

Data Types in Python

Mr. Kasey P. Martin, MIS

Data Types

- Python has five (5) standard data types:
 - Numbers
 - String
 - List
 - Tuple
 - Dictionary

Numbers

- Store numeric values
 - int (signed integers)

```
int_a = -7528
int_b = 132
```

- float (floating point real values)

```
float_a = -75.28
float_b = 13.2e+7
```

- complex (complex numbers)

```
#j is the imaginary unit
complex_a = -75.28j
complex_b = 13.2e-10j
```

Strings

- Contiguous set of characters enclosed by either single or double quotes
- Here are some operations you can use on strings:
- Print the length of a string

```
In [28]: str = "Python is fun!"
print(len(str))
```

14

- Print the first character of a string

```
In [ ]: str = "Python is fun!"
print(str[0])
```

- Print the last character of a string

INDEXING A negative index means you start counting from the end of the string. -1 represents the index of the last character.

```
In [1]: str = "Python is fun!"
print(str[-1])
```

!

- Print the substring of a string
 - string[start:end] - Get all characters from index start to end-1
 - string[:end] - Get all characters from the beginning of the string to end-1
 - string[start:] - Get all characters from index start to the end of the string

```
In [2]: #Let's print the 2nd to 5th characters; string[start:end]
str = "Python is fun!"
print(str[1:5])
```

ytho

```
In [4]: #Let's print the starting character up to the 6th character; string[:end]
str = "Python is fun!"
print(str[:6])
```

Python

```
In [6]: #Let's print the 3rd character up to the last; string[start:]
str = "Python is fun!"
print(str[2:])
```

thon is fun!

- Print the string twice

```
In [7]: str = "Python is fun!"  
print(str * 2)
```

Python is fun!Python is fun!

- Concatenate the string

```
In [9]: str = "Python is fun!"  
print(str + " And so is this class!")
```

Python is fun! And so is this class!

Lists

- Contains items separated by commas and enclosed with square brackets
- Similar to arrays in C or C++. But unlike arrays in these languages, you can store *different* data types in lists!
- Lists are mutable containers which means you can change its contents
- Here are some operations you can use on lists:

- Printing out the list

```
In [17]: list = ["John", 21, "BS Computer Science", 90.5]  
print(list)
```

['John', 21, 'BS Computer Science', 90.5]

- Print the length of a list

```
In [29]: list = ["John", 21, "BS Computer Science", 90.5]  
print(len(list))
```

4

- Print the first and last element of a list

INDEXING Just like in strings, a negative index means you start counting from the end of the list. -1 represents the index of the last element.

```
In [9]: list = ["John", 21, "BS Computer Science", 90.5]
print("First element is: " + list[0])
print(list[-1]) #this works
#but not this: print("Last element is: " + list[-1])
```

```
First element is: John
90.5
```

- Printing out the sublist of a list is similar to strings

```
In [15]: list = ["John", 21, "BS Computer Science", 90.5]
print(list[1:3])
print(list[1:])
print(list[:3])
```

```
[21, 'BS Computer Science']
[21, 'BS Computer Science', 90.5]
['John', 21, 'BS Computer Science']
```

- Append an item at the end of the list

```
In [16]: list = ["John", 21, "BS Computer Science", 90.5]
list.append("Pass")
print(list)
```

```
['John', 21, 'BS Computer Science', 90.5, 'Pass']
```

- Insert an element x at index i
- list.insert[i,x]

```
In [18]: list = ["John", 21, "BS Computer Science", 90.5]
list.insert(3, "Pass")
print(list)
```

```
['John', 21, 'BS Computer Science', 'Pass', 90.5]
```

- Remove an element x from the list if it exists

```
In [21]: list = ["John", 21, "BS Computer Science", "Pass", 90.5]
list.remove("Pass")
print(list)
```

```
['John', 21, 'BS Computer Science', 90.5]
```

- Remove an element from index i
- If no index is specified, it will remove the last item from the list

```
In [27]: list = ["John", 21, "BS Computer Science", 90.5]
         i = 1
         list.pop(i)
         print(list)

['John', 'BS Computer Science', 90.5]
```

Tuples

- Contains items separated by commas and enclosed with parentheses
- Similar to lists where you can store *different* data types
- But unlike lists, tuples are immutable containers (contents cannot be changed)
- Here are some operations you can use on tuples:

- Print the length of a tuple

```
In [38]: tuple = ("Philippines", 300000, "peso", "Asia") #country, land area (km^
         2), currency, continent
         print(len(tuple))

4
```

- Print the nth element of a tuple

```
In [39]: tuple = ("Philippines", 300000, "peso", "Asia") #country, land area (km^
         2), currency, continent
         print(tuple[0])
         print(tuple[-1])

Philippines
Asia
```

- Print the subtuple of a tuple
- tuple(start:end)

```
In [37]: tuple = ("Philippines", 300000, "peso", "Asia") #country, land area (km^
2), currency, continent
print(tuple[0:2])
print(tuple[1:])
print(tuple[:3])

('Philippines', 300000)
(300000, 'peso', 'Asia')
('Philippines', 300000, 'peso')
```

- Concatenate a tuple

```
In [40]: tuple = ("Philippines", 300000, "peso", "Asia") #country, land area (km^
2), currency, continent
tuple_1 = (109000000, "Filipino") #population, major language
print(tuple + tuple_1)

('Philippines', 300000, 'peso', 'Asia', 109000000, 'Filipino')
```

Dictionary

- Consists of key-value pairs where keys are required to be unique
- Enclosed by curly braces {}
- Here are some operations you can use on dictionaries:

- Print the length of a dictionary

```
In [46]: dictionary = {"country": "Philippines", "area": 300000, "currency": "peso",
"continent": "Asia"}
print(len(dictionary))
print(dictionary)

4
{'country': 'Philippines', 'area': 300000, 'currency': 'peso', 'contine
nt': 'Asia'}
```

- Print the keys of a dictionary

```
In [44]: dictionary = {"country": "Philippines", "area": 300000, "currency": "peso",
"continent": "Asia"}
print(dictionary.keys())

dict_keys(['country', 'area', 'currency', 'continent'])
```

- Print the values of a dictionary

```
In [45]: dictionary = {"country": "Philippines", "area": 300000, "currency": "peso",  
                    "continent": "Asia"}  
print(dictionary.values())  
  
dict_values(['Philippines', 300000, 'peso', 'Asia'])
```

- Add a key-value pair to a dictionary

```
In [47]: dictionary = {"country": "Philippines", "area": 300000, "currency": "peso"}  
print(dictionary)  
  
dictionary["continent"] = "Asia"  
print(dictionary)  
  
{'country': 'Philippines', 'area': 300000, 'currency': 'peso'}  
{'country': 'Philippines', 'area': 300000, 'currency': 'peso', 'continent': 'Asia'}
```

- Delete a key-value pair from a dictionary

```
In [48]: dictionary = {"country": "Philippines", "area": 300000, "currency": "peso",  
                    "continent": "Asia"}  
print(dictionary)  
  
del dictionary["area"]  
print(dictionary)  
  
{'country': 'Philippines', 'area': 300000, 'currency': 'peso', 'continent': 'Asia'}  
{'country': 'Philippines', 'currency': 'peso', 'continent': 'Asia'}
```