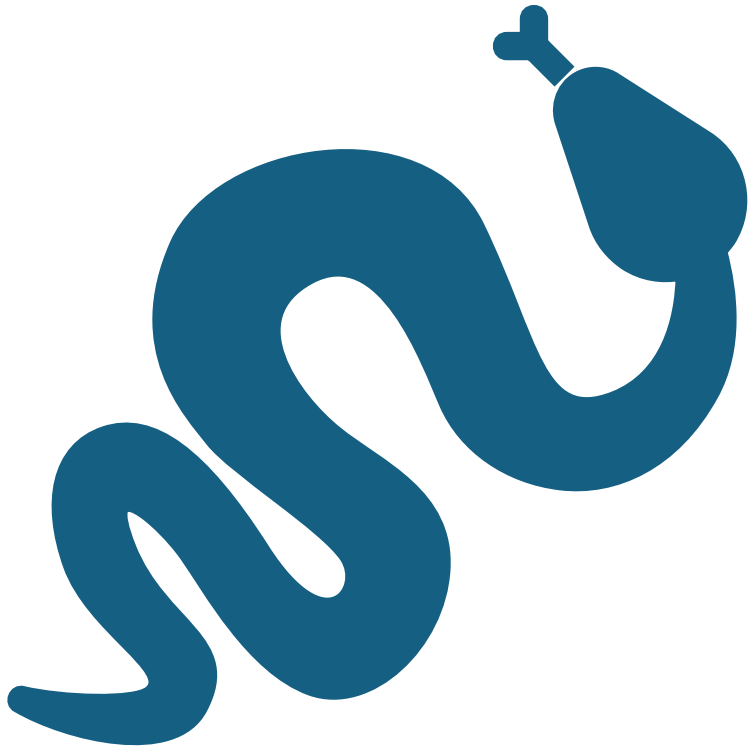


Lesson 4

Python Fundamental

Revisit

For loop reinforcement – enhance dice program

While loop

Comparison of for loop and while loop

Boolean Operators (and, or, not)

Boolean Values (True / False)

Functions: range(start, stop, step)

Control Structure: for loop, while loop

Lab: The dice program enhancement, a while loop to sum positive inputs

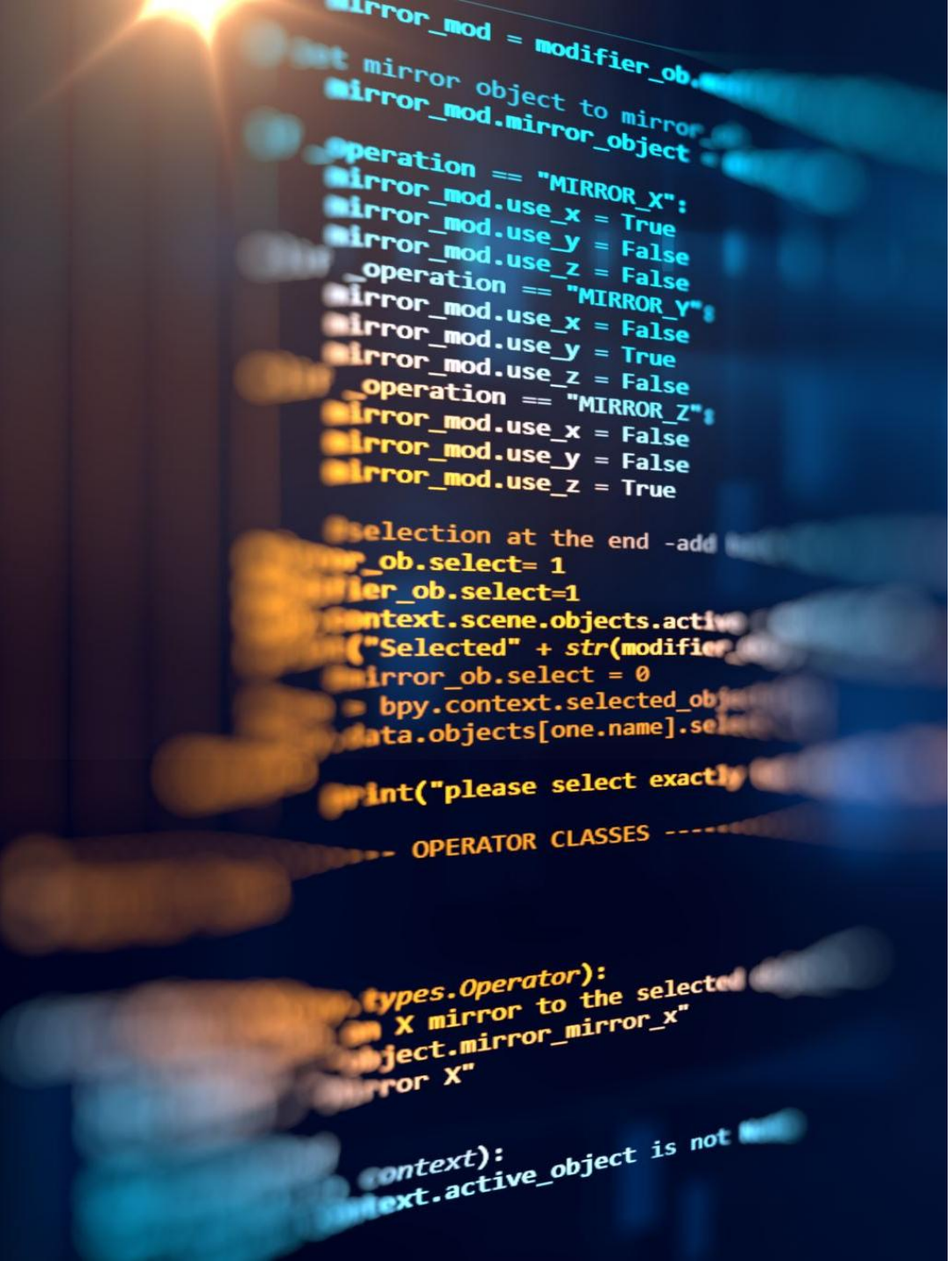
Exercise: a program that simulates basic backing transactions

Technique We've Learned

- Random Library
- Decision Making: if-elif-else
- Logical Operators (==, !=, <, <=, >, >=)
- Boolean Operators (and, or, not)
- Boolean Values (True / False)
- Functions: range(start, stop, step)
- Control Structure: for loop, while loop

Exercise 3 Review

- Refer to code repository



Data Structure: List

- A list is a collection of items stored in a specific order.
- The items can be of any type, like numbers, words, or even other lists.
- What is the output of below *print* statement?

```
mylist = [1, 2, 3, 4, 5]
print(mylist[2])
```

List - Positioning

- Try to print below content:

```
print(mylist[2])
print(mylist[-2])
```

mylist:	1	2	3	4	5
forward position:	0	1	2	3	4
backward position:	-5	-4	-3	-2	-1

Quiz 1

- **True** or **False**:
- A list in Python only store values of the same data type?

(e.g., only numbers or only strings).

Multiple Data Types in a List

- Different data types can be stored in a list

```
1 mylist = [1, 2, 'Apple', 3.141592, [4, 5], 'Banana']
2 print(mylist[3], mylist[4])
```

```
Running: test.py
3.141592 [4, 5]
>>>
```

Question: How can you print the number ‘5’ form **`mylist`**?

List - Slicing

```
1 mylist = [1, 2, 'Apple', 3.141592, [4, 5], 'Banana']
2 print(mylist[3], mylist[4][1])
3 print(mylist[3], mylist[-2][1])
```

```
Running: test.py
3.141592 5
3.141592 5
>>>
```

mylist:	1	2	Apple	3.141592	[4, 5]		Banana
forward position:	0	1	2	3	4		5
					0	1	
backward position:	-6	-5	-4	-3	-2		-1
					0	1	

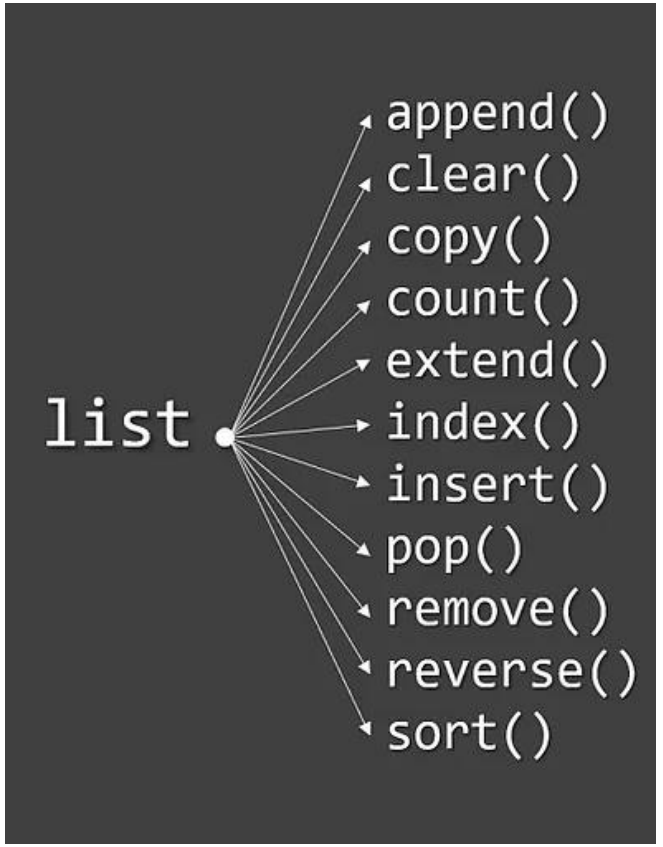
Quiz 2

- What will be the output of the following code?
- A) 1
- B) 3
- C) 4
- D) 5

```
mylist = [1, 2, 3, 4, 5]
print(mylist[-2])
```

List – Manipulations

```
mylist = [1, 2, 3, 4, 5, 4, 3, 2, 1]
mylist.append(6)              # append a new item to mylist
print(mylist)
anotherList = mylist.copy() # copy mylist to anotherList
print(anotherList)
anotherList.clear()           # clear all content of anotherList
print(anotherList)
print(mylist.index(3))        #return the index of first element
mylist.insert(2, 2.5)         # insert an item to position 2
print(mylist)
mylist.pop(2)                 # remove item from position 2
print(mylist)
mylist.remove(1)              # remove the first occurrence of '1' from mylist
print(mylist)
mylist.sort()
print(mylist)                 # sort the list in ascending order
mylist.sort(reverse=True)
print(mylist)                 # sort the list in descending order
```



Quiz 3

- The following list is given:

```
mylist = ["apple", "banana", "cherry", "mango"]
```

- How would you modify this list to:
  - Add "grape" to the end? (.append)
  - Remove "banana"? (.remove)
  - Sot mylist (.sort)
  - Print all elements one by one? (for loop)

List, Tuples, Sets, Dictionaries

Tutorial

Exercise

Create a shopping list:

1. Start with an Empty List
   - Create an empty list called shoppingList.
2. Add Items to the List:
   - Ask the user to enter one shopping item at a time.
   - Use a **`for`** loop to ask for 5 items.
   - After the user enters each item, add (append) it to the shoppingList
   - After all 5 items are entered, sort the shoppingList in alphabetical order.
3. Display the List:
   - Use another for loop to print each item in the sorted shoppingList.
4. Output should look like this:

```
Enter item 1: milk
Enter item 2: bread
Enter item 3: eggs
Enter item 4: apples
Enter item 5: cereal
Your shopping list is: ['milk', 'bread', 'eggs', 'apples', 'cereal']
```