Introduction to Python Basics: Part 2

MARCH 2024

Contents

Input statement

Imagine you're talking to a computer, and you want to ask it a question, like "What's your name?" or "How old are you?" In Python, the input() statement lets you do just that!

It works like this: When you use the **input()** statement in your code, Python stops and waits for you to type something on the keyboard.

Whatever you type is like your answer to the computer's question. Then, Python takes what you typed and stores it so you can use it later.

Input statement

Here's how you use the **input()** statement:

```
name = input("What's your name? ")
```

So, when you run this code, Python will ask you "What's your name?" on the screen. Once you type your name and press Enter, Python will save that data value (your name) in that variable called **`name`**

Print statement

Now, let's say you want to talk back to the computer and tell it something. That's where the **print()** statement comes in handy!

The **print()** statement is like talking to the computer. You can use it to make the computer display messages or show you the results of calculations.

Print statement

Here's how you use the **print()** statement:

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

When you run this code, Python will show `**Hello, world!**` on the screen

You can also use **print()** to display the value of variables, like this:

```
print("Hello, my name is ", name)
```

Input/Print statements

So, with **input()** you ask the computer questions, and with **print()** you tell it things or show results on the screen. It's like having a conversation with the computer!

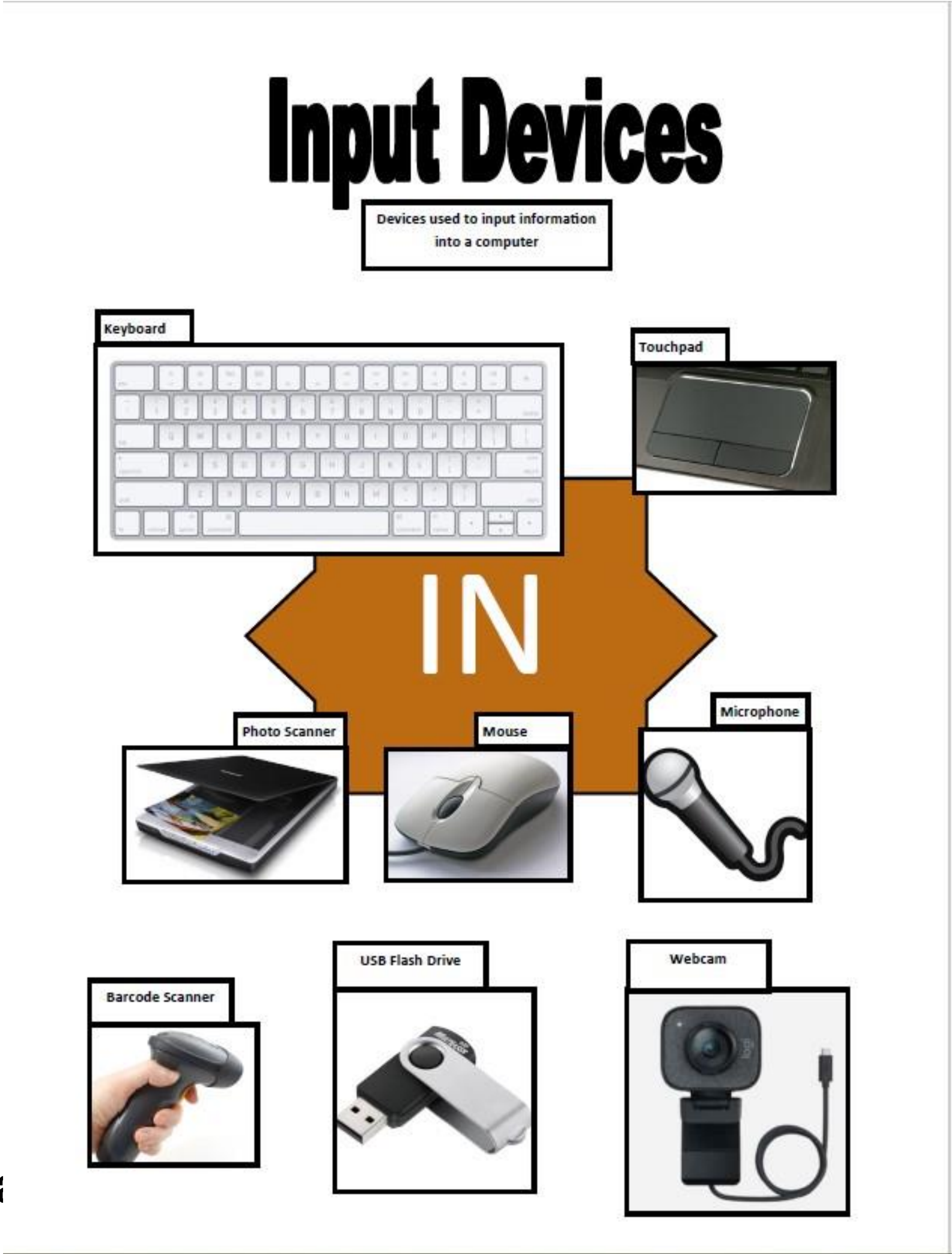
```
name = input("What's your name? ")
print("Hello, " + name + "!")

age = input("How old are you?")
print("Cool, " + name + "It is a pleasure talk
to a" + age + " years old person.")
```

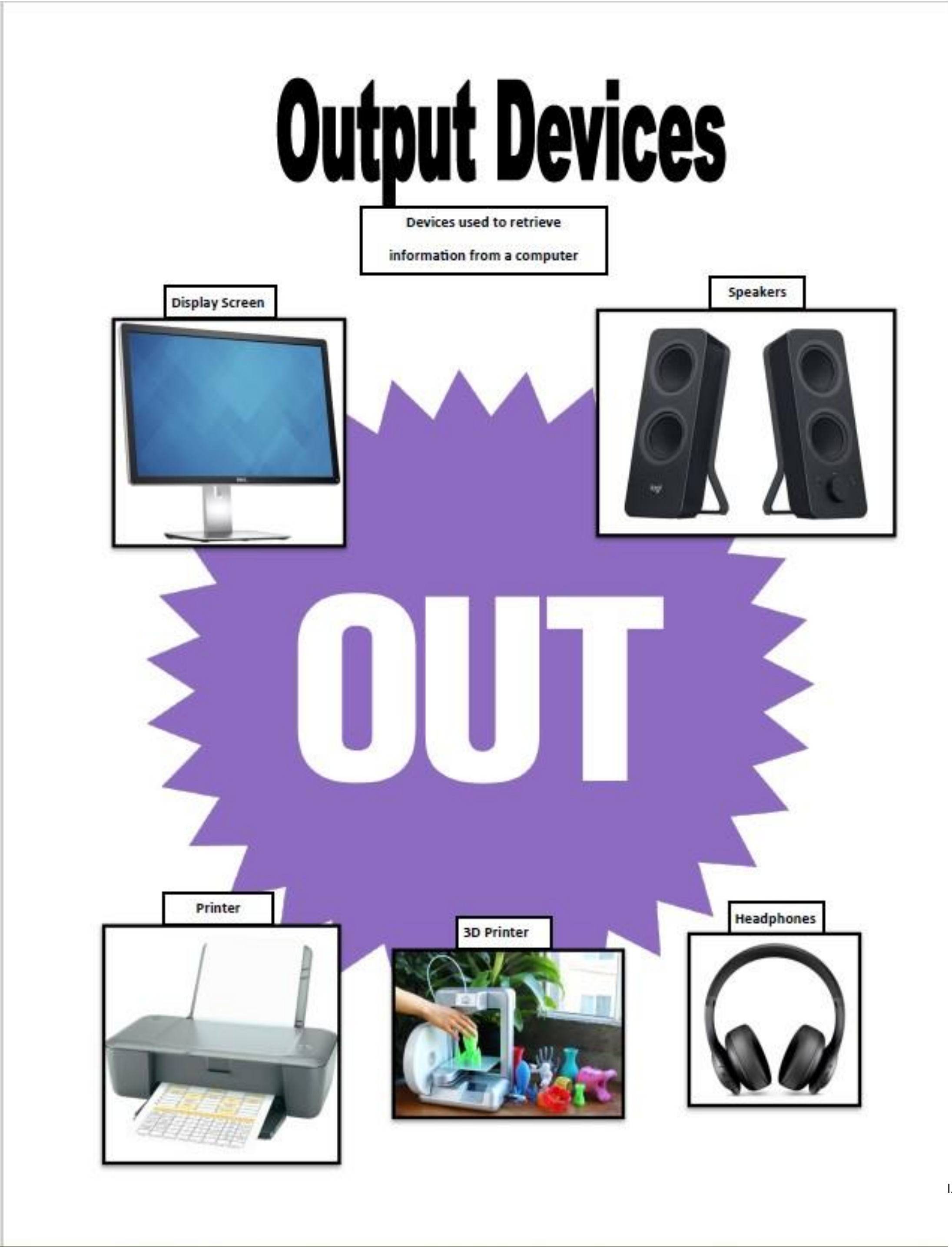
Input/Print statements

```python
name = input("What's your name? ")
print("Hello, " + name + "!")

age = int(input("How old are you? "))
print("You will be", age + 1, "years old next year.")

message = "Enter three numbers separated by spaces:"
num1, num2, num3 = input(message).split()
total = int(num1) + int(num2) + int(num3)
print("The total is:", total)
```

Input/Print statements

```python
name, age, ps5_price = "John Due", 15, 350.85698
print("My name is", name, "and I have", age, "years old")
print("My name is " + name + " and I have " + str(age) + " years old")
print("My name is {} and I have {} years old".format(name, age))
print(f"My name is {name} and I have {age} years old")
print("My PS 5 costs ${:.2f}".format(ps5_price))
print(f"My PS 5 costs ${ps5_price:.2f}")
print("apple", "banana", "orange", sep=" <---> ")
```


Input Devices
Devices used to input information into a computer
Keyboard
Touchpad
IN
Photo Scanner
Mouse
Microphone
Barcode Scanner
USB Flash Drive
Webcam


Output Devices
Devices used to retrieve information from a computer
Display Screen
Speakers
OUT
Printer
3D Printer
Headphones

Variables

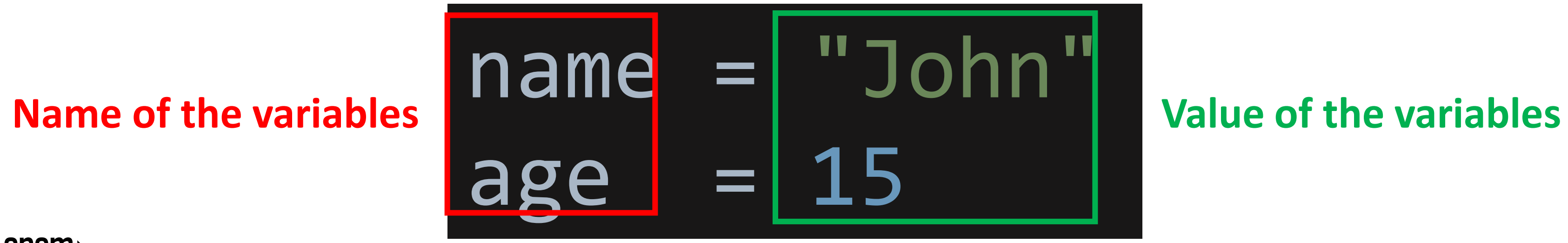
Imagine you're trying to solve a math problem. Let's say you want to find out how much money you'll have after buying a few items. Now, to keep track of different amounts of money, you would use a piece of paper to write down numbers, right?

Well, in programming, a variable is like that piece of paper. It's a way to store information so you can use it later in your code. But instead of writing numbers on paper, you give a name to your variable, and you can store different kinds of information in it, not just numbers.

Variables

- They are simply containers for storing data values
- Python has no command for declaring a variable, so you create one the moment you first assign a value to it
- For example, if you want to remember someone's name, you can create a variable called "name" and store their name in it. If you want to remember someone's age, you can create a variable called "age" and store their age in it.

Here's how you create a variable in Python:



```python
name = "John"
age = 15
```

Variables

```
first_name = "John"
second_name = "Due"
first_name = second_name
day_of_birth = 7
month_of_birth = 11
year_of_birth = 2009
one, two, three = 1, 2, 3
```

Constants

Constants are like variables, but their value doesn't change throughout the program

Imagine you have something that stays the same, like the value of pi (π) in math. It's always the same, right? Constants are used to store such unchanging values.

Here's a simple example using a constant for the value of pi:

```
PI = 3.14159
```

Constants

Python programmers follow a convention of using **UPPER CASE** letters and underscores ("_") to represent constants to make it clear that the value should not be changed

```
PI = 3.141592653589793
SECONDS_IN_AN_HOUR = 3600
ABSOLUTE_ZERO_IN_CELSIUS = -273.15
BASE_URL = "https://www.epam.com"
TOKEN_CHATGPT "12345678-1234-5678-1234-567812345678"
USERNAME = "my_user_name"
PASSWORD = "12345678
DAYS_OF_WEEK = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat",
"Sun"]
```

Python Operators

Arithmetic Operators

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
//	Floor division
**	Exponentiation

Python Operators

Assignment Operators

Operator	Example
=	x = 5
+=	x += 3 (same as x = x + 3)
-=	x -= 3 (same as x = x - 3)
*=	x *= 3 (same as x = x * 3)
/=	x /= 3 (same as x = x / 3)
%=	x %= 3 (same as x = x % 3)
//=	x //= 3 (same as x = x // 3)
**=	x **= 3 (same as x = x ** 3)

Python Operators

Comparison Operators

Operator	Name
==	Equal
!=	Not Equal
>	Greater Than
<	Less Than
>=	Greater Than or Equal to
<=	Less Than or Equal to

Python Operators

Logical Operators

Operator	Name
and	Returns True if both statements are true
or	Returns True if one of the statements is true
not	Reverse the result, returns False if the result is true

Python Operators

Identity Operators

Operator	Name
is	Returns True if both variables are the same object
is not	Returns True if both variables are not the same object

Membership Operators

Operator	Name
in	Returns True if a sequence with the specified value is present in the object
not in	Returns True if a sequence with the specified value is not present in the object

How to save and run a Python Script

Use always the extension ".py", e.g.: my_script.py

```
>>> python my_script.py
```

Module in Python is a file with Python code

How to install an IDE to use Python (PyCharm Community)

The steps how to install PyCharm Community you can find in Presentation Day 1 (in Google classroom)

There are other IDE that can be used:

- Visual Studio (or VS Code)
- Atom
- Eclipse
- Eric, and others.

If you don't want to use PyCharm, feel free to use any of these or others

Homework

Note: Add a screenshot for each completed task below on your Google Classroom

- Install PyCharm Community
- Clone the repository from https://github.com/ericrommel/ekids if not done yet
- Do the homeworks available on the repository for the Week 2: https://github.com/ericrommel/ekids/tree/main/homeworks/week-2
- After finishing the homework, send it back to GitHub following the steps:
    - git add, git commit, git push

Book Python-for-Kids

https://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/Python-for-Kids.pdf

Thank you!