Functions2

```python
def greet(name,city):
    print("hey", name, "How are you?")
    print("How is weather in your", city)
```

```python
greet("Yogeswar", "Pune")
```

```
hey Yogeswar How are you?
How is weather in your Pune
```

```python
greet("Pune", "Yogeshwar")
```

```
hey Pune How are you?
How is weather in your Yogeshwar
```

```python
# String formatting
```

```python
n = "Rahul"
last = 23423
# {} is a place holder in our string
print("My name is {} and my last name is {}".format(n, last))
```

```
My name is Rahul and my last name is 23423
```

```python
def greet(name,city):
    print("hey {}, How are you?".format(name))
    print("How is weather in your {}".format(city))
```

```python
greet("Monika", "Pune")
```

```
hey Monika, How are you?
How is weather in your Pune
```

```python
greet("Pune", "Monika")
```

```
hey Pune, How are you?
How is weather in your Monika
```

Keyworded arguments

```python
def f(a, b, c):
    print("Value of a is {}".format(a))
    print("Value of b is {}".format(b))
    print("Value of c is {}".format(c))
    print(2*a + b + c)

f(2, 3, 4)
```

```
Value of a is 2
Value of b is 3
Value of c is 4
11
```

```
f(3, 2, 4)
```

```
Value of a is 3
Value of b is 2
Value of c is 4
12
```

```
f(a = 2, c = 4, b = 3)
```

```
Value of a is 2
Value of b is 3
Value of c is 4
11
```

```
def f(a, b, c):
    print((a + b + c)/3)

f(3, 5, 7)
```

```
5.0
```

```
def f(a, b, c):
    print('First number: {}'.format(a))
    print('Second number: {}'.format(b))
    print('Third number: {}'.format(c))
    print('The result is: {}'.format(2*a + b - 3*c))
f(1, 2)
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call 
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_18093/19846
95380.py in <module>
      4     print('Third number: {}'.format(c))
      5     print('The result is: {}'.format(2*a + b - 3*c))
----> 6 f(1, 2)
```

```
TypeError: f() missing 1 required positional argument: 'c'
```

```
def greet(name,city):
    print("hey {}, How are you?".format(name))
    print("How is weather in your {}".format(city))

greet(city = "Pune", name = "Monika")
```

```
hey Monika, How are you?
How is weather in your Pune
```

```
# It will give an error
# greet(city = "Pune")
```

Mixing Keyworded ags and positional arguments

```
def f(a, b, c):
    print("Value of a is {}".format(a))
    print("Value of b is {}".format(b))
    print("Value of c is {}".format(c))
    print(2*a + b + c)

f(2, 3, c = 4)
```

```
Value of a is 2
Value of b is 3
Value of c is 4
11
```

```
f(2, c = 3, b = 5)
```

```
Value of a is 2
Value of b is 5
Value of c is 3
12
```

```
f(2, 3, a = 4)
```

```
-----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_18093/33868
88599.py in <module>
----> 1 f(2, 3, a = 4)

TypeError: f() got multiple values for argument 'a'
```

Positional argument can't follow keyword argument

```
f(c = 2, 4, 3)
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_18093/1103
725724.py", line 1
    f(c = 2, 4, 3)
                 ^
SyntaxError: positional argument follows keyword argument
```

```
f(2,c=4,5)
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_18093/8070
09256.py", line 1
    f(2,c=4,5)
             ^
SyntaxError: positional argument follows keyword argument
```

```
f(2, 3, 4)
```

```
Value of a is 2
Value of b is 3
Value of c is 4
11
```

```
f(a = 4, b = 4, c = 5)
```

```
Value of a is 4
Value of b is 4
Value of c is 5
17
```

```
f(2, 3, c = 5)
```

```
Value of a is 2
Value of b is 3
Value of c is 5
12
```

```
f(2, c = 4, b = 6)
```

```
Value of a is 2
Value of b is 6
Value of c is 4
14
```

```python
def greet(name, location):
    # string formatting
    print("Hi {} how are you doing?".format(name))
    print("Isn't it a nice weather today in {}?".format(location))
greet("Nowhere", "Krishan")
```

```
Hi Nowhere how are you doing?
Isn't it a nice weather today in Krishan?
```

```python
def power(number,exponent):
    return number**exponent

print(power(3, 2))
```

```
9
```

```python
def power(number,exponent):
    return number**exponent

power(exponent=2,number=3)
```

```
9
```

Default Arguments

```python
print(2, 3, 4, 5, end=" -> ")
print(3, 5, 6)
```

```
2 3 4 5 -> 3 5 6
```

```python
def power(number,e = 1):
    return number**e

power(3)
```

```
3
```

```python
def power(x,e=1):
    return x**e

power(3, 3)
```

```
27
```

```python
power(2, 4)
```

```
16
```

Date

```python
def print_date(d, m, y, style=0):
    if style == 0: # American
        print(m, '/', d, '/', y)
    elif style == 1: # European
        print(d, '/', m, '/', y)
    else:
        print('Invalid Style')
print_date(30, 5, 2022)
print_date(4, 12, 2022)
```

```
5 / 30 / 2022
12 / 4 / 2022
```

```python
def print_date(d = 1, m = 1, y = 2005, style=0):
    if style == 0: # American
        print(m, '/', d, '/', y)
    elif style == 1: # European
        print(d, '/', m, '/', y)
    else:
        print('Invalid Style')
```

```python
print_date()
```

```
1 / 1 / 2005
```

```python
print_date(d = 2, m = 4)
```

```
4 / 2 / 2005
```

```python
print_date(d = 2, y = 2006)
```

```
1 / 2 / 2006
```

```python
print_date(d = 2, y = 2006, style = 3)
```

```
Invalid Style
```

Scope of a variable

```python
## This king is in global scope
king = 20
```

```python
def my_family():
    # this king is in local scope of function
    king = 10
    print("King in local scope", king)

print(my_family())
print("King outside family", king)
```

```
King in local scope 10
King outside family 20
```

```python
## This king is in global scope
king = 20

def my_family():
    # Now using global keyword we are affecting global variable only
    global king
    king = 10
    print("King in local scope", king)

print("King outside family", king)
my_family()
print(king)
```

```
King outside family 20
King in local scope 10
10
```

```python
king = 20
king = 10
print(king)
```

```
10
```

```python
# doubts

x = 20
def f():
    y = 30
    print(y)
```