What are first class functions ?

Functions in programming language are considered first class citizens if they exhibit the following characteristics.

- They can be stored in a variables.
- They can be stored in data structures.
- They can be passed as an argument to a function.
- They can be returned as values from another function.

Python supports all these features so functions are treated as first class citizens.

This is not a python only feature by any means, many other languages treat functions as their first class citizens, namely JavaScript, PHP, Perl. They are pretty much a necessity for the functional programming style.

Defining functions

As you must already know we define functions in Python with the `def` keyword.

```
def foo():
    pass
```

The function gets stored at a memory location as shown below at the address `0x0376CB28`.

```
>>> foo
<function foo at 0x0376CB28>
```

Storing functions in variables

As function in python are first class citizens you can store the function in a variable.

```
>>> foo
<function foo at 0x0376CB28>
>>> bar = foo
>>> bar
<function foo at 0x0376CB28>
```


Even though the function is assigned to a seperate variable `bar` notice that the location of the function still stays the same. Its merely 2 variables storing the location of the same function in the memory.

Not only you can store them in simple variables but you can store functions in a list, dict and other data structures as shown below.

```
# storing in list
>>> mylist = []
>>> mylist.append(foo)

# storing in dict
>>> mydict = {}
>>> mydict['function_foo']= foo
```

Passing functions as arguments

Functions can be passed as arguments to another function. Check out the sample below.

```
def main(func):
    print(func)

def foo():
    pass

print(foo)

main(foo)
```

Output shown below…

```
<function foo at 0x0163CD68>
<function foo at 0x0163CD68>
```

The function `foo` is passed as an argument to the function `main` and we can confirm that it is the same function `foo` which `main` has received as the argument by printing the memory address (feature of CPython) of the function globally as well as within the `main` function.

Returning functions

Similarly a function can return another function.

```
def foo():
    pass

def main():
    return foo  # returns a function

bar = main()
```

As you see the memory address is still the same for `foo` and `bar`.

app.rb

```
>>> foo
<function foo at 0x009CF6F0>
>>> bar
<function foo at 0x009CF6F0>
```

Advantages of first class functions

There can be numerous benifits of this approach, some of them are listed below. You can check out the below topics to further dig deeper.

- *Decorators*
- *Dynamic Code Inspection*
- *Callbacks*