



Welcome to the world of programming!

Every computer program, whether it's a video game, WhatsApp, or a car's artificial intelligence, is based on a very simple principle: **it receives data, processes it, and returns a result.**

In professional programming languages like Python, C, or JavaScript, the most basic way to show information on the screen is called *Print*, and the way to ask the user for data is called *Input*. Today we are going to learn how to do this using Snap! blocks.

Objectives of this practice:

- Understand what Data Input and Output are.
- Use Looks blocks to display messages on the screen.
- Use Sensing blocks to ask the user for information via the keyboard.
- Use the temporary variable `answer` to remember data.

Step by Step

1. Data Output: The "Say" block

In languages like Python, if we want the computer to say hello, we write: `print("Hello World!")`. In Snap!, we use the purple block from the **Looks** category.

- Go to the **Looks** category (purple color).
- Drag the say [Hello!] for (2) secs block to the scripting area.
- Change the text so the character says: *"Hello World! I am a program."*
- Click on the block to run it. Your first Output!

2. Data Input: Asking the user

Programs are boring if they don't interact. Let's ask the user to type something. In Python, this would be done with `input("What is your name?")`.

- Go to the **Sensing** category (light blue color).
- Drag the ask [What is your name?] and wait block and place it under the previous block.
- When you run it, you will see a text box appear on the screen. The program has paused waiting for your Input!

3. Temporary Memory: The Answer

When you type your name and press Enter, where does that information go? The computer saves it in a space in its memory. Snap! has a special block (a system variable) to store the last thing the user typed.

- In the same **Sensing** category, look for the oval block called `answer`.
- If you click the checkbox next to it, you will see the stored value on the screen.

4. Putting it all together: Text operators

We want the character to greet the user by name (for example: *"Hello, Maria!"*). To do this, we must "glue" or **concatenate** a fixed text with the `answer` variable.

- Go to the **Operators** category (green color).
- Look for the `join [hello] [world]` block.
- In the first slot, type *"Nice to meet you, "* (don't forget the space at the end!).
- In the second slot, drag the blue `answer` block inside.
- Put this entire green block inside a new purple `say` block.

THE CHALLENGE (Assignment exercise)

Create a new program (starting with the green flag block) that does exactly the following in order:

1. Greet the user and explain what the program is about (e.g., *"Hello, I am going to calculate what year you were born"*).
2. Ask for their name.
3. Greet them using their concatenated name.
4. Ask for their current age.
5. Perform a mathematical subtraction using the **Operators** blocks (current year minus the age answer) and tell them the year they were born using another `join` block.

Got it? Save your project and notify the teacher for review!

Did you know... what you just did with blocks, a professional programmer would write in Python like this:

```
name = input("What is your name?")
print("Nice to meet you, " + name)
```