

[Day-4]

Functions in Python - Functions are block of code that performs a specific task. Functions help in organizing code, reusing code, and improving readability.

```
def func_name(parameters):
```

```
# Function body
```

```
return expression
```

[multiple parameters
default parameters]

Positional arguments

```
def print_num(*args):
```

```
for num in args:
```

```
    print(number)
```

(key-value pairs)
keywords Arguments

```
def print_details(**kwargs):
```

```
for key, value in kwargs.items():
```

```
    print(f"{key} : {value}")
```

Function examples:

(1) Temperature conversion

```
def convert_temp(temp, unit):
```

```
    if unit == 'C':
```

```
        return temp * 9/5 + 32
```

```
    elif unit == 'F':
```

```
        return (temp - 32) * 5/9
```

```
    else:
```

```
        return None
```

```
print(convert_temp(25, 'C'))
```

```
print(convert_temp(78, 'F'))
```

(2) Password strength checker

```
def is_strong(password):
```

```
    if len(password) < 8:
```

```
        return False
```

```
    if not any(char.isdigit() for char in password):
```

```
        return False
```

```
    if not any(char.islower() for char in password):
```

```
        return False
```

```
    return True
```

(3) Total cost of items in cart

def calculate_total_cost(cart):

total_cost = 0

for item in cart:

total_cost += item['price'] * item['quantity']

return total_cost

Example cart

cart = [{'name': 'Apple', 'price': 0.5, 'quantity': 5},
 {'name': 'Pen', 'price': 2, 'quantity': 10}]

total_cost = calculate_total_cost(cart)

print(total_cost)

(4) Check if palindrome

def is_palindrome(s):

s = s.lower().replace(" ", "")

return s == s[::-1]

Lambda Function - These are small anonymous functions defined using the lambda keyword. They can have any number of arguments but only one expression. They are commonly used for short operations or as arguments to higher-order functions.

lambda arguments : expression

Ex: lambda a, b : a+b