

1. Previous Learning

Variables: A variable is a named space in memory. A program can store data in that memory space, read data from it, and change data within it. Variables can be used in flowcharts, block-based programming languages, and text-based programming languages.

Standard Data Types:

Integer: Whole numbers only (e.g. 23, 7, 52).

Real: Numbers containing decimal points (e.g. 22.52, 32.5, 17.22).

Character: A single letter, number, or symbol (e.g. "h", "0", "1").

String: Any combination of one or more letters, numbers, and symbols enclosed in quotation marks (e.g. "hello", "123", "YES!!", "001", "22.111").

Known Casting Functions in Python:

int() : Converts data inside the brackets to an integer (e.g. int("0")).

str() : Converts data inside the brackets to a string (e.g. str(12)).

float(): Converts data inside the brackets to a real number (e.g. float("21.42")).

2. Learning Objectives

- Learn the purpose of constants in algorithms.
- Write an algorithm that includes a constant.
- Follow algorithms that use the Boolean data type.
- Write programs in Python that use the Boolean data type.

3. Classroom Discussion:

A. The Boolean Data Type

Definition: Boolean data can only hold one of two values: true or false.

Syntax Rules:

Boolean values do not use quotation marks. Writing `"True"` creates a string, whereas writing `True` without quotes creates a Boolean.

Pseudocode: May be written in all lowercase (true, false), all uppercase (TRUE, FALSE), or title case (True, False). The style chosen must remain consistent throughout the algorithm.

Binary Values: In some languages, binary 1 represents True and 0 represents False.

Python Syntax: Python requires capitalized initial letters: True and False.

Flags:

A flag is a variable whose value indicates whether a specific event has occurred (True) or has not occurred (False).

Analogy: A sports referee or linesman raising a flag when an event happens (such as a foul in football) represents True, keeping the flag lowered represents False.

Storing comparison results directly: An expression like `age = (input1 >= 18)` automatically evaluates and assigns True or False to `age`.

Conditional shortcut: Writing `if value1:` is shorthand for `if value1 == True:` because the condition is met directly when `value1` is True.

Translating Checkpoint (In my own words):

A Boolean is like a light switch with only two positions: on (True) or off (False). When we use it as a flag, it acts as a reminder in the code that marks whether a specific goal or event has taken place.

B. Casting to Boolean

Definition: Casting is the process of converting data from one data type to another.

The `bool()` Function in Python:

`bool(0)` converts to False.

`bool(number)` converts to True for any number that is not 0 (such as `bool(1)` or `bool(-22)`).

`bool("")` (an empty string) converts to False.

`bool("string")` converts to True for any string containing one or more characters (e.g., `bool("true")`, `bool("FALSE")`, `bool("Hello")`).

C. Constants

Definition: A constant has an identifier (name) pointing to a space in memory that holds a value. Unlike a variable, the value stored in a constant cannot be changed later during the execution of the program.

Why do we need constants:

In mathematics, the value of pi (Pi) is always 3.141592... (rounded to 3.14) in formulas like 2 times pi times r. If a program allows this value to be modified mid-way, every subsequent calculation breaks. A constant locks this value in place.

Advantages of using constants?

- 1. Accidental Change Prevention:** Ensures values cannot be unintentionally altered while the program runs.
- 2. Typo Reduction:** Prevents typing errors across repetitive code (e.g., avoiding typing 1243 instead of 1234).
- 3. Efficiency of Maintenance:** If a value must change (such as a tax rate or grade boundary), you only have to modify it once at the declaration line rather than tracking down every occurrence in the code.
- 4. Memory Trade-off:** Constants occupy a small space in memory.

Structure of a Constant Declaration:

A constant declaration consists of four parts:

1. A keyword (e.g., constant).
2. An identifier (e.g., ten or cost).
3. An assignment symbol (=).
4. The value to be stored (e.g., 10 or 2.5).

Representation Across Formats:

Flowchart: Declared inside a rectangular process box at the beginning (e.g., constant ten = 10).

Pseudocode: Placed at the top on a single line: constant ten = 10.

Python: Python programmers use ALL UPPERCASE IDENTIFIERS (e.g., TEN = 10, COST = 2.5) to remind developers not to modify these identifiers.

D. Troubleshooting Friction

Consider this flawed Python segment:

```
```python
Intended: Check if the user entered text, then flag approval
username = input("Enter your username: ")
has_name = bool(username)

if has_name == "True":
 status = "Active"
```
```

Diagnosing the Friction: The variable `has_name` holds a Boolean value (True or False). The conditional statement erroneously compares it to the string "True" enclosed in quotation marks. Because a Boolean True is never identical to the string "True", this check fails even if a name was entered.

Correction: Remove the double quotes: `if has_name == True:` or streamline it to `if has_name:`.

4. Questions and Answers

Q1: What is the main operational difference between a variable and a constant?

A: A variable holds data in a memory location that can be read, updated, or changed while a program runs. A constant points to a memory location whose stored value cannot be modified later during program execution.

Q2: What two features do both constants and variables share?

A: Both constants and variables possess an identifier (a name) and occupy an allocated space in computer memory to store a value.

Q3: What values can be stored in a Boolean data type?

A: A Boolean data type can only store one of two possible values: True or False. It cannot represent any other value.

Q4: How does a computer distinguish between the data "True" and True?

A: The value "True" surrounded by quotation marks is treated as a string made of characters. The value True written without quotation marks is recognized by the computer as a Boolean value.

Q5: What is a "flag" in computer programming?

A: A flag is a variable holding a Boolean value that signals whether a specific condition or event has occurred (True) or has not occurred (False). It allows a program to remember a status check without recalculating it later.

Q6: What are the four structural parts of a constant declaration statement?

A: A constant declaration contains a declaration keyword (such as constant), an identifier name, an assignment symbol (=), and the value to be assigned.

Q7: How do Python programmers visually designate that a variable should act as a constant?

A: Because standard Python variables can technically still be overwritten, programmers write the identifier name in all capital letters (e.g., COST = 2.5) to signal that its value must remain fixed.

Q8: What rule determines the Boolean result when casting an integer using bool()?

A: Casting the integer 0 yields False, whereas casting any number other than zero (including positive and negative integers) yields True.

Q9: What is the Boolean result of casting the string "FALSE" using bool("FALSE"), and why?

A: It evaluates to True. In Python, casting any non-empty string containing one or more characters produces True, regardless of the word inside the quotes.

Q10: Why does placing constant declarations at the very start of a program help programmers?

A: Declaring constants at the top makes them easy to locate, inspect, and update in a single place if values change, without having to review or edit lines deep in the code logic.

5. Second-Order Thinking Skills Questions (Analyze, Evaluate, Create)

Q1 Examine the following pseudocode segment:

```
``text
INPUT itemWeight
OUTPUT "The cost is", itemWeight * 3.5
INPUT secondWeight
OUTPUT "The cost is", secondWeight * 3.5
...

```

Identify which items should be converted into constants and analyze how this change improves code maintenance if the rate increases to 4.5.

Answer:

The numerical factor 3.5 should be declared as constants at the beginning of the algorithm (e.g., constant rate = 3.5).

Without constants, changing the pricing rate requires finding and editing every instance of 3.5 in the calculation statements.

With constants, the programmer updates the value once at the top (constant rate = 4.5), automatically applying the update across all calculations without risking typing errors.

Q2 A student writes the following validation check to detect whether a user left an input blank:

```
```python
user_response = input("Enter your username: ")
check = bool(user_response)
if check == False:
 print("No data")
...
```
```

Evaluate whether this logic correctly detects empty input and explain how Python's casting rules handle the input.

Answer:

The logic works correctly. When the user presses enter without typing anything, `input()` produces an empty string `""`.

Based on Python casting rules, `bool("")` converts an empty string directly to `False`. If the user enters any characters, `bool()` converts the string to `True`. Therefore, checking **if check == False:** accurately identifies an empty submission.

Q3 A cinema ticket system gives discounts based on age. Write a Python algorithm that asks for an age, uses a comparison directly to set a Boolean flag named `is_child` (true if under 16), and outputs a message based on the flag.

Answer:

```
```python
age = int(input("Enter your age: "))

is_child = (age < 16)

if is_child:
 print("Child discount applied")
else:
 print("Standard admission rate")
...
```
```