SDEV 1001

Programming Fundamentals

Arrays and Loops - 1

Expectations - What I expect from you

- No Late Assignments

- No Cheating

- Be a good classmate

- Don’t waste your time

- Show up to class

Agenda

On the right is what we will cover today.

- What is a List (Python) or Array (other languages)?
- Introduction to Lists in Python
- Creating a List
- Accessing List Items
- Modifying List Items
- Slicing Lists
- Getting the Length of a List
- Out-of-Range Index
- Summary

What is a List (Python) or Array (other languages)?

A list (or array) is a collection of items stored in a single variable.

Lists can hold multiple values of different data types, such as strings, integers, and floats.

- 99.9% of the time you'll keep the same data type in a list.

Analogy it's a like a grocery list where you can store what need to buy at the grocery store.

Introduction to Lists in Python

Introduction

- Lists are used to store multiple items in a single variable.
- They are versatile and can hold any data type.
- Common in real life: todo lists, shopping lists, rankings, etc.

Why are lists important?

- Lists (or arrays in other lang) are the building blocks of data management in every programming language.
- This data structure is used every single program you will write after this course.

Creating a List

To create a list, use square brackets `[]` and separate items with commas.

- Lists can be printed directly.

Example:

```
colors = ['red', 'blue', 'green', 'yellow']
print("Available colors:")
print(colors)
```

Accessing List Items

To access items in a list, use the index inside square brackets `[]` next to the list name.

- Indexing starts at 0.
- Negative indices access items from the end.

Example:

```
animals = ['cat', 'dog', 'rabbit']
print("First animal:", animals[0])
print("Last animal:", animals[-1])
```

Modifying List Items

Just like you can access items, you can also modify them by assigning a new value to a specific index.

Example:

```
numbers = [10, 20, 30]
numbers[1] = 25
print("Modified numbers:", numbers)
```

Here we changed the second item (index 1) from 20 to 25.

Slicing Lists

Sometimes you only want to get a part of the list. You can do this using slicing.

- Use `start:stop` inside brackets to get a sublist.
- The start index is inclusive, stop is exclusive.

```
letters = ['a', 'b', 'c', 'd', 'e']
print("First three letters:", letters[:3])
print("Last two letters:", letters[-2:])
```

Here in the first `print`, we get the first three letters, and in the second `print`, we get the last two letters.

Getting the Length of a List

Many times you need to know how many items are in a list. You can use the `len()` function for this.

- Note: you'll use this a lot more in the future.

```
tasks = ['email', 'meeting', 'code review']
print("Number of tasks:", len(tasks))
```

Here we find out there are 3 tasks in the list.

Out-of-Range Index

If you try to access an index that doesn’t exist, Python will raise an `IndexError`.

Example:

```
fruits = ['apple', 'banana']
print(fruits[5])  # This will raise an IndexError
```

This will cause an error because there is no item at index 5 in the `fruits` list.

Summary

- Lists are a core data structure in Python.
- You can create, access, modify, slice, and measure lists.
- Practice with your own examples to get comfortable with list operations!


WE ARE
ESSENTIAL TO ALBERTA
NAIT

Example

Let’s go run a few examples together