



VIETCODE ACADEMY INTRODUCTION TO PROGRAMMING

Class 5: Function





TEACHER INFORMATION

TEACHER



TEACHING ASSISTANT





Function and why

We have optimized the code programs with the for function, instead of writing 1000 lines of print, we only need 2 to 3 lines

So let's try for example the problem of printing odd numbers from 1 to n, but now the number of n we want the program to run is 10.

Then writing 10 for and 10 if to print odd numbers is pointless and laborious

So the function helps us to solve this problem just by defining and **calling** the function name 10 times in a very short way

=> A function is a reusable subroutine that performs some operation or returns a value through its name.





Use function



So to handle the problem of printing odd numbers from 1 to n using a function, how should we program it?

```
def print_odd_numbers(n):  
    for i in range(1,n+1):  
        if(i%2!=0):  
            print(i)  
  
print_odd_numbers(5)  
print_odd_numbers(100)
```

In the above code, we have defined the function `print_odd_numbers` which has the function to print odd numbers from 1 to n, and then we call the function with 2 numbers n is 5 and n is 100. So now the terminal will show odd numbers. from 1 to 5 and then odd numbers from 1 to 100.



Structure of function

As seen in the above code, the function has the following basic structure:

```
def function_name(parameter1, parameter2, parameter3, ...) :  
    code
```

First you need to name the function and name the parameters to be used (the number of parameters depends on the purpose of the function). Then the code will execute a set of statements using the parameters we declared next to `function_name`. The parameters here can be string, int, bool or even list.

* Note that this is the function definition, and for the code to execute, we need to call the function name and the parameters have values corresponding to what we have defined. For example: `function_name(1,2,3,...)`



Exercises

Define a function named `print_sum` that prints the sum of two parameters `a` and `b`.

Define a function named `print_sum_of_odd_numbers` that prints the sum of all odd numbers in the list `a` parameter

=> The input parameter can have different number of parameters or different types depending on the purpose of use. However when calling the function we need to give the **corresponding** values

```
def print_sum(a,b):  
    result=a+b  
    print(result)
```

```
print_sum([3,6])
```

```
def print_sum_of_odd_numbers(a):  
    result=0  
    for i in range(len(a)):  
        if(a[i]%2!=0):  
            result+=a[i]  
    print(result)
```

```
print_sum_of_odd_numbers([1,3,4,5,7,12])
```





Return value to function and why

In addition to printing values, the function can also receive values. Because in programming we not only print out, but we need to use the value to perform other operations.

For example, using a function to get the sum of two parameters a and b, then print the square of that value, how do we program it?

```
def sum_of_two_number(a,b):  
    result=a+b  
    return result  
  
result=sum_of_two_number(2,3)  
result=result*result  
print(result)
```





Syntax of function to retrun value

As seen in the example above, to return the value to the function has the following structure:

```
def function_name(parameter1, parameter2, parameter3, ...) :  
    code  
    return value
```

Now function_name will receive the value (value). Value can be a string, int, bool, or even a list depending on the purpose of the function

*** Note that return is the same as break once run, all the code below in the function will not be run and exit returns the value to the function.**



Exercises

```
def mysterious_function(a):  
    result=[]  
    for i in range(1,a+1):  
        if(i%2==0):  
            result.append(i)  
    return result
```

The mysterious_function function will receive a list of even numbers from 1 to the parameter a .



```
def mysterious_function(a):  
    if(a%2==0):  
        return True  
    else:  
        return False
```

The mysterious_function will check if the number a is even. If the number is even, the value will be True, if not, the function will have the value False



PRACTICE WITH INCLASS5