

ED5340 - Data Science: Theory and Practise

L4 - Conditional statements and Loops

Ramanathan Muthuganapathy (<https://ed.iitm.ac.in/~raman>)

Course web page: <https://ed.iitm.ac.in/~raman/datascience.html>

Moodle page: Available at <https://courses.iitm.ac.in/>

Decision control (if - elif - else)

colon and indentation

if condition:

↔ st1

st2

if condition:

st1

st2

else:

st1

st2

if condition1:

statements

elif condition2:

statements

elif condition3:

statements

else:

statements

Relational operators

almost same as C

- <
- >
- <=
- >=
- ==
- !=

Ramanathan Muthuganapathy

Demo using L4_if_statements.py

Some nuances than what C offers

- $a < b < c$
- $a == b == c$
- $a != b != a$ (Will this condition return True or False?)
- Any non-zero number is treated as True (0 as False) - Same as C

**REDO the largest of three numbers
using ONLY relational operators**

Logical operators

and, or, not (NOT &&, ||, ! but works similarly)

- cond1 and cond2 - returns True only if both are true
- cond1 or cond2 - returns True if even one of the is true.
- NOTE: We can replace 'condition' with any 'valid expression'

HW: Redo the largest one including logical operators.

HW: Get to know how 'not' operator works.

HW: Also, get to know about the functions any() and all().

Loops

while and for (No do-while) - there are differences

while condition:

statement

.....

.....

.....

while condition:

statement

.....

else:

statement

.....

Loops

for loop - this iterates over each element in a sequence (string, range, list, tuple etc.)

for ele in seq:

statement

.....

.....

.....

for ele in seq:

statement

.....

else:

statement

.....

Key differences between for and while

- **for** iterates over the iterable (string/ list etc..), where as **while** does not
- **while** uses a condition where as **for** does not

Ramanathan Muthuganapathy

Demo using L4_loops_example.py

range() function

similar to for (exp1; exp2; exp3)

- range(10) - generates numbers from 0 to 9
- range(5,15) - generates numbers from 5 to 14 (Note this)
- range(0, 10, 2) - generates numbers from 0 to 10 in steps of 2
- In general, range(start, stop, step) - numbers from 'start' up to 'stop' (but excluding it) and incrementing/decrementing according to 'step'

**Back to the demo using
L4_loops_example.py**

HW: Find about break, continue, and pass statements and use them in a program.

HW: Print numbers 1 to 10 on the same line, breaking out of an infinite loop.

HW: Print all unique combinations of 1, 2 and 3.

Conditional expression

similar to the ternary operator ? : in C

<expr1> if condition else <expr2>

Equivalent to

if condition:

 expr1

else:

 expr2

Ramanathan Muthuganapathy