

# Coding in Python

## Python Basics - Part 2

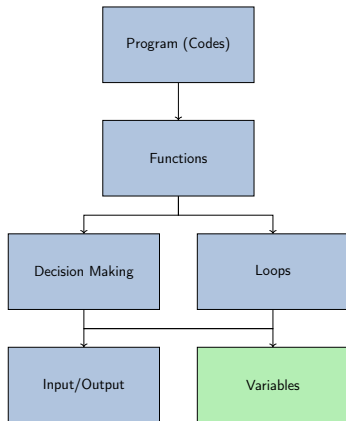
IDPO 2910 Group 5

4 May 2024



THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

# Contents



Imagine you have a bunch of variables you want to store. For example, if you have a bunch of people's names.

```
name0 = "Chris Wong"  
name1 = "Desmond Tsoi"  
name2 = "Phoebe Mok"  
name3 = "Nancy Ip"
```

That is annoying to store and access.

Imagine you have a bunch of variables you want to store. For example, if you have a bunch of people's names.

```
name0 = "Chris Wong"  
name1 = "Desmond Tsoi"  
name2 = "Phoebe Mok"  
name3 = "Nancy Ip"
```

That is annoying to store and access.

What if instead, we store it in the same thing, as a...

Imagine you have a bunch of variables you want to store. For example, if you have a bunch of people's names.

```
name0 = "Chris Wong"  
name1 = "Desmond Tsoi"  
name2 = "Phoebe Mok"  
name3 = "Nancy Ip"
```

That is annoying to store and access.

What if instead, we store it in the same thing, as a... list?

# Lists

```
names = ["Chris Wong", "Desmond Tsoi",  
         "Phoebe Mok", "Nancy Ip"]  
print(names[0]) # Chris Wong
```

Lists are declared by surrounding the items with [ ], and separating each item with a comma.

We can access the name from a list by getting the corresponding item. The first item in the list is the 0<sup>th</sup> item, second is 1<sup>st</sup> item, etc... We call this zero-indexing.

Note: Some programming languages use one-indexing instead.

If you approach another programming language, be careful.

# Lists

```
names = ["Chris Wong", "Desmond Tsoi",  
         "Phoebe Mok", "Nancy Ip"]  
print(names[0], names[1], names[2], names[3])  
# Output: Chris Wong Desmond Tsoi Phoebe Mok Nancy Ip
```

Another example:

```
# Indices: 0  1  2  3  4  5
numbers = [0, 1, 1, 2, 3, 5]
print(numbers[0], numbers[1], numbers[2],
      numbers[3], numbers[4], numbers[5])
# Output: 0 1 1 2 3 5

print(numbers)
# Output: [0, 1, 1, 2, 3, 5]
```

## Printing the whole list

To print the whole list, simply put it in the `print()` function.



To get the length of a list, we can use the `len()` function.

To get the length of a list, we can use the `len()` function.

```
numbers = [0, 1, 1, 2, 3, 5]  
print(len(numbers)) # 6
```

To edit an element of a list, assign the new value to the correct index.

To edit an element of a list, assign the new value to the correct index.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers) # [0, 1, 1, 2, 3, 5]
numbers[1] = 100 # Edit the second element (index 1)
print(numbers)
```

To edit an element of a list, assign the new value to the correct index.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers) # [0, 1, 1, 2, 3, 5]
numbers[1] = 100 # Edit the second element (index 1)
print(numbers)  # [0, 100, 1, 2, 3, 5]
```

To add an element to the end to a list, we use the `append(value)` list function.

To add an element to the end to a list, we use the `append(value)` list function.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.append(100) # Add 100 to the end of the list
print(numbers, "length:", len(numbers))
```

To add an element to the end to a list, we use the `append(value)` list function.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.append(100) # Add 100 to the end of the list
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5, 100] length: 7
```



# Lists

To insert an element to a particular position in a list, we use the `insert()` list function.

The `insert(i, value)` inserts the `value` at index `i`, and push everything after to the right.

# Lists

To insert an element to a particular position in a list, we use the `insert()` list function.

The `insert(i, value)` inserts the value at index `i`, and push everything after to the right.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.insert(2, 100) # Add 100 to index 2 of the list
print(numbers, "length:", len(numbers))
```

# Lists

To insert an element to a particular position in a list, we use the `insert()` list function.

The `insert(i, value)` inserts the value at index `i`, and push everything after to the right.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.insert(2, 100) # Add 100 to index 2 of the list
print(numbers, "length:", len(numbers))
# Output: [0, 1, 100, 1, 2, 3, 5] length: 7
numbers.insert(7, 200) # Same as numbers.append(200)
print(numbers, "length:", len(numbers))
```

# Lists

To insert an element to a particular position in a list, we use the `insert()` list function.

The `insert(i, value)` inserts the value at index `i`, and push everything after to the right.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.insert(2, 100) # Add 100 to index 2 of the list
print(numbers, "length:", len(numbers))
# Output: [0, 1, 100, 1, 2, 3, 5] length: 7
numbers.insert(7, 200) # Same as numbers.append(200)
print(numbers, "length:", len(numbers))
# Output: [0, 1, 100, 1, 2, 3, 5, 200] length: 8
```

To remove an element from a list, we use the `remove()` list function. The `remove(value)` function removes the **first** occurrence of `value`.

To remove an element from a list, we use the `remove()` list function. The `remove(value)` function removes the **first** occurrence of value.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.remove(1) # Remove the first occurrence of number 1
print(numbers, "length:", len(numbers))
```

To remove an element from a list, we use the `remove()` list function. The `remove(value)` function removes the **first** occurrence of value.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.remove(1) # Remove the first occurrence of number 1
print(numbers, "length:", len(numbers))
# Output: [0, 1, 2, 3, 5] length: 5
```

The `reverse()` list function reverses a list's contents.



# Lists

The `reverse()` list function reverses a list's contents.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.reverse() # Reverse the list
print(numbers, "length:", len(numbers))
```

The `reverse()` list function reverses a list's contents.

```
numbers = [0, 1, 1, 2, 3, 5]
print(numbers, "length:", len(numbers))
# Output: [0, 1, 1, 2, 3, 5] length: 6
numbers.reverse() # Reverse the list
print(numbers, "length:", len(numbers))
# Output: [5, 3, 2, 1, 1, 0] length: 6
print(numbers[0])
# Output: 5
```

The `count(item)` list function counts the number of occurrence of `item` in a list.

The `count(item)` list function counts the number of occurrence of `item` in a list.

```
numbers = [0, 1, 1, 2, 3, 5]
```

```
print(numbers.count(1))
```

```
# Output: 2
```

```
print(numbers.count(100))
```

The `count(item)` list function counts the number of occurrence of `item` in a list.

```
numbers = [0, 1, 1, 2, 3, 5]
```

```
print(numbers.count(1))
```

```
# Output: 2
```

```
print(numbers.count(100))
```

```
# Output: 0
```

The `index(item)` list function finds the index of the first occurrence of `item` in a list.

The `index(item)` list function finds the index of the first occurrence of `item` in a list.

```
numbers = [0, 1, 1, 2, 3, 5]
```

```
print(numbers.index(1))
```

```
# Output: 1
```

```
print(numbers.index(5))
```

```
# Output: 5
```

```
print(numbers.index(100))
```

The `index(item)` list function finds the index of the first occurrence of `item` in a list.

```
numbers = [0, 1, 1, 2, 3, 5]
```

```
print(numbers.index(1))
```

```
# Output: 1
```

```
print(numbers.index(5))
```

```
# Output: 5
```

```
print(numbers.index(100))
```

```
# Error, 100 is not in the list
```



The `sort()` list function sorts a list's contents.

# Lists

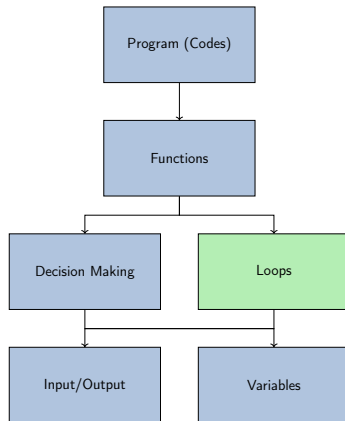
The `sort()` list function sorts a list's contents.

```
numbers = [6, 5, 1, 2, 3]
print(numbers, "length:", len(numbers))
# Output: [6, 5, 1, 2, 3] length: 5
print(numbers[0])
# Output: 6
numbers.sort() # Reverse the list
print(numbers, "length:", len(numbers))
```

# Lists

The `sort()` list function sorts a list's contents.

```
numbers = [6, 5, 1, 2, 3]
print(numbers, "length:", len(numbers))
# Output: [6, 5, 1, 2, 3] length: 5
print(numbers[0])
# Output: 6
numbers.sort() # Reverse the list
print(numbers, "length:", len(numbers))
# Output: [1, 2, 3, 5, 6] length: 5
print(numbers[0])
# Output: 1
```



The end  
Written in  $\text{\LaTeX}$   
Last updated: 15 Mar 2024