COP 3035

Intro Programming in Python

Summer 2024

Lecture 8 – part 1

Lab 4 - Due Date: 06/10/2024
Homework 2 - Due date: 06/07/2024
Homework 3 - Due date: 06/14/2024

Lecture 8 – part 2

Review

Conditional Statements

- If / else
- If / elif / else

If/else statement

- Syntax of the `if/else` statement

```
if True:
    # do something
    print(a)
else:
    # do something else
    print(b)
```

If/elif/else statement

- Syntax of the **`if/else`** statement

```
if some_condition:
    # do something
    print(a)
elif some_other_condition:
    # some other condition
    print(b)
else:
    # do something else
    print(c)
```

Comparison operators (a= 3, b=4)

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	`(a == b) is not true.`
!=	If values of two operands are not equal, then condition becomes true.	`(a != b) is true`
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	`(a > b) is not true.`
<	If the value of left operand is less than the value of right operand, then condition becomes true.	`(a < b) is true.`
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	`(a >= b) is not true.`
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	`(a <= b) is true.`

Chained Comparisons

Expression Type	Example	Equivalent Boolean Expression	Description
Chained Comparisons	`A <= B <= C`	`A <= B and B <= C`	Checks if A is less than/equal to B and B is less than/equal to C.
	`X >= Y != Z`	`X >= Y and Y != Z`	Checks if X is greater than/equal to Y and Y is not equal to Z.
and & or	`A < B and B < C or C == D`	_	Checks if A<B and B<C, or if C is equal to D.
Using not	`not (A == B)`	`A != B`	Returns True if A is not equal to B.
	`not (A > B and C > D)`	`A <= B or C <= D`	Checks if A is less than/equal to B or C is less than/equal to D.
Nested Conditions	`(A < B or C > D) and E == F`	_	Checks if A<B or C>D, and if E is equal to F.
Chaining with not	`not A < B < C`	`not(A < B and B < C) or A >= B or B >= C`	Negates the entire chained comparison.
Multiple Operators	`A < B < C or D != E and not F > G`	_	A combination of chaining, and, or, and not.

Exercise

Consider a number:

- Determine if it is **positive or negative**.
- Additionally, determine if the number is **even or odd**.
- If the number is **zero**, simply state that the number is **zero**.

Lecture 8 – part 3

For Loops

for loops

- We can use for loops to execute a block of code for each iteration.
- Many objects in Python are **"iterable",** meaning we can iterate over each element.
- Iterate over every item in a **list**,
- Iterate over every character in a **string**,
- Iterate over every key in a **dictionary.**

for loops

- **Syntax of a for loop:**

```
my_iterable = [1,2,3]
for item in my_iterable:
    print(item)
```

Lecture 8 – part 4

While loops

while loops

- While loops continue to execute a block of code **while** some condition remains **True**.

Syntax of the while loop:

```
while some_condition:
    # Do something
else:
    # Do something different
```

break, continue, pass

break – Breaks out the current closest enclosing loop.

continue – Goes to the top of the closest enclosing loop.

pass – Does nothing at all.