Lists

Save As: lists_example.py

Python Collection Data Types

- **LIST**
- TUPLE
- DICTIONARY

Python Collection Data Type: List

- A **LIST** contains a collection of data in a ordered sequence and **can be changed**.
- Enclosed in **[ ]**.
- **TIP:** Keep values in a list of the **same data type**.
- However, values in lists can be different types, **NOT RECOMMENDED**.

EXAMPLES for a LIST of values

- **Scores** in all courses taken

  scores = [76, 88, 95]

- Shopping List

  grocery = [“bread”, “carrots”, “cheese”]

- Athletic team roster

  roster = [“John”, “Lee”, “Jackie”]

- Guest list for a wedding

  guests = [“Mike”, “Sandy”, “Kelly”]

- Names in a phone book

  names = [“Jane”, “Chris”, “Al”, “Phil”]

List Syntax

empty_list = [ ]

listname = [item1, item2, separated by commas]

guests = [“Michael”, “Sandy”, “Kelly”, “Joe”]

You can create an empty list and add values to it.

```
guests = [ ]
scores = [ ]
animals = [ ]

print(guests)
print(scores)
print(animals)
```

List Syntax

guests	guests[0]	guests[1]	guests[2]	guests[3]
	Michael	Sandy	Kelly	Joe

guests = [“Michael”, “Sandy”, “Kelly”, “Joe”]

print(**guests**[0])

print(**guests**[1])

print(**guests**[2])

print(**guests**[3])

#REMINDER: begins with index or position 0.

#**index** must be **integers**

#**Error** if index number #exceeds list

ACCESSING AN ITEM in a List by their INDEX [position]

animals	animals[0]	animals[1]	animals[2]
	cat	bat	rat

animals = [“cat”, “bat”, “rat”]

```
print(animals)
print(animals[0])
print(animals[1])
```

#**What is the output?**

ACCESSING AN ITEM in a List by their INDEX [position]

EXAMPLE:

scores	scores[0]	scores[1]	scores[2]	scores[3]	scores[4]
	78	85	62	49	98

```
scores = [78, 85, 62, 49, 98]          #scores is the list

print(scores)                          #What is the output?
print(scores[2])
print(scores[1] + scores[2])           #Calculations can be performed
                                       on lists of numeric values
```

You can even access a list backwards [Negative Indexes]

animals	animals[0]	animals[1]	animals[2]
	cat	bat	rat
		animals[-2]	animals[-1]

animals = [“cat”, “bat”, “rat”]

#To access the last item in a list

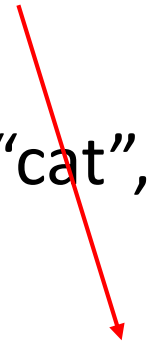
print(**animals**[-1])

#-1 refers to last index in list

print(**animals**[-2])

 # What is output?

CHANGING Values in a List – use assignment = statement



```
animals = [“cat”, “bat”, “rat”, “bird”]

animals[1] = “ant”

print(animals)
```

#you can also use an index of a list
#to **change the value**

#What is the output?

List ERRORS

```
animals = [“cat”, “bat”, “rat”]
```

#**REMINDER:** values in list begins with index 0.

```
print(animals[1.0])
```

#ERROR: index must be INTEGER

```
print(animals[9])
```

#ERROR: index cannot exceed range of list