Functions 2

```
# String formatting

name = input()

 Rahul

print("rahul")

rahul


def greet(name, place):
    print("Hey my name is", name, "I live in", place)

greet("Rahul", "Gurgaon")

Hey my name is Rahul I live in Gurgaon


name = input()
place = input()

 Rahul
 Guragon

print("Hey my name is", name, ".", "I live in", place)

Hey my name is Rahul . I live in Guragon




# String formatting

# This formating is available in python 3

print("Hey my name is {} and I live in {}.".format(name, place))

Hey my name is Rahul and I live in Guragon.


# print("Hey my name is {} and I live in {}.".format(name))

# Upper code will not work
```

```
print("Hey my name is {} and I live in {}.".format(place, name))
```

```
Hey my name is Guragon and I live in Rahul.
```

```
# For format method we use only {} these brackets
# it is available in python 3.6 above version only
print(f"Hey my name is {name} and I live in {place}.")
```

```
Hey my name is Rahul and I live in Guragon.
```

Keyworded arguments

```
def family_intro(father, mother, sibling):
    print(f"My father's name is {father}")
    print(f"My mother's name is {mother}")
    print(f"My sibling's name is {sibling}")
```

```
# Following are positional arguments
family_intro("Mummy", "father", "sibling")
```

```
My father's name is Mummy
My mother's name is father
My sibling's name is sibling
```

```
family_intro(sibling="Chotu", father="Papa", mother="Mummy")
```

```
My father's name is Papa
My mother's name is Mummy
My sibling's name is Chotu
```

```
# family_intro(sibling="Chotu", father="Papa", mother="Mummy",
father="Amma")

# Dont assign multiple values to a parameter
```

```
# Quiz

def greet(name, location):
    # string formatting
    # print("Hi", name, "how are you doing?")
    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?
```

```
# def greet(name):
#     print("Hey", name, "how are you doing?")
# greet()
```

```
## Difference between parameter and arguments

# While defining the function the value we pass in function is called
parameter

def xyz(parameter):
    print(parameter)

# The value which we pass while calling the funciton is called
argument

xyz("Rahul")
```

```
Rahul
```

Mixing Keyworded args and positional arguments

```
def family_intro(father, mother, sibling):
    print(f"My father's name is {father}")
    print(f"My mother's name is {mother}")
    print(f"My sibling's name is {sibling}")
```

```
# following are positional

family_intro("papa", "mummy","chotu")
```

```
My father's name is papa
My mother's name is mummy
My sibling's name is chotu
```

```
# Following are keyworded

family_intro(sibling="Chotu", father="Papa", mother="Mummy")
```

```
My father's name is Papa
My mother's name is Mummy
My sibling's name is Chotu
```

```
family_intro("papa", sibling="chotu", mother="mummy")
```

```
My father's name is papa
My mother's name is mummy
My sibling's name is chotu
```

```
# using positional argument after using keyword
# positional argument should not follow keyword argument

family_intro(sibling="chotu", mother="mummy", "papa")
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_6639/42582
00939.py", line 1
    family_intro(sibling="chotu", mother="mummy", "papa")
                                                        ^

SyntaxError: positional argument follows keyword argument
```

```
family_intro("papa", father="papa", mother="mummy", sibling="chotu")
```

```
---------------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_6639/209762
```

```
0504.py in <module>
----> 1 family_intro("papa", father="papa", mother="mummy",
sibling="chotu")

TypeError: family_intro() got multiple values for argument 'father'
```

```
family_intro("papa", "mummy", sibling="chotu")
```

```
My father's name is papa
My mother's name is mummy
My sibling's name is chotu
```

```
family_intro(father="papa", "mummy", sibling="chotu")
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_6639/30616
7423.py", line 1
    family_intro(father="papa", "mummy", sibling="chotu")
                                                        ^
SyntaxError: positional argument follows keyword argument
```

Rule: Positional argument can't follow keyword argument

```
# Yes I know that already
```

Default Parameters

```
# print?

# nationality in below example is default parameter

def passport(name, nationality="Indian"):
    print(f"{name} is {nationality}")

passport("Aniket")
```

```
Aniket is Indian
```

```
# Changing by default values
```

```
passport("Rahul", nationality="German")
```

```
Rahul is German
```

```
passport("Karan ji", "American")
```

```
Karan ji is American
```

```
# quiz
```

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

```
power(3, 3)
```

```
27
```

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

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

```
9
```

```
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_6639/198469
5380.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 power(number,exponent):
    return number**exponent

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

```
9
```

Date

```
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
```

```
# Quiz

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)
```

```
5 / 30 / 2022
```

```
def print_date(d=1, m=1, y=2022, s=0):
    if s == 0:  # American
        print(m, '/', d, '/', y)
    elif s == 1:  # European
        print(d, '/', m, '/', y)
    else:
        print('Invalid Style')

print_date(d=30, y=2022)
```

```
1 / 30 / 2022
```

```
def print_date(d=1, m=1, y=2022, s=0):
    if s == 0:  # American
        print(m, '/', d, '/', y)
    elif s == 1:  # European
        print(d, '/', m, '/', y)
    else:
        print('Invalid Style')

print_date(d=30, y=2022, s = 1)
```

```
30 / 1 / 2022
```

Scope of a variable

```
# Inside out king

def kingdom():
    # Local scope of Kingdom
    king = "Sathy ji"
    print(king)

kingdom()
# print(king)
```

```
Sathy ji
```

```python
# outside inside king

# Global scope
king = "Rahul"

def kingdom():
    print(king)

kingdom()
print(king)
```

```
Rahul
Rahul
```

```python
# Both kings

# Global scope
king = "Rahul"

def kingdom():
    # Local king
    king = "Dravid"
    print(king)

kingdom()
print(king)
```

```
Dravid
Rahul
```

```python
# Changing global value

# global scope

king = "Rahul"

def coup():
    # global keyword will change value globally
    global king
    king = "Krishna"
    print(king)

coup()
print(king)
```

```
Krishna
Krishna
```

```
# global scope

king = "Rahul"

def kingdom():
    # global keyword will change value globally
    global king

    king = "Krishna"
    print(king)

print(king)
kingdom()
print(king)
```

```
Rahul
Krishna
Krishna
```

```
# Quiz

# What does the following code print?
a = 10
def f():
    print('Hello')
    b = 20
f()
print(a)
print(b)
```

```
Hello
10

----------------------------------------------------------------------
-----
```

```
NameError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_6639/399974
6037.py in <module>
      6 f()
      7 print(a)
----> 8 print(b)

NameError: name 'b' is not defined
```