# #8. Pseudocode

14/05/2025

- Pseudocode is a relaxed format of code for human-only readibility.

example

INFORMAL Pseudocode

```
For a function that determines which number in
a collection has the greatest value.
    - Given a collection of numbers.
  □ Iterate through the collection of numbers one-
    by-one
      -> save the first value as the starting value
      -> for each iteration, compare the current value
         with the starting value.
           - if the current number is greater
               - reassign the ~~current~~ starting value as the current
                 value
           - otherwise, if the current value is smaller or equal
               - move to the next value in the collection
```

Note: I wanted to write this out in code.

```
collection = [1, 2, 100, 2, 5, 3, -5, 4, 1]

def my_val(collection):
    saved_val = collection[0]

    for num in collection:
        if num > saved_val:
            saved_val = num
        else:
            continue
    print(saved_val)

my_val(collection)
```

remove indent.

14/05/2025

- Two layers to solving any problem:
  1. The logical domain layer
  2. The syntactical programming language layer.

## Formal Pseudocode

- Keywords to assist writing Pseudocode

| KEYWORD | Meaning |
|---|---|
| START | start of the program |
| SET | setting a variable |
| GET | get user input |
| PRINT | display output to user |
| READ | retrieve a value from a variable |
| IF/ELSE IF/ELSE | show conditional branches in logic |
| WHILE | show looping logic |
| END | end of the program. |

### example

```
START
# Given a collection of integers called "numbers"

SET iterator = 1
SET savedNumber = values within numbers collection at space 1.

WHILE iterator <= length of numbers
    SET currentNumber = value within numbers collection at space "iterator"
    IF currentNumber > savedNumber
        savedNumber = currentNumber

    iterator = iterator + 1

PRINT savedNumber
```

13/05/2025

## Translating Pseudocode to Program Code

- start iterator at index 0 as list indexing in Python starts ~~at~~ at 0.

* • ~~iterator~~ Use less than (<) rather than <= as you want the loop to stop once the element at the last index has been accessed. len(numbers) is 1 greater than the last index

```
def find_greatest(numbers):
    iterator = 0
    saved_number = numbers[iterator]

    while iterator < len(numbers):
        current_number = numbers[iterator]
        if current_number > saved_number
            saved_number = current_number

        iterator += 1

    return saved_number
```

* See point above

REMEMBER: Pseudocode is a guess at the solution; there is no verification that the logic is correct. Need to program it to verify the logic.

- For more complex problems, cannot Pseudocode the problem as likely to make a mistake and it'll take a long time to arrive at 'final' correct logic.

HINT: USE FLOWCHARTS (see next Assignment)