

Lists and dictionaries



LESSON OBJECTIVES

- Explore Python lists and dictionaries
- Functional illustration



First we must understand collections





Collections

Collections in Python are container data types used to store and organize multiple items. They are versatile and widely used in Python programming.

- Lists
- Tuples
- Sets
- Dictionaries



```
_mod = modifier_ob
                     eration == "MIRROR_X":
                   irror mod.use x
                    drror_mod.use
                    irror_mod.use
                     rror_mod.use_z = False
NODE 04
                     operation == "MIRROR
                     rror_mod.use_x = False
NODE 05
                       lection at the end -add
                      Selected" + str(modific
                           ob.select = 0
                      bpy.context.selected_ol
                      ata.objects[one.name].se
                            lease select exact
BLOCK O
                                object is no
```

Characteristics of lists

Lists



- Created using square brackets []
- Data is mutable
- Allows duplicate values



Characteristics of dictionaries

Dictionaries



- Unordered collections of key-value pairs
- Created using curly braces with colons {key:value}
- Data is mutable
- Duplicate values are not allowed

List code example

```
fruits = ["apple", "banana", "cherry", "apple"]
print(fruits)
print(fruits[1]) # Accessing by index
fruits[1] = "blueberry" # Modifying an element
print(fruits)
```



Dictionary code example

```
student = {
    "name":"John Doe",
    "age": 20,
    "major": "Computer Science",
    "gpa": 3.8
}

print(student)
print(student["major"]) #Accessing by key
student["age"] #Updaing the age value
print(student)
```





Functional illustration



