13th March. 2023

Conditional Statement

If statements

Used to conditionally do something. the structure/ conditions are:

else if

- if ↓ once start
- elif ↓ as many middle
- else ↓ once ending

Loops

- For loop ÷ Iterate through items
- While loop ÷ Iterate through items till the condition is true

Commands : break & continue are used

break → stops the loop

continue → skips an element based on the condition and continues iterating

List Comprehensions

- They are nested for loops

conditional (optional)

```
x = [1, 2, 3, 4, 5]
y = [item for item in x if x > 2]
Print y ;   3, 4, 5
```

iterable

expression

List comprehension for nested for loops

For the nested for loop below, which list comprehension is correct?

```python
# Nested for loops
words = [["Am", "ate", "ATOM", "apple"], ["bE", "boy", "ball", "bloom"]]
small_words = []
for letter_list in words:
    for word in letter_list:
        if len(word) < 3:
            small_words.append(word.lower())
print (small_words)
```

```
['am', 'be']
```

- ○ `[word.lower() if len(word) < 3 for word in letter_list for letter_list in words]`
- ○ `[word.lower() for word in letter_list for letter_list in words if len(word) < 3]`
- ✓ `[word.lower() for letter_list in words for word in letter_list if len(word) < 3]`

Show answer

Python syntax is usually very straight forward, so the correct answer involves just directly copying the statements from the nested for loop from top to bottom!

- ○ `[word.lower() if len(word) < 3 for word in letter_list for letter_list in words]`
- ○ `[word.lower() for word in letter_list for letter_list in words if len(word) < 3]`
- ✓ `[word.lower() for letter_list in words for word in letter_list if len(word) < 3]`