COMP 110

LS 12 - Introduction to Lists

Lists

A list is a **data structure**–something that lets you reason about multiple items.

Examples of lists:

- To-do list
- Assignment Due Dates
- Grocery List

List syntax

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]
```

List syntax

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]

<list name>: list[<item type>] = [<item1>, <item2>, …]
```

List syntax

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]

<list name>: list[<item type>] = [<item1>, <item2>, …]
```

str, int, float, etc.

List syntax

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]
```

```
<list name>: list[<item type>] = [<item1>, <item2>, …]
```

***Lists can be an arbitrary length! (Not a fixed number of items.)*

Initializing an empty list

```
<list name>: list[<item type>] = list()

grocery_list: list[str] = list()
```

Adding an item to a list

```
grocery_list.append(“bananas”)
```

Adding an item to a list

```
grocery_list.append(“bananas”)
```

Method:
Like calling append(grocery_list, “bananas”),
but we are modifying grocery list

Indexing

```
grocery_list: list[str] = [“bananas”, “milk”, “bread”]

grocery_list[0]
```

***Starts at 0, like with strings!*

Modifying by Index

```
grocery_list: list[str] = [“bananas”, “milk”, “bread”]
```

```
grocery_list[1] = “eggs”
```

Length of a List

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]

len(grocery_list)
```

Remove an Item From a List

```
grocery_list: list[str] = [“eggs”, “milk”, “bread”]
```

```
grocery_list.pop(2)
```

Index of item you want to remove