

Objectives for class 10

--- Chapter 6 ---

- 6.1 To understand what is a function (§[6.1](#))
- 6.2 To define functions with formal parameters (§[6.2](#)).
- 6.3 To distinguish the differences between the functions that return and do not return a value (§[6.4](#)).
- 6.4 To invoke functions with actual parameters (i.e., arguments) (§[6.3](#)).
- 6.5 To determine the scope of variables (§[6.8](#))

Let us use a loop to keep reading user's options

System Design

While **True**:

Step 1: display a menu to user

Step 2: ask user to enter an option

Step 3: if user enters 'a', print "hello world" **once**

Step 4: if user enters 'b', print "hello world" **twice**

Step 5: if user enters 'c', print "hello world" **100** times

Step 6: if user enters 'd', **stop** the program.

```
while True:
```

No function

```
    print("a. Print hello world once")
```

```
    print("b. Print hello world twice")
```

```
    print("c. Print hello world one hundred times")
```

```
    print("d. Stop the program")
```

```
    choice=input("Please enter your choice")
```

```
    if choice == 'a':
```

```
        print("Hello World")
```

⇒ Let us name sequence 1 **print1hello**

```
    elif choice == 'b':
```

```
        print("Hello World")
```

```
        print("Hello World")
```

⇒ Let us name sequence 2 **print2hello**

```
    elif choice == 'c':
```

```
        for i in range(0,100):
```

```
            print("Hello World")
```

⇒ Let us name sequence 3 **print100hello**

With Functions

```
def print1hello():  
    print("Hello World")  
  
def print2hello():  
    print("Hello World")  
    print("Hello World")  
  
def print100hello():  
    for i in range(0,100):  
        print("Hello World")
```

```
while True:  
    print("a. Print hello world  
once")  
    print("b. Print hello world  
twice")  
    print("c. Print hello world one  
hundred times")  
    print("d. Stop the program")  
  
    choice=input("Please enter your  
choice")  
  
    if choice == 'a':  
        print1hello()  
    elif choice == 'b':  
        print2hello()  
    elif choice == 'c':  
        print100hello()  
    elif choice == 'd':  
        break
```

Function is a named sequence of statements

- A sequence of statements
- Executed only when you CALL it

```
def print1hello():  
    print("Hello World")
```

```
>>>print1hello()  
Hello World
```

Practice

- Read program and answer questions
 - How many functions?
 - What is the execution order of lines?
 - What is the output?

```
1 def f1():
2     s = '-- Inside f1()'
3     print(s)
4
5 def f2():
6     s = '-- Inside f2()'
7     print(s)
8
9 print('Before calling f2()')
10 f2()
11 print('After calling f2()')
12 f1()
```

Pass Data Into a Function

- Passed data called **arguments**
- Arguments separated by comma
- A function “**takes**” no or **multiple** argument(s) !!!

```
>>>print1hello()  
Hello World  
>>>'Hello'.lower()  
hello  
>>>int('32')  
32  
>>>int('Hello')  
ValueError: invalid literal  
for int(): Hello  
>>>math.pow(2, 3)  
8
```

Return Data from a Function

- May “**return**” a value

Interactive mode - Python displays the return value

Script mode – Return value is lost unless you **insert** it into a statement

```
>>>int('32')
```

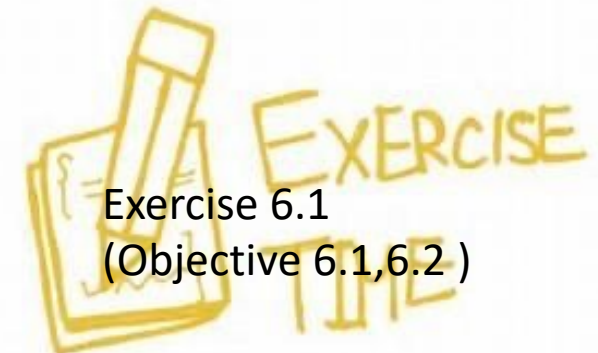
```
32
```

```
>>>print(int('32')*10)
```

```
>>>x=int('32')
```

```
>>>y=x*10
```

```
320
```



How to create your own function?

Header

```
def <name> (<parameter variable list>) :
```

Body

[

```
<statement>
```

```
<statement>
```

```
<statement>
```

```
<Next line of code>
```

Passed data are stored in **parameter variables**.

Parameter variables are used in function body.

How to create a function without a return value?

```
def <name> (<paralist>) :  
    <statement>  
    <statement>  
    <statement>  
<Next line of code>
```

```
def print1hello() :  
    print("Hello World")  
def printnhello(num) :  
    for i in range(0, num) :  
        print("Hello World")  
def nPrint(message, n) :  
    for i in range(0, n) :  
        print(message)  
def repeatnhello(num) :  
    return "Hello World"*num
```

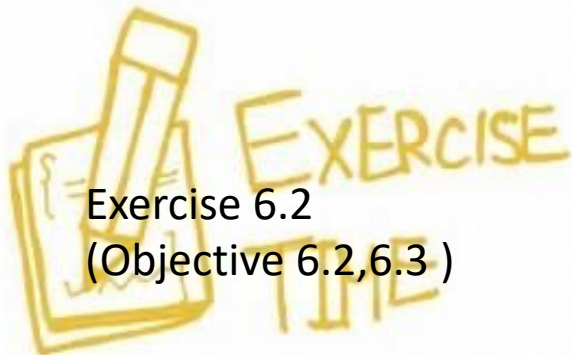
- Function body uses the data stored in **parameters**(variables) defined at function header.
- Parameters separated by comma.

How to create a function with a return value?

```
def <name> (<paralist>) :  
    <statement>  
    <statement>  
    <statement>  
    return <expression>  
<Next line of code>
```

```
def repeatnhello(num) :  
    return "Hello World"*num
```

- return statement returns a value of expression followed by.



Exercise 6.2
(Objective 6.2,6.3)

How to call a function?

Name (**arg1**, **arg2**, ...)

- Each argument is an expression

```
def printnhello(num):  
    for i in range(0, num):  
        print("Hello World")  
  
>>> printnhello(1)  
Hello World  
>>> n=1  
>>> printnhello(n)  
Hello World  
>>> printnhello(n+1)  
Hello World  
Hello World  
>>> printnhello(math.pow(n, 1))  
Hello World  
Hello World
```

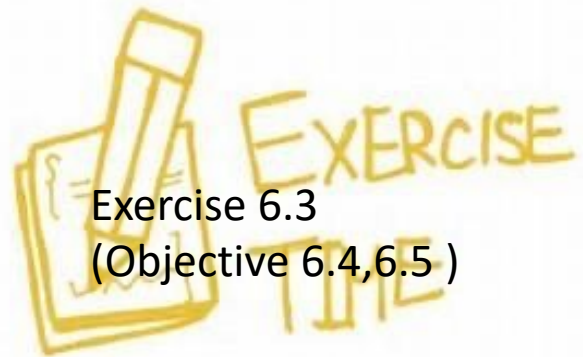
How to use a function call?

- Use function call as
 - A single statement
- Expression
- A component in another expression or statement

```
>>>def repeatnhello(num):  
    return "Hello World"*num  
  
>>>repeatnhello(1)  
Hello World  
  
>>>result = repeatnhello(1)  
>>>print(result)  
Hello World  
  
>>>result = repeatnhello(1)+"end"  
>>>print(result)  
Hello Worldend  
  
>>>print(repeatnhello(1)+"end")  
Hello Worldend
```

Variable created inside a function is local

- **Local** variable can only be used inside the function
- **Parameters** are also local



Exercise 6.3
(Objective 6.4,6.5)

```
def printnhello(num):  
    for i in range(0,num):  
        print("Hello World")  
>>>n=2  
>>>printnhello(n)  
Hello World  
Hello World  
>>>print(i)  
NameError: name 'i' is not  
defined  
>>>print(num)  
NameError: name 'num' is not  
defined
```