More Loops

CS 1111
Introduction to Programming
Spring 2019

[The Coder’s Apprentice, §7]
Loops in Python

- **while** loop – a condition-controlled loop
- **for** loop using the **in** operator with a list
- **for** loop using the **range** operator – a count-controlled loop
- Nested loops
While Loop vs. For Loop

while <condition>:
    <statements>
    <handler>

To break the condition such that the loop terminates

for <iterate_var> in <collection>:
    <statements>

for <iterate_var> in range(start, stop, step):
    <statements>

Iterate up to (or down to) this but not include

default = 1
While Loop vs. For Loop

**While loop**

- Use when you don’t know the number of iterations
- Need to explicitly handle the termination

```python
i = 5
while i > 0:
    print(i * i * i)
    i -= 1
```

**For loop**

- Use when you know the number of iterations
- Loop stops as it reaches the end of range, or the end of list

```python
for i in range(5, 0, -1):
    print(i * i * i)
```
Nested Loops

outer loop <condition>:
  inner loop <condition>:
    <statements>
    <handler>

- Nest with **while** loops, **for** loops, or mix **while** loops and **for** loops

- **break** and **continue** may be used to handle the flow of execution
  - **break** and **continue** are unnecessary with proper loop conditions
Summary

• Must know (based on exam2 topic list, as of 03/04/2019)
  • while loop
  • for loop
  • Using loops with collections

  • range(end)
  • range(start, end)
  • range(start, end, step)

• Nested loop
• Using nested loop with collections