Lists and Loops

In this lecture, we will cover

- Python List Structure
  - Python Tuple Structure
- For Loops

Opening Question

How are you feeling about the coding side of things?
0 - I am not coding yet
3 - I am comfortable and still learning
5 - We need to pick up the speed/do more

Your future in CS

I used to include this on my slides, but since these slides have changed - going to just leave it up here for every notebook. I get a lot of questions about more programming courses, the concentrations, and minors in computer science. Here is a brief reminder.

CS 164 – Next Course In Sequence, also consider CS 220 (math and stats especially)

- CO Jobs Report 2021 – 77% of *all* new jobs in Colorado require programming
- 60% of all STEM jobs requires *advanced* (200-300 level)
- 31% of all Bachelor of Arts degree titled jobs also required coding skills
- 2016 Report found on average jobs that require coding skills paid $22,000 more

- Concentrations in CS:

  - Computer science has a number of concentrations. General concentration is the most flexible, and even allows students to double major or minor pretty easily. The others are specialized paths in CS. Software Engineering, Computing Systems, Human Centered Computing (Psychology+CS), Networking and Security, Artificial Intelligence, and Computer Science Education.
  - Minors: We have three minors. Minor in computer science (choose your own adventure minor), Minor in Machine Learning (popular with stats/math), and Minor in Bioinformatics (Biology+Computer Science)

Python Lists

At this point, anytime we wanted to store information

- We created a variable

- Stored information in it

For example:

```
item1 = "Oranges"
item2 = "Pineapples"
item3 = "Mangos"
```

Not only is this tedious, but not wise! Imagine needing to keep track of 100 items or 1000 items?

Introducing List

A python list is a **sequential** structure designed to hold more than one item in order!

In [ ]:
```
shopping = ["Oranges", "Pineapples", "Mangoes"]
grades = [3, 4, 4, 3]

print(shopping)
print(shopping[0]) ## notice we start at 0!
print(shopping[1])
print(shopping[2])

print(grades)
```

```
['Oranges', 'Pineapples', 'Mangoes']
Oranges
Pineapples
Mangoes
[3, 4, 4, 3]
```

Changing elements

You can both access, and change an element at a location!

But if you need to add an item, you use the `.append(val)` function.

In [ ]:
```
print(shopping)

shopping[0] = "Apples"

print(shopping)

shopping.append("Lychee")

print(shopping)
```

```
['Oranges', 'Pineapples', 'Mangoes']
['Apples', 'Pineapples', 'Mangoes']
['Apples', 'Pineapples', 'Mangoes', 'Lychee']
```

Looping over elements

We can use a while loop to loop over elements

In [ ]:
```
index = 0
while index < len(shopping):  # len gives us the total items in a list the "length" of
    print(shopping[index])
    index += 1
```

```
Apples
Pineapples
Mangoes
Lychee
```

for loop - a better way

Introducing the for-loop (called for-each in other languages).

- It looks at every element in a list
- grabs each element, storing it in a variable
  - you can use that variable in the loop

```
for val in lst:
    # i can now use val in the loop
```

In [ ]:
```
for item in shopping:
    print(f"Need to buy: {item}")

# also, if i need to loop over numbers, range function exists!
# inclusive of the first position, exclusive of the second
for val in range(0, 10):
    print(val, end = ' ')
```

```
Need to buy: Apples
Need to buy: Pineapples
Need to buy: Mangoes
Need to buy: Lychee
0 1 2 3 4 5 6 7 8 9
```

In class activity

1. Create a list made up with the names of people as your table
2. Print out each persons name in the list
   - use the for loop, not direct access

In [ ]:
```
## put your code here
```

```
Biggs Wedge Cid Boko Mog Sarah
```

List Functions

Lists have a number of functions that help make life easier.

A few common functions are:

- len(list) - gives you the length / number of items in the list
- min(list) - gives you the smallest item (works on strings also!)

- max(list) - gives you the largest item (also works on strings)
- sum(list) - adds all the items together - does *NOT* work on strings
- + - the plus operator is used for adding two lists and returning a single list of items

In [ ]:
```
shopping_length = len(shopping)
print(f"The length of the shopping list is: {shopping_length}")

smallest_grade = min(grades)
largest_grade = max(grades)
sum_grade = sum(grades)
print(f"The smallest grade is {smallest_grade}, the largest: {largest_grade}, and the

oddList = shopping + grades

print(oddList)
```

```
The length of the shopping list is: 4
The smallest grade is 3, the largest: 4, and the sum is 14
['Apples', 'Pineapples', 'Mangoes', 'Lychee', 3, 4, 4, 3]
```

Splitting Lists

We also have the split operators **[]**, just like we do for Strings

In [ ]:
```
sublist = shopping[1:] ## gets the item in possition one, and returns the sublist
print(sublist)
```

```
['Pineapples', 'Mangoes', 'Lychee']
```

In [ ]:
```
print(shopping[1:3])
print(shopping[:3])
print(shopping[0:3]) ## this is the same as [:3]
```

```
['Pineapples', 'Mangoes']
['Apples', 'Pineapples', 'Mangoes']
['Apples', 'Pineapples', 'Mangoes']
```

In Class Activity

- Create a list of grades (all ranging)
- print only the grades that are 3 or above
- print some stats about the grades (min grade, max grade, average grade?)
- print a list of only the last two grades
  - remember len(list)-1 gives you the last location!

In [ ]:
```
grades = [4, 3, 2, 1, 4, 3, 2, 3, 4]
## your code here
```