

# Python\_basic\_programming\_23

In [ ]:

```
1. Create a function that takes a number as an argument and returns True or False depending on whether the number is symmetrical or not.  
A number is symmetrical when it is the same as its reverse.  
Examples:  
is_symmetrical(7227) True  
is_symmetrical(12567) False  
is_symmetrical(44444444) True  
is_symmetrical(9939) False  
is_symmetrical(1112111) True
```

In [1]:

```
def is_symmetrical(in_num):  
    if str(in_num) == str(in_num)[::-1]:  
        print(f'{in_num} {True}')  
    else:  
        print(f'{in_num} {False}')  
is_symmetrical(7227)  
is_symmetrical(12567)  
is_symmetrical(44444444)  
is_symmetrical(9939)  
is_symmetrical(1112111)
```

```
7227 True  
12567 False  
44444444 True  
9939 False  
1112111 True
```

In [ ]:

```
2. Given a string of numbers separated by a comma and space, return the product of the numbers.  
Examples:  
multiply_nums("2, 3") 6  
multiply_nums("1, 2, 3, 4") 24  
multiply_nums("54, 75, 453, 0") 0  
multiply_nums("10, -2") -20
```

In [3]:

```
def multiply_nums(in_string):  
    out_string = in_string.replace(' ', '').split(',')  
    out_num = 1  
    for ele in out_string:  
        out_num*=int(ele)  
    print(f'{in_string} {out_num}')
```

```
multiply_nums("2, 3")  
multiply_nums("1, 2, 3, 4")  
multiply_nums("54, 75, 453, 0")  
multiply_nums("10, -2")
```

```
2, 3 2  
2, 3 6  
1, 2, 3, 4 1  
1, 2, 3, 4 2  
1, 2, 3, 4 6  
1, 2, 3, 4 24  
54, 75, 453, 0 54  
54, 75, 453, 0 4050  
54, 75, 453, 0 1834650  
54, 75, 453, 0 0  
10, -2 10  
10, -2 -20
```

In [ ]:

```
3. Create a function that squares every digit of a number.  
Examples:  
square_digits(9119) 811181  
square_digits(2483) 416649  
square_digits(3212) 9414  
Notes:  
The function receives an integer and must return an integer.
```

In [4]:

```
def square_digits(in_num):  
    in_list = [str(int(ele)**2) for ele in str(in_num)]  
    out_list = ''.join(in_list)  
    print(f'{in_num}{int(out_list)}')
```

```
square_digits(9119)  
square_digits(2483)  
square_digits(3212)
```

```
9119811181  
2483416649  
32129414
```

In [ ]:

```
4. Create a function that sorts a list and removes all duplicate items from it.  
Examples:  
setify([1, 3, 3, 5, 5]) [1, 3,5]  
setify([4, 4, 4, 4]) [4]  
setify([5, 7, 8, 9, 10, 15]) [5, 7, 8, 9, 10, 15]  
setify([3, 3, 3, 2, 1]) [1, 2, 3]
```

In [6]:

```
def setify(in_list):  
    out_list = sorted(set(in_list))  
    print(f'{in_list}{out_list}')
```

```
setify([1, 3, 3, 5, 5])  
setify([4, 4, 4, 4])  
setify([5, 7, 8, 9, 10, 15])  
setify([3, 3, 3, 2, 1])
```

```
[1, 3, 3, 5, 5][1, 3, 5]  
[4, 4, 4, 4][4]  
[5, 7, 8, 9, 10, 15][5, 7, 8, 9, 10, 15]  
[3, 3, 3, 2, 1][1, 2, 3]
```

In [ ]:

```
5. Create a function that returns the mean of all digits.  
Examples:  
mean(42) 3  
mean(12345) 3  
mean(666) 6  
Notes:  
1. The mean of all digits is the sum of digits/how many digits there are (e.g. mean of digits in 512 is (5+1+2)/3(number of digits) =8/3=2).  
2. The mean will always be an integer.
```

In [7]:

```
def mean(in_num):  
    in_list = [int(ele) for ele in str(in_num)]  
    out_num = sum(in_list)/len(str(in_num))  
    print(f'Mean of{in_num} {out_num:.0f}')
```

```
mean(42)  
mean(12345)  
mean(666)
```

```
Mean of42 3  
Mean of12345 3  
Mean of666 6
```