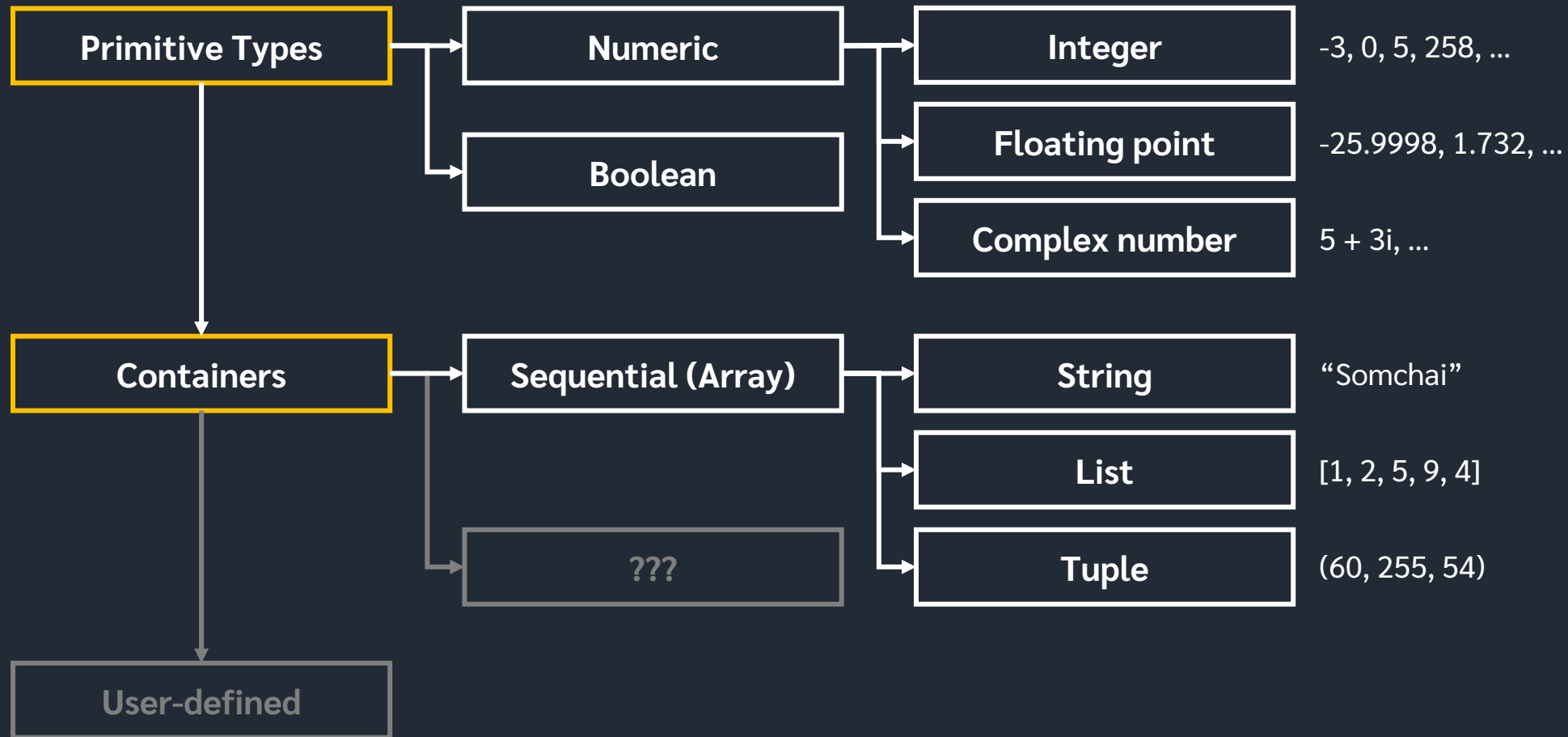1. Create temporary variable in RAM with value 10.5

y

2. Create variable *y* in RAM which will hold some value

y

3. Assign (copy) value of 10.5 into *y*'s value

y

General Data Types in Python Programming

Primitive Types → **Numeric** → **Integer** -3, 0, 5, 258, …

Boolean → **Floating point** -25.9998, 1.732, …

Complex number 5 + 3i, …

Containers → **Sequential (Array)** → **String** "Somchai"

List [1, 2, 5, 9, 4]

??? → **Tuple** (60, 255, 54)

User-defined

Data Types: Examples

```python
# Integer
a = 3
b = -5

# Floating point
c = 43.668

# String (An array of characters)
name = "Somchai"

# List of integers
d = [1, 1, 2, 3, 5, 7]
```

Array Representation of Python List

Value	1	1	2	3	5	7
Index	0	1	2	3	4	5

Python
Very Basic String Operations

In Python, addition and multiplication operator can be used on string type.

<u>Addition</u>: Concatenate any number of strings of any lengths
e.g., a + b + c + …

<u>Multiplication with integer</u>: Duplicate string *n* times.

```python
# String 1
a = 'Peter'

# String 2
b = 'Griffin'

# String Concatenation by addition
print(a + b)
print(b + a)

# String Duplication by multiplication
print(3 * a)
```

Input data from Keyboard

```python
s1 = input('What\'s your name?: ')

s2 = input('What\'s your age?: ')

print('Your name is ' + s1 + ' and you are ' + s2 + 'years old.')
```

Python has a function allowing user to input a "string" from keyboard.

You can type something in, then press Enter (which Python will detect a new line character ("\n").

For example, you type "hello" and Enter in the input prompt, the program will assign "hello" string to designated variable.

Python
Type casting

```
a = input()

b = input()

# Without casting, just concatenating strings
print(a + b)

# Cast string to integer
print(int(a) + int(b))
```

Python has a function allowing user to input a "string" from keyboard.

You can type something in, then press Enter (which Python will detect a new line character ("\n").

For example, you type "hello" and Enter in the input prompt, the program will assign "hello" string to designated variable.
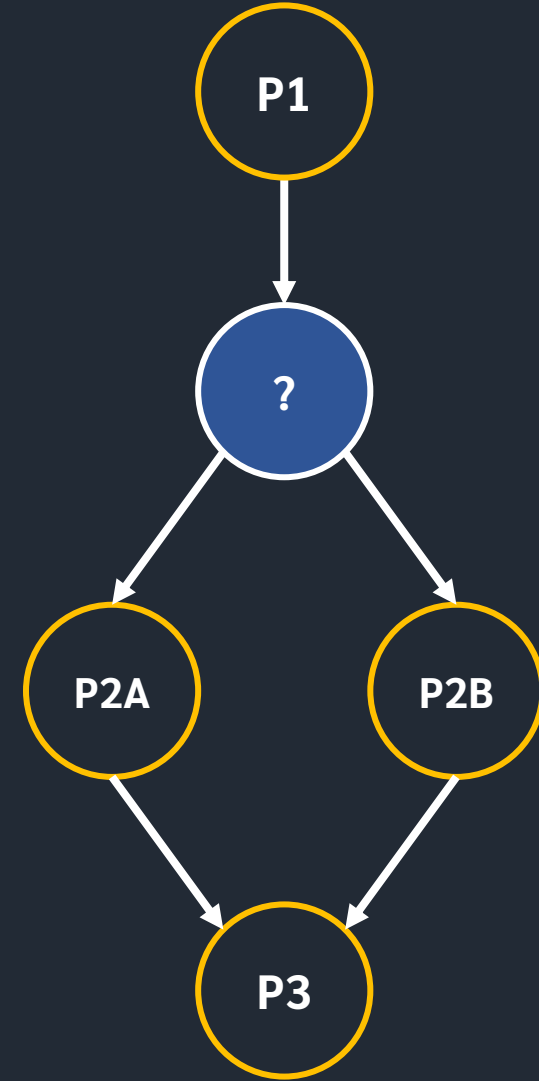
Program Flow & Control Flow

A program flow is the sequence and set of processes the program will operate.

A program may have a control flow *controlling* in what order and what processes should the program follow.

There are 3 types of control flow:
1. Selection statements,
2. Iterative (loop) statements,
3. Transfer statements,

which we will discuss in detail later.

Language Features
Statements vs Expressions

A **statement** "does" something.

An **expression** always "produces" at least one value.

We will discuss later in control flow.

Examples: