INIAD CS Essentials

1-2: Understanding Variables

Computers store various information that can be retrieved. In Python you can use the memory with a mechanism called variable.

1. Understanding variables and assignments

Variable is a place to store data in Python

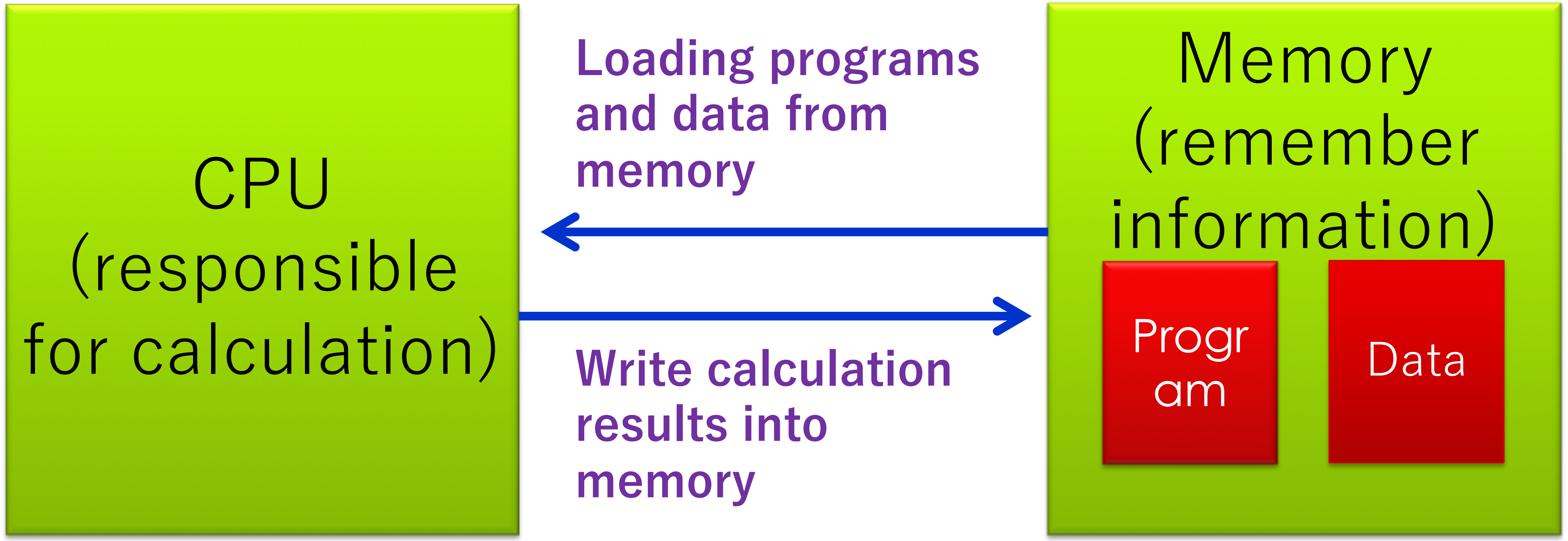
Computer needs "memory"

- For a computer to work, mechanism of "memory" is needed
  - If a cook cannot remember how long they have heated or whether he has put the seasonings, he cannot cook food properly

I have put salt and peppers, so the taste shall be OK. The rest is to heat 3 minutes more!

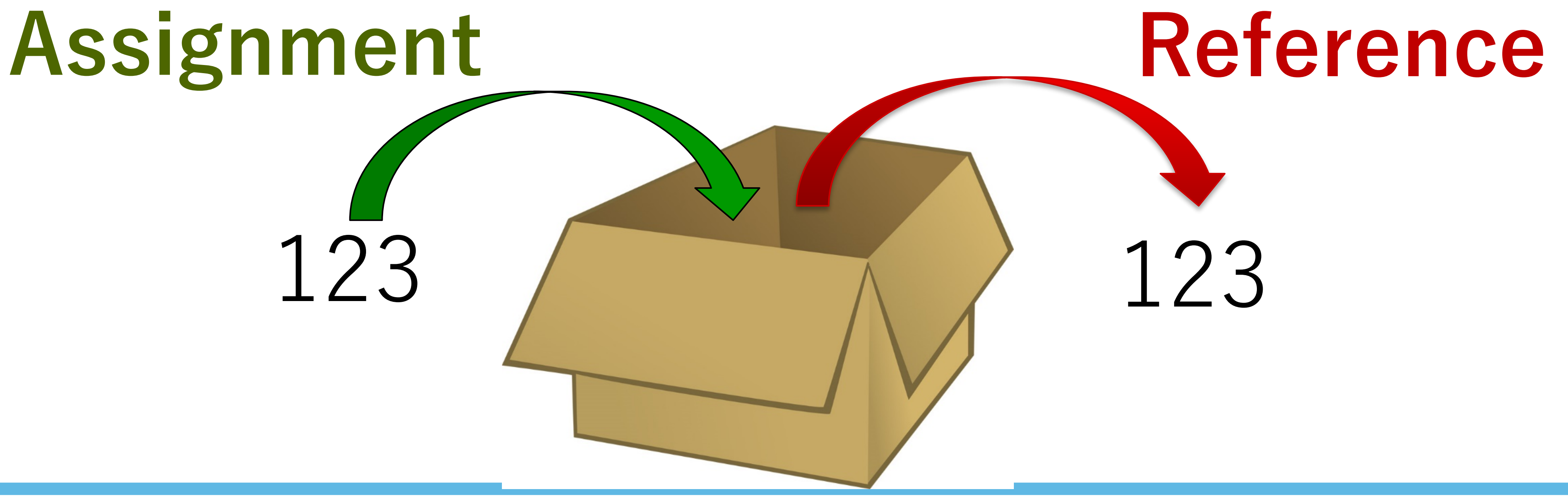
Computer and Memory

- How computers work
  - **CPU** is the component responsible for calculation
  - Computer does not work with CPU alone; it works by storing programs and data in **memory**, and by reading or updating the stored contents


CPU (responsible for calculation)
Loading programs and data from memory
Write calculation results into memory
Memory (remember information)
Progr am
Data

Let's store data in memory using Python

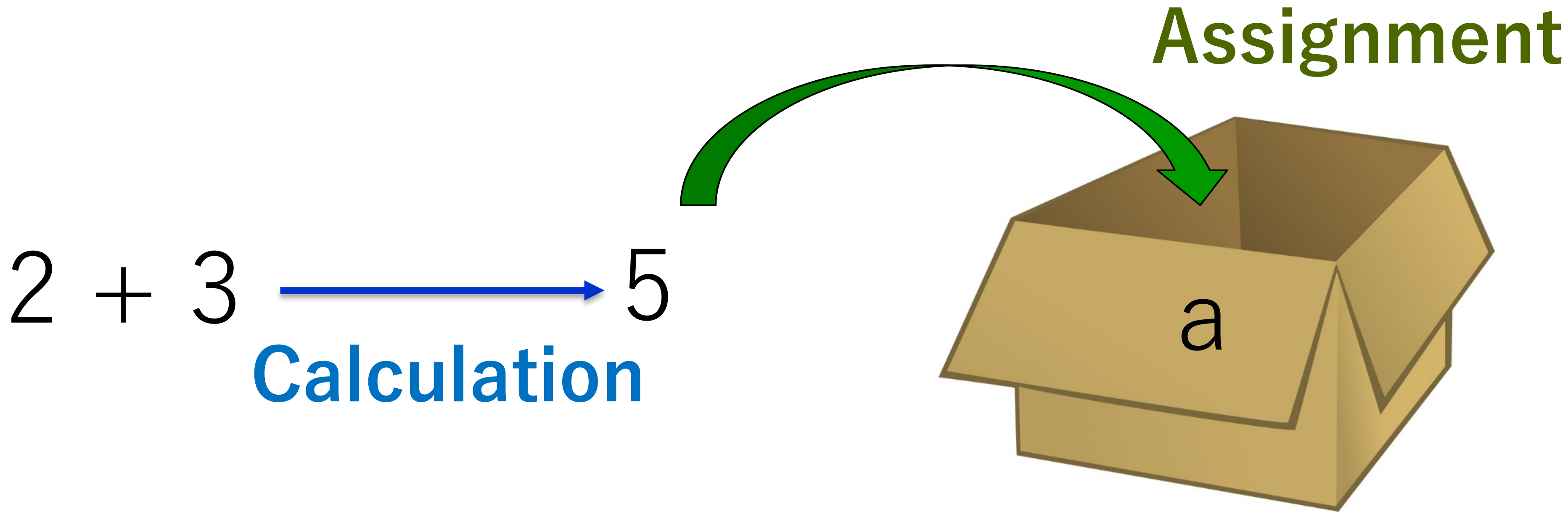
- In Python, you can use **variable(s)** to memorize and remember data
- For variables, two basic operations are provided
  - Assignment: Putting data into a variable
  - Reference: Getting the contents of the data in a variable



Let’s assign values to variables

- You can assign values to variables by writing:
 「*variable_name* = *expression*」

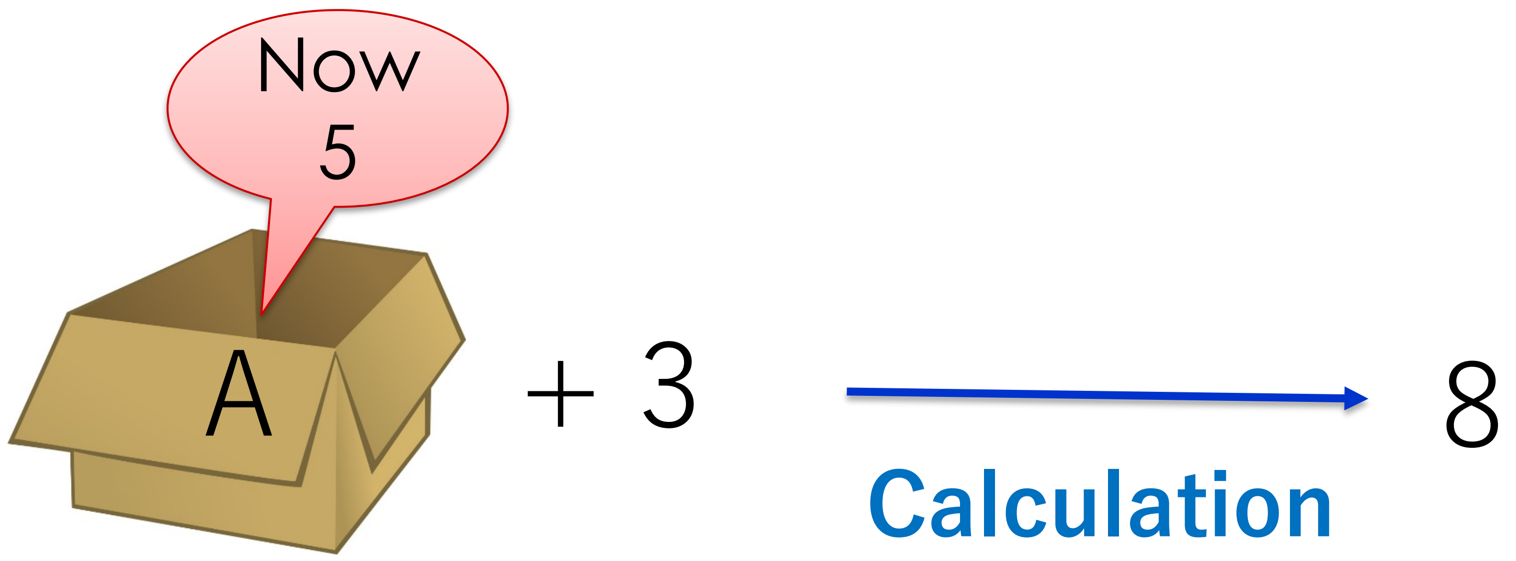
- For example, `> a = 2 + 3` means
  - Calculate 2 + 3 and store the result into variable a



Let’s refer to variables

- You can read the variable content just by its name in expression

- For instance, `> a + 3` means
  - Get the variable a content, and calculate addition of it with 3



Now 5

A

+ 3

Calculation

8

Assignment is NOT the equal symbol...

- Assignment (=) is different from mathematical equal (=) symbol
  - It simply puts the right-hand-side value to the left-hand-side variable
  - For example, if you assign a = 1, and then assign a = 2, the previous assignment will be lost

- For example, what does the following mean?

  -
    ```
    > a = a + 1
    ```

Meaning of a = a + 1

- Assuming variable a has an initial value of 3
  - a + 1 will first calculate the value as 4
  - Then the result 4 will be assigned to variable a

- In other words, a = a + 1 means:
  - “Increment the variable content by + 1”
  - Variables in Python are only containers, their values can change; unlike mathematical variables used to mean “unknown values”

2. Naming variables

In order to make the program easy to understand, it is important to name variables appropriately

Variable names can be any character or strings ...

- It can be single characters like a, b, or it can be strings like amount, faculty_name, etc.

- Basic naming rules
  - The first letter must be an alphabetic character or underscore (_)
  - From second character onwards you can use alphanumeric characters and underscores
  - Some words known as “Reserved Words” cannot be used as variable names
    - For example, if, else, lambda, finally, …

Variable names are easy to understand

- Compare these two calculating the area of a circle
  - Two examples: same process, only variable names are different
  - Which one is easier to understand?

```
> x = 3.14159
> y = 5
> x * y * y
```

```
> pi = 3.14159
> r = 5
> pi * r * r
```

How to make easy-to-understand variable names?

- What do a, b, and c mean??
- Basically you should use “meaningful” names in English

  -
    ```
    > a = b + '␣' + c
    ```
    What do a, b, and c mean??
  - Rather use
    ```
    > full_name = first_name + '␣' + last_name
    ```
- Not so good examples, using non-English names

  -
    ```
    > goukei_kingaku = int((1980 + 98) * 1.08)
    ```

  -
    ```
    > 合計金額 = int((1980 + 98) * 1.08)
    ```

Follow the programming norms and conventions

- In Python, it is customary to use underscore for variables containing multiple words
  - Example: `number_of_days, time_elapsed, ...`
  - This naming method is called “Snake case”
    - Some other languages use Camel case method(e.g., numberOfDays, timeElapsed), but it is good to use the community norms

- There are shorter recommended names, which is frequently used for specific usages
  - Example: variable i, j used as index and iterator

N.B. Distinguish variables from strings

- Don‘t confuse with “string” you learned last time!
- Can you explain what happens when you do the following?

```
> inoue = "enryo"
> enryo = "sakamura"
> toyo = sakamura
> inoue = enryo
> inoue
```

3. What to do when error is reported?

Python gives you an error message when your program contains some mistakes

I made a mistake!... But don't worry

- Read the error message.
  - Python is not easily broken even if you made something wrong

  -
    ```
    > ' 123 ' + 45
    ...
    TypeError: Can't convert 'int' object to str
    implicitly
    ```

- Normally there are two parts in an error message
  - Category of the error : TypeError
  - Error description : Can't convert... implicitly

Types of errors

- There are a lot of different types of errors
  - ZeroDivisionError : Tried to calculate 「÷ 0」
  - NameError : A name is not available
    (Example: variable name not defined)
  - SyntaxError : Python’s grammar (syntax) not followed
  - TypeError : Error due to type inconsistency
  - …

Description of error message

- Error messages help you to correct your program

```
> '123' + 45
...
TypeError: Can't convert 'int' object to str implicitly
```