CoGrammar

Sequences

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: Open Class Questions

Software Engineering Lecture Housekeeping cont.

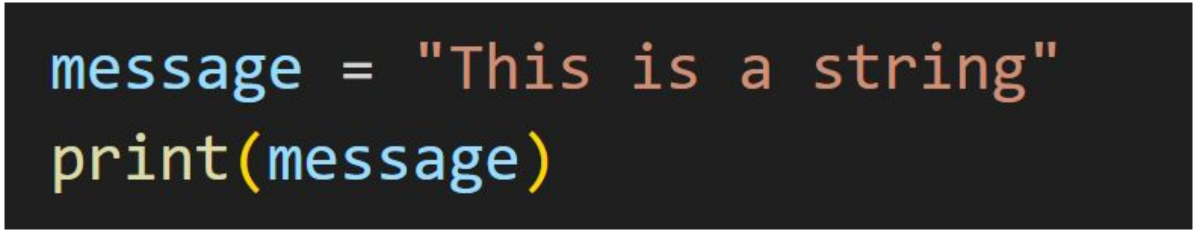
- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Lecture Objectives

1. Describe sequences such as strings, lists and dictionaries.
2. Implement sequence types within your own python projects.
3. Use string, list and dictionary methods to manipulate and perform operations on data.

Strings

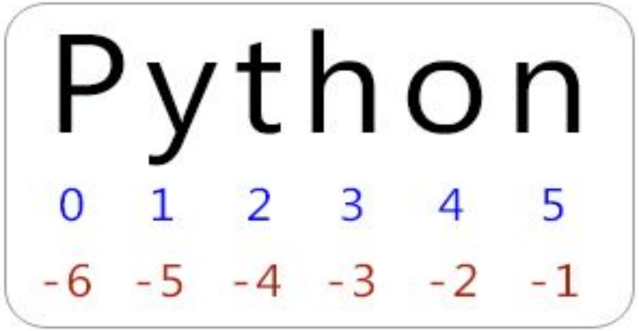
- Strings are a sequence of characters that we usually use to represent text.



```python
message = "This is a string"
print(message)
```

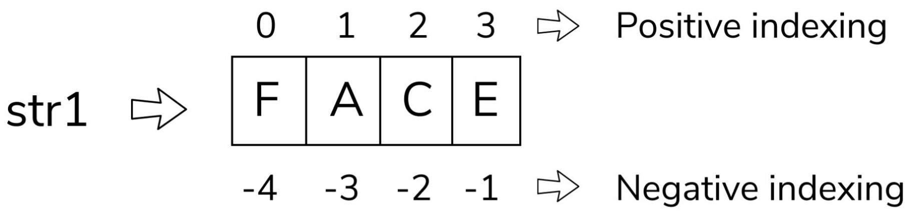
CoGrammar

String Indexing



String Indexing

String Slicing

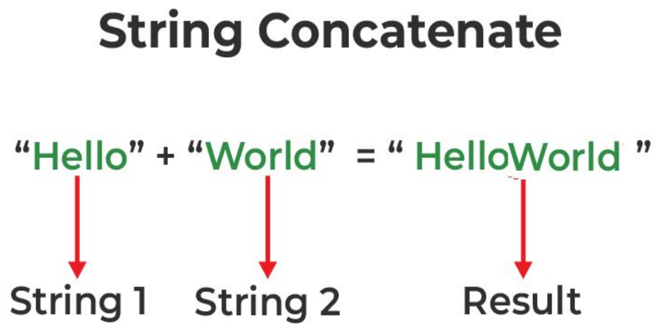


str1[1:3] = AC

str1[-3:-1] = AC

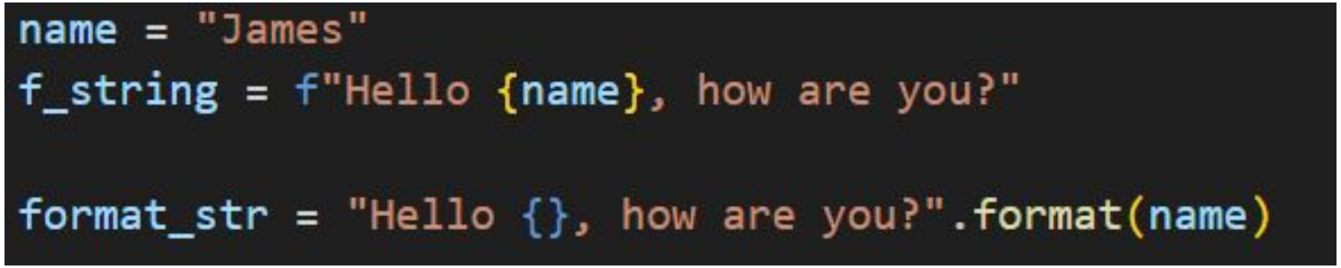
CoGrammar

String Concatenation & Formatting



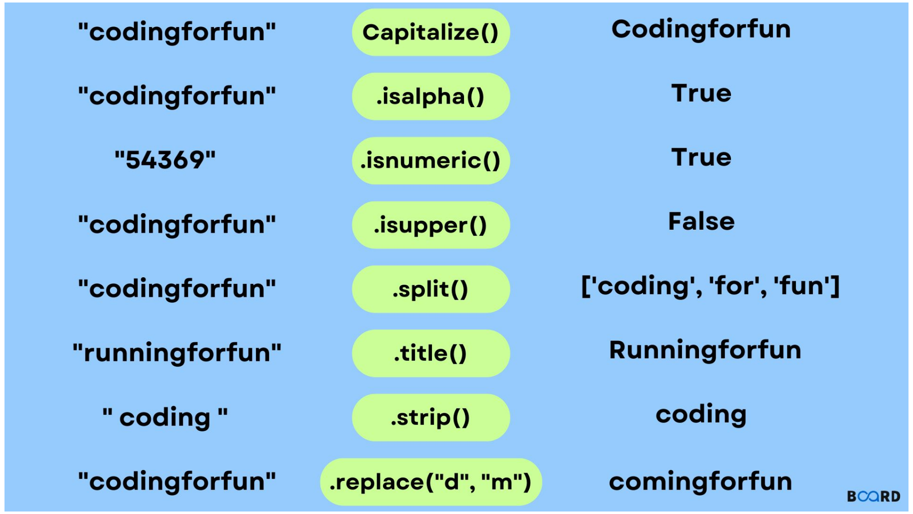
https://www.geeksforgeeks.org/string-concatenation-in-cpp/

f-strings and format() function



```python
name = "James"
f_string = f"Hello {name}, how are you?"

format_str = "Hello {}, how are you?".format(name)
```

Basic String Methods



"codingforfun"	Capitalize()	Codingforfun
"codingforfun"	.isalpha()	True
"54369"	.isnumeric()	True
"codingforfun"	.isupper()	False
"codingforfun"	.split()	['coding', 'for', 'fun']
"runningforfun"	.title()	Runningforfun
" coding "	.strip()	coding
"codingforfun"	.replace("d", "m")	comingforfun

Strings Are Immutable

- When an object is immutable is means the object cannot be changed.
- When we apply methods to a string that appear to make changes, they are actually creating and returning new string objects.
- This means we have to store the changes we make in a variable to be reused.

CoGrammar

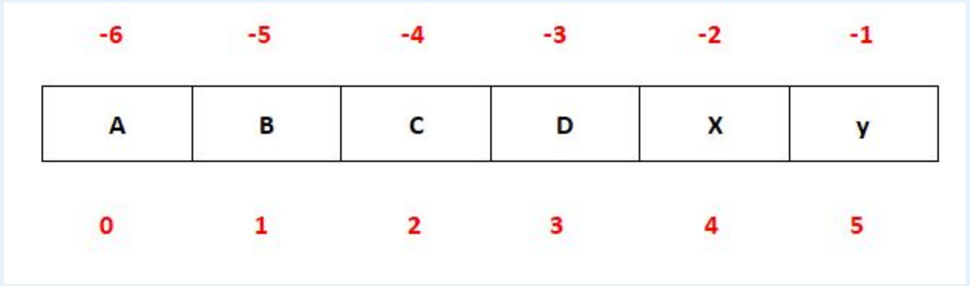
Lists (Arrays)

Python lists are ordered collections of items. They are defined using square brackets '[ ]'.

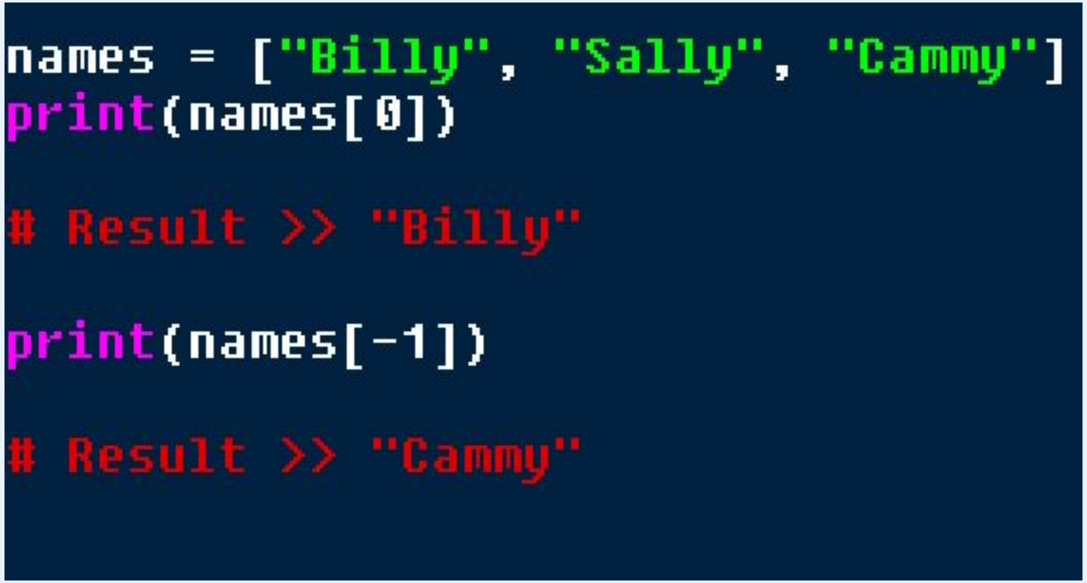
Can contain elements like numbers, strings, or even other lists.

List characteristics:

★ Ordered
★ Mutable
★ Heterogeneous
★ Indexed
★ Supports Slicing
★ Length
★ Common Operations



List Syntax



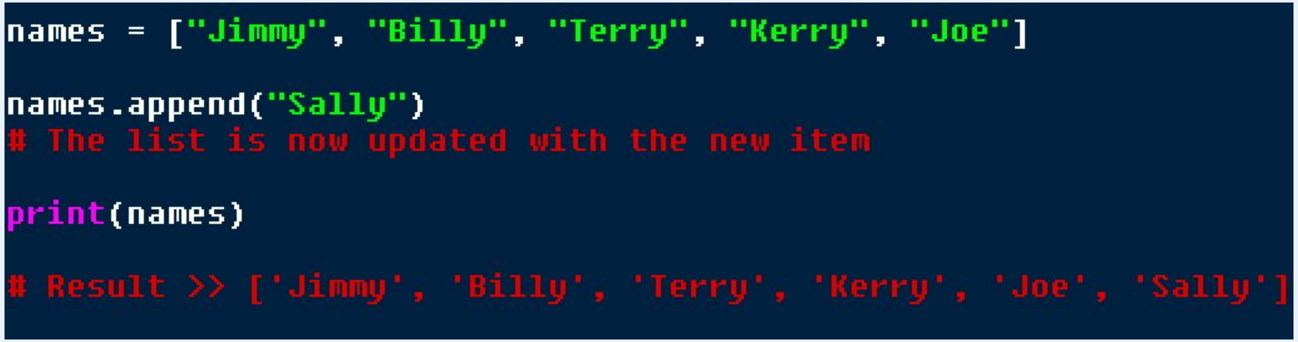
```
names = ["Billy", "Sally", "Cammy"]
print(names[0])

# Result >> "Billy"

print(names[-1])

# Result >> "Cammy"
```

CoGrammar

Appending to Lists

★ You can add new items to a list by using the .append() method, keep in mind that append will only add to the end of a list and nowhere else.

★ Example:



```python
names = ["Jimmy", "Billy", "Terry", "Kerry", "Joe"]

names.append("Sally")
# The list is now updated with the new item

print(names)

# Result >> ['Jimmy', 'Billy', 'Terry', 'Kerry', 'Joe', 'Sally']
```

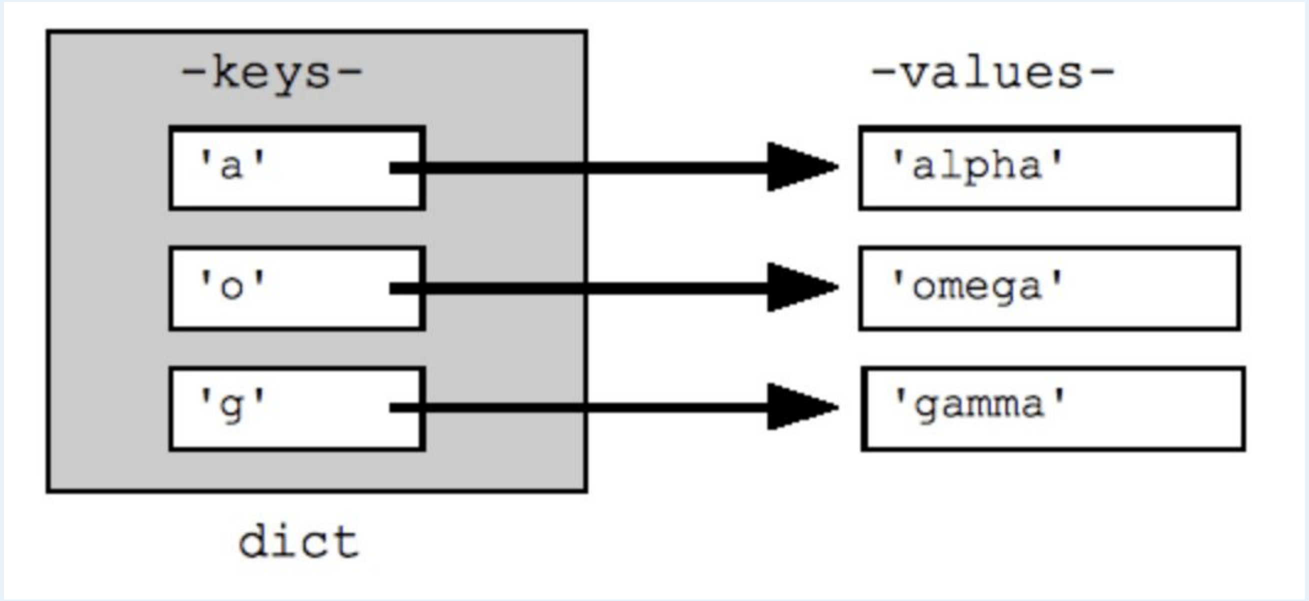
Dictionaries

Python dictionaries are unordered collections of key-value pairs. They are defined using curly braces '{ }'.

Consist of keys and their corresponding values, separated by colons.

Dictionary characteristics:

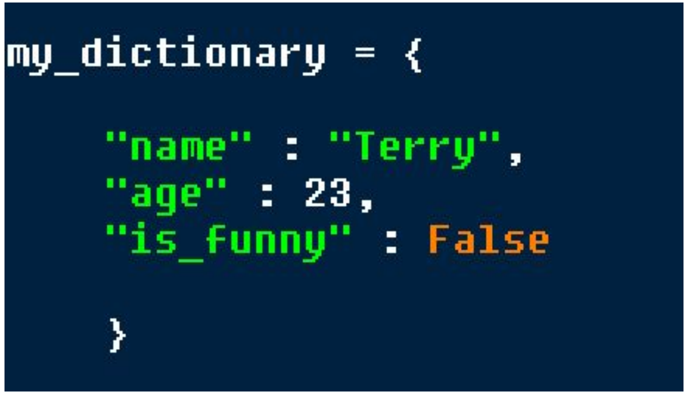
★ Key-Value Mapping
★ Unordered
★ Mutable
★ Heterogeneous Values
★ Access by Key


-keys-
-values-
'a'
'alpha'
'o'
'omega'
'g'
'gamma'
dict

CoGrammar

Dictionaries

★ Dictionaries are enclosed in curly brackets; key value pairs are separated by colon and each pair is separated by a comma.
★ On the left is the key, on the right is the value.



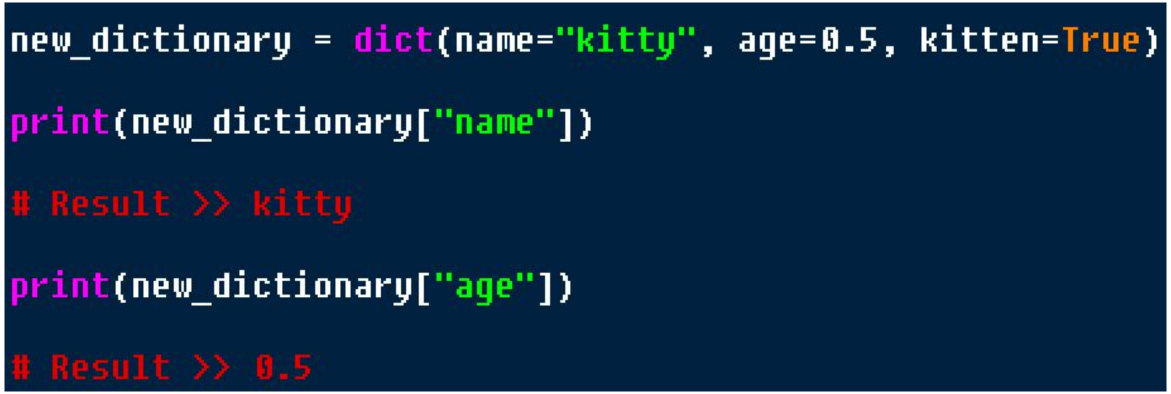
```
my_dictionary = {

    "name" : "Terry",
    "age" : 23,
    "is_funny" : False

    }
```

CoGrammar

Accessing Values

★ To access a value in a dictionary, we simply call the key and Python will return the value paired with said key.

★ Similar to indexing, however we provide a key name instead of an index number.



```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)

print(new_dictionary["name"])

# Result >> kitty

print(new_dictionary["age"])

# Result >> 0.5
```

CoGrammar

Questions around Sequences

CoGrammar

Thank you for joining