COMP10001 Foundations of Computing
Semester 2, 2019

Tutorial Questions: Week 4

— VERSION: 1474, DATE: AUGUST 15, 2019 —

Discussion

1. What is "Boolean"? What values does it store? Can other types be converted to it?
2. For each of the following, identify whether it is: (a) a Boolean value; (b) a relational operator; or (c) a logical operator. What do they do?

`==`		`>`		`False`	
`!=`		`and`		`<=`	
`or`		`>=`		`not`	
`True`		`<`			

3. How do we use an if statement? What are the variants? How do we know what is contained inside it and what is after?

   Now try Exercises 1 - 4

4. What is a "Sequence"? What sequences have we seen so far?
5. What is indexing? How can you do it?
6. What is slicing? How can you do it?
7. **Bonus question:** How do you change the "step size" of a slice?

   Now try Exercises 5 & 6

Exercises

1. Evaluate the following truth expressions:

   (a) `True or False`

   (b) `True and False`

   (c) `False and not False or True`

   (d) `False and (not False or True)`

2. For each of the following if statements, give an example of a value for `var` which will trigger it and one which will not.

   (a) `if 10 > var >= 5:`

   (b) `if var in ("VIC", "NSW", "ACT"):`

   (c) `if var[0] == "A"and var[-1] == "e":`

   (d) `if var:`

3. What's wrong with this code? How can you fix it?

```
letter = input("Enter a letter: ")
if letter == 'a' or 'e' or 'i' or 'o' or 'u':
    print("vowel")
else:
    print("consonant")
```

4. What's wrong with this code? How can you fix it?

```
eggs == 3
if eggs = 5:
    print("spam")
else:
    print("not spam")
```

5. Evaluate the following given the assignment `s = "pythonisation"`

   (a) `s[1]`
   (b) `s[-1]`
   (c) `s[2:4] + s[6:8]`
   (d) `s[25]`
   (e) `s[25:]`
   (f) `s[-7:-3]`
   (g) `s[:-3]`
   (h) `s[::2]`
   (i) `s[::-1]`

6. Evaluate the following given the assignment `lst = [4, ("green", "eggs", "ham"), False]`

   (a) `lst[2]`
   (b) `lst[1][-2]`
   (c) `lst[1][-2][:3]`

Problems

1. Write a program which asks the user for two numbers and an operator out of +, -, / and * and performs that operation on the two numbers, printing the result.
2. Write a program which asks the user for their name and prints a shortened version consisting of the first three letters and then every second letter in the rest of the word.
3. Write a program which asks the user to write a sentence and checks whether it is a "correct sentence". To do this, it should check that the first letter is capitalised and the last character is a full stop.