



# Quiz 2 Review Session

# Content on Quiz 2

- Variables
- While loops
- Lists

Disclaimer: We haven't seen the quiz; this review session covers the main topics in the unit.

# Variables

Variable declaration:

<name>: <type>

Example:

students: int

# What is Variable assignment?

Binds a new value to a variable name in memory

Example:

```
students = 300
```

## Common variable errors

`UnboundLocalError` – Occurs when attempting to access a variable that is declared in a function but not yet initialized.

`NameError` – Occurs when attempting to access a variable that has not been declared. Commonly from typos or renaming a variable and not updating all accesses

# While Loops

True or False questions:

1. In a while loop, the condition is checked before the loop executes
2. A while loop executes its body at least once
3. If a condition in a while loop never becomes False, the loop will continue indefinitely

# Diagramming a Nested while Loop

```
1  def triangle(n: int) -> None:
2      i: int = 1
3      line: str
4      while i <= n:
5          line = ""
6          while len(line) < i:
7              line += "*"
8          print(line)
9          i += 1
10
11
12  triangle(2)
```

# Function writing using while loops

- The `count_down` function should accept a `int` parameter called `start` and return `None`
- The function must decrement `start` by 1 until `start` reaches a value equal to or less than 0
- Every time we decrement `start`, we want to print its current value before decrementing
- A docstring should be included: Counts down from a given number to 0
- You should explicitly type all variables, parameters, and return types.

Examples:

1.) `count_down(3)`, RV is `None`, Output: `3 // 2 // 1 // 0`



# Solution

```
def count_down(start: int) -> None:
    """Counts down from a given number to 0"""
    while start >= 0:
        print(start)
        start -= 1
    return None
```

# Lists

True or False questions:

1. If we set `my_list=["a", "b", "c"]`, then `my_list.pop(2)` changes the list to equal `["a", "c"]`
2. Lists' elements are stored in the heap.
3. Lists can contain elements of any data type we've learned about so far (str, int, float, list)
4. One list cannot contain multiple different data types
  - a. (e.g. `this_list = [1, "a", 3.6]`)

```
1 def summation(list_1: list[int]) -> None:
2     """Converting each element into the sum of itself and all previous elements in the list."""
3     i: int = 0
4     while i < len(list_1):
5         if i != 0:
6             list_1[i] = list_1[i] + list_1[i - 1]
7         i += 1
8
9
10 values: list[int] = [1, 2, 3, 4]
11 summation(values)
12 print(values[len(values) - 1])
```

```

1 def summation(list_1: list[int]) -> None:
2     """Converting each element into the sum of itself and all previous elements in the list."""
3     i: int = 0
4     while i < len(list_1):
5         if i != 0:
6             list_1[i] = list_1[i] + list_1[i - 1]
7         i += 1
8
9
10 values: list[int] = [1, 2, 3, 4]
11 summation(values)
12 print(values[len(values) - 1])

```

Output  
10

Stack

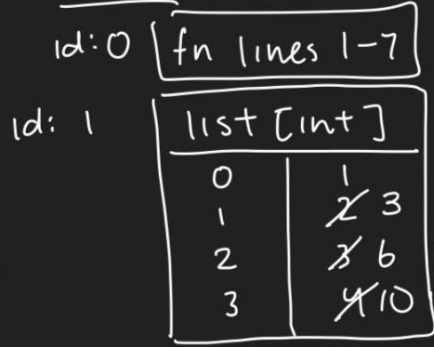
Globals    summation [id: 0]  
          values [id: 1]

summation

RA [11]  
RV [None]

list\_1 [id: 1]  
i [0] x [2] 3 4

Heap



Questions?

# Other Resources!

- Practice quiz on the course site with answers and explanations
  - We would recommend trying the problems out on your own, then checking your answers
- Tutoring
  - Thursday 3 - 5 in FB 141
- Office Hours
  - Tomorrow and Friday 11 - 5 in SN008