Default Parameter

- We pass while defining a function

```
## Default values in print function?

# Family



print("Rahul", "Deepanshu", "Rahul","Jaguar", sep = "->")
print("Karthik")
```

```
Rahul->Deepanshu->Rahul->Jaguar
Karthik
```

```
print?
```

```
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current
sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
Type:      builtin_function_or_method
```

```
def family_intro(father, mother, sibling = "Harry potter"):
    print("Name of father is", father)
    print("Name of mother is", mother)
    print("Name of sibling is", sibling)


family_intro("James", "Lilly")
```

```
Name of father is James
Name of mother is Lilly
Name of sibling is Harry potter
```

```
family_intro("James", "Lilly", "Hermoine")
```

```
Name of father is James
Name of mother is Lilly
Name of sibling is Hermoine
```

```
father = "James"
mother = "Lilly"
sibling = "Harry potter"

print(sibling, father, mother)
```

```
Harry potter James Lilly
```

```
def family_intro(sibling = "Harry potter", father, mother):
    print("Name of father is", father)
    print("Name of mother is", mother)
    print("Name of sibling is", sibling)
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/15049
32931.py", line 1
    def family_intro(sibling = "Harry potter", father, mother):
                                                       ^
SyntaxError: non-default argument follows default argument
```

```
family_intro()
```

```
def another(a = 5, b, c):
    print(a, b, c)
```

```
  File
"/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/19112
16228.py", line 1
    def another(a = 5, b, c):
                          ^
SyntaxError: non-default argument follows default argument
```

```
another(4, 5)
```

Keyword Arguments

- It's a game of using parameter's name while calling the function
- When the number of arguments are too many use it
- No need to remember the order / position of arguments now

```
def food(starter, main, dessert, soups, paan, snacks, price):
    print(starter, main, dessert, soups, paan, snacks, price)

food(starter = "chai", main = "paneer", soups = "lentils", dessert =
"Rasmalai", paan = "paan", price = 420, snacks = "Samosa")
```

```
chai paneer Rasmalai lentils paan Samosa 420
```

```
## Default follows positional
```

```
def family(father, mother, sibling = "Harry potter"):
    print(father, mother, sibling)

family(mother = "Lilly", father = "James", sibling = "Hermoine")
```

```
James Lilly Hermoine
```

```
## date month year style example

## date / month / year = European
## month / date / year = US

def print_date(date, month, year, style = 0):
    if style == 0: ## European style
        print(date, "/", month, "/", year)
    elif style == 1: ## US style
        print(month, "/", date, "/", year)
    else:
        print("Invalid style")


print_date(1, 6, 2022, 1)
```

```
6 / 1 / 2022
```

Docstrings

- Should I have Kept some Documentation?
- Tell me something about yourself please...

```
def intro(name, age):
    """
    This is an intro function
    name: You need to put your name here
    age: Put your age here
    """
    print(name, age)

intro?
```

```
Signature: intro(name, age)
Docstring:
This is an intro function
name: You need to put your name here
age: Put your age here
File:
```

```
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/130618
8451.py
Type:      function
```

Import module

```
# import name_module/library
```

```
## import this
```

```
import math
```

```
math.pi
```

```
3.141592653589793
```

```
math.floor(2.34)
```

```
2
```

```
int(math.ceil(2.34))
```

```
3
```

```
def mul(a, b):
    c = a * b
    return c
```

```
print(mul(2, 3))
```

```
6
```

```
a = int(input())
```

```
 2 3 4
```

```
----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/230584
3072.py in <module>
----> 1 a = int(input())

ValueError: invalid literal for int() with base 10: '2 3 4'
```

```
R, C = map(int, input().split())
```

```
 2 2.4
```

```
----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/146802
0298.py in <module>
----> 1 R, C = map(int, input().split())

ValueError: invalid literal for int() with base 10: '2.4'
```

```
print(R, C)
```

```
2 3
```

```
int("2.4")
```

```
----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_7014/290379
3479.py in <module>
----> 1 int("2.4")

ValueError: invalid literal for int() with base 10: '2.4'
```

```
2 ** 3
```

```
8
```

```
int(8 ** (1/3))
```

```
2
```

```
dir(math)
```

```
['__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'ceil',
 'comb',
 'copysign',
 'cos',
 'cosh',
 'degrees',
 'dist',
 'e',
 'erf',
 'erfc',
 'exp',
 'expm1',
 'fabs',
 'factorial',
 'floor',
 'fmod',
 'frexp',
 'fsum',
 'gamma',
 'gcd',
 'hypot',
 'inf',
 'isclose',
 'isfinite',
 'isinf',
 'isnan',
 'isqrt',
 'lcm',
 'ldexp',
 'lgamma',
 'log',
 'log10',
 'log1p',
 'log2',
 'modf',
 'nan',
 'nextafter',
 'perm',
 'pi',
 'pow',
 'prod',
 'radians',
 'remainder',
 'sin',
 'sinh',
 'sqrt',
 'tan',
 'tanh',
 'tau',
 'trunc',
 'ulp']
```

```
## Square root
```

```
## Area of circle
```