Asssignment : 2

1. Integer (int)

The int data type represents whole numbers.

```
num = 10

# Using help() to see methods

help(int)

# Common methods

print(num.bit_length())  # Returns the number of bits required to represent the number in binary.

print(num.to_bytes(2, byteorder='big'))  # Converts the integer to bytes.
```

2. Float (float)

```
# Example of float

num = 3.14

# Using help() to see methods

help(float)

# Common methods

print(num.is_integer())  # Checks if the float is a whole number.

print(num.as_integer_ratio())  # Returns a pair of integers whose ratio is equal to the float.
```

3. Complex (complex)

```
# Example of complex

num = 3 + 4j

# Using help() to see methods

help(complex)

# Common methods

print(num.real)  # Returns the real part of the complex number.

print(num.imag)  # Returns the imaginary part of the complex number.

print(num.conjugate())  # Returns the complex conjugate.
```

4. List (list)

The list data type is an ordered, mutable collection of items.

```
# Example of list
my_list = [1, 2, 3, 4, 5]
# Using help() to see methods
help(list)
# Common methods
my_list.append(6)  # Adds an element to the end of the list.
my_list.extend([7, 8])  # Extends the list with another list.
my_list.remove(3)  # Removes the first occurrence of the value.
print(my_list.index(4))  # Returns the index of the first occurrence of the value.
print(my_list.count(2))  # Returns the number of occurrences of the value.
```

5. Tuple (tuple)

The tuple data type is an ordered, immutable collection of items.

```
# Example of tuple
my_tuple = (1, 2, 3, 4, 5)
# Using help() to see methods
help(tuple)
# Common methods
print(my_tuple.index(3))  # Returns the index of the first occurrence of the value.
print(my_tuple.count(2))  # Returns the number of occurrences of the value.
```

6. String (str)

The str data type represents a sequence of characters.

```
# Example of string
my_string = "Hello, World!"
# Using help() to see methods
help(str)
```

```
# Common methods
print(my_string.upper())  # Converts the string to uppercase.
print(my_string.lower())  # Converts the string to lowercase.
print(my_string.replace("World", "Python"))  # Replaces a substring with another.
print(my_string.split(","))  # Splits the string into a list based on a delimiter.
print(my_string.find("World"))  # Returns the index of the first occurrence of the substring.
```

7. Set (set)

The set data type is an unordered collection of unique elements.

```
# Example of set
my_set = {1, 2, 3, 4, 5}
# Using help() to see methods
help(set)
# Common methods
my_set.add(6)  # Adds an element to the set.
my_set.remove(3)  # Removes an element from the set.
print(my_set.union({6, 7, 8}))  # Returns the union of two sets.
print(my_set.intersection({4, 5, 6}))  # Returns the intersection of two sets.
```

8. Dictionary (dict)

The dict data type is an unordered collection of key-value pairs.

```
# Example of dictionary
my_dict = {"name": "Alice", "age": 25, "city": "New York"}
# Using help() to see methods
help(dict)
# Common methods
print(my_dict.keys())  # Returns a list of keys.
print(my_dict.values())  # Returns a list of values.
print(my_dict.items())  # Returns a list of key-value pairs.
my_dict.update({"age": 26})  # Updates the dictionary with new key-value pairs.
```

```
print(my_dict.get("name"))  # Returns the value for the specified key.
```

9. Boolean (bool)

The bool data type represents True or False.

```
# Example of boolean

flag = True

# Using help() to see methods

help(bool)

# Common methods

print(flag.__bool__())  # Returns the boolean value.

print(flag.__and__(False))  # Performs a logical AND operation.
```