In [1]:
```
def say_hello():
    print("Hello")
```

In [2]:
```
say_hello()
```

```
Hello
```

Lets say we had a more complex code,

In [3]:
```
def say_hello():
    print("Hello")
    print("how")
    print("are")
    print("you?")
```

In [4]:
```
say_hello()
```

```
Hello
how
are
you?
```

In [10]:
```
def say_hello(name):
    print(f'Hello {name}' )
```

In [11]:
```
say_hello('Keegan')
```

```
Hello Keegan
```

In [12]:
```
def say_hello(name='Default'):
    print(f'Hello {name}' )
```

In [16]:
```
say_hello()
```

```
Hello Default
```

Pictured above, we can set a default value for a function, instead of throwing an error when no input is provided.

Now, lets utilize the RETURN instead of PRINT

So what is the difference between return and print?

The return keyword allows you to actually save the result of the output of a function as a variable. The print() function simply displays the output to you, but doesnt save it for future use.

In [17]:
```
def add_num(num1,num2):
    return num1+num2
```

In [18]:
```
add_num(10,20)
```

Out[18]:
```
30
```

Loading [MathJax]/extensions/Safe.js

However, we can actually assign it to the 'result' variable

In [19]:
```
result = add_num(10,20)
```

In [20]:
```
result
```

Out[20]:
```
30
```

In [21]:
```
def print_result(a,b):
    print(a+b)
```

In [22]:
```
def reurn_result(a,b):
    return a+b
```

In [23]:
```
print_result(10,20)
```

```
30
```

In [25]:
```
result = add_num(10,20)
```

In [27]:
```
result
```

Out[27]:
```
30
```

In [29]:
```
result = print_result(10,20)
```

```
30
```

In [31]:
```
result
```

In [33]:
```
def add_num(num1,num2):
    return num1+num2
```

In [34]:
```
result = add_num(10,20)
```

In [35]:
```
result
```

Out[35]:
```
30
```

In [36]:
```
def print_result(a,b):
    print(a+b)
```

In [37]:
```
def return_result(a,b):
    return a+b
```

In [38]:
```
(a,b):
```

Loading [MathJax]/extensions/Safe.js

```
    print(a+b)
```

In [39]:
```
print_result(10,20)
```

```
30
```

In [40]:
```
def return_result(a,b):
    return a+b
```

In [41]:
```
return_result(10,20)
```

Out[41]:
```
30
```

In [42]:
```
result = return_result(10,20)
```

In [43]:
```
result
```

Out[43]:
```
30
```

In [44]:
```
def myfunc(a,b):
    print(a+b)
    return a+b
```

In [45]:
```
result = myfunc(10,20)
```

```
30
```

If we wanted to check the data types, we can do so by performing the following:

In [47]:
```
def sum_numbers(num1,num2):
    return num1+num2
```

In [48]:
```
sum_numbers(10,20)
```

Out[48]:
```
30
```

In [49]:
```
sum_numbers('10','20')
```

Out[49]:
```
'1020'
```

Pictured above, we set the values as string, which is not what we want. We need user input validation

In [ ]:
```
```

Loading [MathJax]/extensions/Safe.js