PALs CSC-152

Section 02, Professor Kim, M-W-F

Tony Stone

November 12, 2025

Lesson I

1 Input, Processing, and Output

Many tasks in computer science involve collecting some kind of data, manipulating that data, and producing something from it. Imagine a bank, the input might be your account information, and request. The processing is all of the action the bank must do in order to make your request happen, or perhaps explain why they cannot complete your request. The output is that the bank completes your request, whether that be cash from a withdraw, or a receipt for a deposit.

1.1 Input

Here, our data is going to come from a user. We are going to prompt them for some kind of input. When we receive the user's input, we would like to store it in a variable, for further use, i.e. processing. This is how we will get input from a user. The prompt string we stick inside of input() will be output to the console, where the user can then type their answer and hit enter. It is generally good practice to include a break-line character '\n', or a simple space in the string, as Python does not automatically add any sort of spacing. Also, in the following examples, I will use a '|' to indicate the parts where a user would be typing.

```
userName = input("What is your name? ")
```

Output:

```
What is your name? |
```

1.2 Processing

Now that we have input, and it is stored, we can use it! There are some considerations, however. One consideration is that the input is a string, even when input a number. Consider:

```
num = input("input a number: ")
print(type(num))
```

Output:

```
<class 'str'>
```

class 'str' gets printed, even with a number-only input. Sometimes this is useful, when we are dealing with strings... But when we would like to use the numbers as actual numbers, we need to **typecast** them:

```python
num = input("input a number: ")
num = int(num) # Typecast string input into an integer
print(type(num))
```

Output:

```
<class 'int'>
```

This now prints: **class 'int'** and can be used as such. This typecasting works for most datatypes: int(), float(), str(), etc.

Making sure your datatypes are in order is only a small snippet of what processing entails. Think of a calculator, the processing there is ... the calculating. We'll consider an example with arithmetic later on.

1.3 Output

So now what? We would like to produce some kind of output with the data that we had just collected. In keeping with the earlier example, we ask a user for their name, why not greet them!

```python
userName = input("What is your name?")
print("Hello, " userName) # Add output to greet the user
```

Output:

```
What is your name? Tony|
Hello,  Tony
```

Now, when the user inputs their data, we then process it in order to output a greeting message.

2 Moving forward

What happens if the user inputs something absurd? This will most certainly crash our program! We will explore **Error-handling** in the future, this is just something to keep in the back of our minds. This lesson is meant to build a basis for console input and output. In later lessons, we will start to acquire tools that will make our input/output much more interesting! Be on the lookout!

3 Key Terms

- **Console**:
- **Typecast**: The process of converting a variable from one datatype to another.
- **Error-handling**: The process of dealing with potential issues that make code prone to crashing.

- ____ **code block**: A section of statements related statements in a code file. "____" is a placeholder for the different kinds of code blocks which we will certainly explore later.

4 Availability

- In Lecture:
  - Wednesday: 10:00am - 11:00am
- PAL Sessions:
  - Monday: 2:00pm - 3:00pm
  - Wednesday: 9:00am - 10:00am
  - Wednesday: 2:00pm - 3:00pm

References

[1] Tony Gaddis. *Starting Out with Python*. Pearson, 5th edition, 2021.

[2] GeeksforGeeks. Python Tutorial — Learn Python Programming Language - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/python/python-programming-language-tutorial/. [Accessed 29-10-2025].

[3] Bradley N. Miller and David L. Ranum. Problem solving with algorithms and data structures using python. https://runestone.academy/ns/books/published/pythonds/index.html, 2014. Interactive online edition — accessed: 2024.

[4] Python Software Foundation. *Python: A dynamic, open source programming language*, 2025. Version 3.x documentation.

[5] W3Schools. W3Schools.com — w3schools.com. https://www.w3schools.com/python/default.asp. [Accessed 29-10-2025].