Python Environment and Program Control

- **Function**: A block of code that only runs when called.

```
def greet():
    print("Hello!")
greet()  # Output: Hello!
```

INPUT: Used to take user input.

```
name = input("Enter your name: ")
print("Hello, " + name)
```

2. Conditionals and Loops

- **IF Statement**: Executes a block of code if a condition is true.

```
if age >= 18:
    print("You are an adult.")
```

ELSE and ELIF: Else runs when the if condition is false, and elif allows more conditions.

```
if age < 18:
    print("You are a minor.")
elif age == 18:
    print("You are exactly 18.")
else:
    print("You are an adult.")
```

WHILE Loop: Repeats a block of code while a condition is true.

```
count = 1
```

```
while count <= 5:
    print(count)
    count += 1
```

FOR Loop: Iterates over a sequence (like a list or range).

```
for i in range(5):
    print(i)
```

RANGE: Generates a sequence of numbers.

```
for i in range(2, 10, 2):  # From 2 to 9, stepping by 2
    print(i)
```

MAIN: The main function is where code execution starts.

```
def main():
    print("This is the main function.")

if __name__ == "__main__":
    main()
```

Open File: Used to read from or write to files.

```
file = open("sample.txt", "r")  # Open file for reading
content = file.read()
print(content)
file.close()
```

Collections

- **TUPLE**: Immutable list.

```
my_tuple = (1, 2, 3)

print(my_tuple)
```

LIST: Mutable collection.

```
my_list = [1, 2, 3]

my_list.append(4)

print(my_list)
```

SET: Collection of unique elements.

```
my_set = {1, 2, 3, 3}

print(my_set)  # Output: {1, 2, 3}
```

SORT: Sort a list in place.

```
numbers = [4, 1, 3, 2]

numbers.sort()

print(numbers)
```

Dictionary and List Operations

- **Dictionary**: Stores data in key-value pairs

```
my_dict = {"a": 1, "b": 2}

print(my_dict["a"])  # Output: 1
```

APPEND: Add an item to the end of a list

```
my_list.append(5)
```

```
print(my_list)  # Output: [1, 2, 3, 4, 5]
```

INSERT: Insert at a specific index

```
my_list.insert(2, 6)
print(my_list)  # Output: [1, 2, 6, 3, 4, 5]
```

REMOVE: Remove the first occurrence of an element.

```
my_list.remove(3)
print(my_list)  # Output: [1, 2, 6, 4, 5]
```

INDEX: Find the index of an item.

```
index = my_list.index(6)
print(index)  # Output: 2
```

POP: Removes and returns the last item

```
last_item = my_list.pop()
print(last_item)  # Output: 5
```

Function with return

```
def add(a, b):
    return a + b

result = add(2, 3)  # result is 5
```