

Week 4—Tuesday

Loops

Loops

What is a Loop?

- A way to repeat a block of code for a specific number of times.
- i.e., A way to iterate over a collection of items.
- bad: infinite loop

When to use loops?

- When you want to add 2 to every number in a list.
- When you want to print every item in a list.
- When you want to sum all the numbers in a list.

Types of Loops

- `for` loop
 - for each item, do something.
- `while` loop
 - While some condition is true, do something.

Syntax

```
for item in collection:
    # do something

while condition:
    # do something
```

Example: For Loop

```
items = ["apple", "banana", "cherry", "date", "elderberry", "fig"]

for item in items:
    print(f"item: {item}!")
```

```
item: apple!
item: banana!
item: cherry!
item: date!
item: elderberry!
item: fig!
```

Example: For Loop, II

- Sometimes we want the index of the item we are looping over. Use `enumerate` for this.

```
items = ["apple", "banana", "cherry", "date", "elderberry", "fig"]

for index, item in enumerate(items):
    print(f"index: {index}, item: {item}!")
```

```
index: 0, item: apple!
index: 1, item: banana!
index: 2, item: cherry!
index: 3, item: date!
index: 4, item: elderberry!
index: 5, item: fig!
```

Example: While Loop

```
import random

counter = 0
items = ["apple", "banana", "cherry", "date", "elderberry", "fig"]
results = []

while counter < 5:
    results.append(random.choice(items))
    counter += 1
print(results)
```

['banana', 'banana', 'apple', 'date', 'cherry']

Example: While Loop, II

```
items = ["apple", "banana", "cherry", "date", "elderberry", "fig"]
counter = 0

while counter < 3:
    print(f"Item is {items[counter]}")
    counter += 1
```

Item is apple
Item is banana
Item is cherry

1-line loops

- Sometimes you can write a loop in one line.
- But only for for loops, not while loops.

```
from datetime import datetime, timedelta

ds = [
    {"name": "John", "age": 25},
    {"name": "Jane", "age": 30},
```

```

    {"name": "Jack", "age": 35},
]
names = [d["name"] for d in ds]

date = datetime.now()
dates = []

while date < datetime.now() + timedelta(days=10):
    dates.append(date)
    date += timedelta(days=1)

print(names)
formatted_dates = [datetime.strftime(date, "%Y-%m-%d") for date in dates]
print("dates", formatted_dates)

```

```
['John', 'Jane', 'Jack']
```

```
dates ['2024-09-16', '2024-09-17', '2024-09-18', '2024-09-19', '2024-09-20', '2024-09-21', '2024-09-22']
```

Nested Loops

- We can have loops inside loops.
- This can be particularly useful when working with 2D data.
- Be careful, though, as it can be hard to read.
- Make your naming conventions clear.

Nested Loops, 2d items

- 2-dimensional items:

```

items = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

objs = [
    {"name": "John", "cities": ["New York", "Los Angeles", "Chicago"]},
    {"name": "Jane", "cities": ["San Francisco", "Seattle", "Portland"]},
]

```

Nested Loops, 2d items

```

items = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

objs = [
    {"name": "John", "cities": ["New York", "Los Angeles", "Chicago"]},
    {"name": "Jane", "cities": ["San Francisco", "Seattle", "Portland"]},
]

for i1, row in enumerate(items):
    print(f"Row {i1}")
    for i2, item in enumerate(row):
        print(f"Item {i2}: {item}")

```

```

Row 0
Item 0: 1
Item 1: 2
Item 2: 3
Row 1
Item 0: 4
Item 1: 5
Item 2: 6
Row 2
Item 0: 7
Item 1: 8
Item 2: 9

```

Break, Continue, Else

- **break**: stop the loop.
- **continue**: skip the rest of the loop and go to the next iteration.
- **else**: run when the loop is done without **break**.

Example: Break

- Stop the loop when we find the item “date”.

```

items = ["apple", "banana", "cherry", "date", "elderberry", "fig"]
new = []

for item in items:
    if item == "date":
        break

```

```
new.insert(0, item)
print(new)
```

['cherry', 'banana', 'apple']

Example: Continue

```
items = ["apple", "banana", "cherry", "date", "elderberry", "date", "fig"]
new = []

for item in items:
    if item == "date":
        continue
    new.insert(0, item)
```

Example: Else

- Run the else block if the loop completes without breaking.

```
items = ["apple", "banana", "cherry", "date", "elderberry", "date", "fig"]
new = []

for item in items:
    if item == "date":
        new.insert(0, item)
    new.append(item)
else:
    print("Done", new)
```

Done ['date', 'date', 'apple', 'banana', 'cherry', 'date', 'elderberry', 'date', 'fig']