

functions (part-2)

- **Write a function to check whether the number is even or not, if yes then return true otherwise return false.**

```
#first do the question without using the functions, then do the problem using
#the functions

def checkNumber(num):
    if num%2==0 :
        return "true";
    else :
        return "false";

x=checkNumber(7);
print(x);  # or i can write print(checkNumber(7))
```

- Now you can use this function anywhere to check whether the number is even or not by just calling the function.
- **Take an example of the evaluation if you get a question like you have to count the even numbers in a limit of 1-100, then you can simply use this function to determine an even number and count it.**

```
#First solve this question without using the function, then tell the to use
#function, also tell them the importance.

#step 1: I will declare a function to check the even number
def checkNumber(num):
    if num%2==0 :
        return "true";
    else :
        return "false";

#step 2: I can a run a loop within the given range and use this function to do
#the job.
count=0
for i in range(100):
    x=checkNumber(i)
    if (x=="true"):
        count=count+1

print(count);
```

- **Use the above function to count the odd numbers in a given limit.**

```
#First solve this question without using the function, then tell the to use
#function, also tell them the importance.
```

```

#step 1: I will declare a function to check the even number
def checkNumber(num):
    if num%2==0 :
        return "true";
    else :
        return "false";

#step 2: I can a run a loop within the given range and use this function to do
#the job.
count=0;
for i in range(100):
    x=checkNumber(i)
    if (x=="false"):
        count=count+1

print(count);

```

- **Write a function to multiply the elements of two arrays**

```

arr1=[20,40,60,80];
arr2=[3,7,9,4];

def multiply(a,b):
    final=[]
    for i in range(len(a)):
        product=a[i]*b[i]
        final.append(product)
    print(final)

print(multiply(arr1,arr2)) # [60,280,540,320]

```

Inbuilt Functions

Definition: Inbuilt functions are those functions that are created by JavaScript developers and you use them to make your job easy.

- **Number:** Used to convert a string into a number.

```

x="12";
y="24";
print(x+y); #1224

num1=int(x);
num2=float(y);
print(num1+num2); #36

```

- **toString:** Used to convert to string.

```
num1=12;
num2=36;
print(num1+num2); #48

x=str(num1)
y=str(num2)
print(x+y);    #1236
```

- Talk about **append** and **Pop**, They were also inbuilt functions to perform their specific task in case of arrays.

Importance of Standard Documentation

- Introduce them to documentation, to show them the logic behind these functions and also to show them some more inbuilt functions.
- Take them to MDN Docs, and make them familiar.
- What is **Documentation**?
 1. Documentation is the true source of information about the tech that you want to use, wrote by the developers, who created them.
 2. Best way is to read the documentation if you want to know about the tech that you are using.
 3. These are like best notes already written for you, that you can see and learn many things.
 4. It is not important to learn everything, whenever you want to use these things just go and refer the documentation.
- For python documentation, we can refer [Python docs](#)

- **Lists - some in-built functions**

- **append():- Adds an element at the end of the list**

```
a = ["apple", "banana", "cherry"]
b = ["Ford", "BMW", "Volvo"]
a.append(b)

print(a) #appends an element to the end of the list.
```

- **pop():- It returns the removed element from the List.**

```
list_1=[1,2,3,4];
x=list_1.pop();
print(x); #It will return the removed element, in this case 4str1 = "this is string example....wow!!!";
str2 = "is";

print str1.rindex(str2)
print str1.index(str2)
```

- **rindex()**:- It returns the last index where that particular element is present after searching.

```
str1 = "this is string example...wow!!!";
str2 = "i";

print str1.rindex(str2)    #5
```

- **index()**:- It returns the first appearance of the searched value.

```
str1 = "this is string example...wow!!!";
str2 = "i";

print str1.index(str2)    #2
```

- **insert()**: method inserts the specified value at the specified position.

```
fruits = ['apple', 'banana', 'cherry']

fruits.insert(1, "orange")
```

- **del()**:- It removes the element from the specified index:
- **join()**:-It will join the elements of the list with the thing that you have passed in it (separator).

```
#This will be an activity basically.
#Before this ask them how to join the following list to get an output
#something like "chunnumunnutunnu";
list =["chunnu","munnu","tunnu"];

#we can basically use concatenation

bag=""
for i in range(len(list)):
    bag=bag+list[i];

print(bag); #chunnumunnutunnu
```

Now explain to them the inbuilt join() function.

```
list =["chunnu","munnu","tunnu"];
x= "".join(list)

print(x); #chunnumunnutunnu
```

- **slice()**:- The `slice()` function returns a slice object.

A slice object is used to specify how to slice a sequence. You can specify where to start the slicing, and where to end. You can also specify the step, which allows you to e.g. slice only every other item.

```
a = ["a", "b", "c", "d", "e", "f", "g", "h"]
x = slice(2)
print(a[x])  #['a', 'b', 'c', 'd', 'e']
```

- **Go to Strings Section.**

- **upper().**
- **lower().**
- **split():-** It splits a string into a list. You can specify the separator, default separator is any whitespace.

```
txt = "hello, my name is Peter, I am 26 years old"

x = txt.split(", ")

print(x)

['welcome', 'to', 'the', 'jungle']
```

Problem 1: Create our own Split function .

```
str="the quick      brown fox jumps";
def mysplit(str):
    bag=""
    output=[];
    for i in range(len(str)):
        if str[i]!=" ":
            bag=bag+str[i]
        else:
            if(bag!=""):
                output.append(bag);
                bag="";
    print(output)

mysplit(str)
```