What are higher order functions?

Higher order functions are functions that have other functions as parameters or if the function returns another function

Basically functions that use other functions inside it are known as higher order functions.

Scala does support higher order functions, here are some simple example of the two types of higher order functions. One which uses a function as a parameter and the other that returns a function.

Example with function as a parameter:

We can start out by creating the function first which can take in a parameter. In this case, lets make a simple higher order function that can do some kind of math depending on what they pass.

Ex:

```
def arith(x: Int, y: Int, f(Int, Int) => Int): Int = f(x,y);
```

So the function above passes some arbitrary integers to x and y. Then, you can pass a function that takes in two integers. Those two integers will turn into another integer depending on what we choose to do to those values.

Let’s try to do some multiplication of two integers using our arith function we created above and print that result:

Start by creating a main function:

```
def main(args: Array[String]) {
        val result = arith(10, 2, (x,y) => x*y)
        println(result) //returns 20
}
```

What we did here was used two integers for the first two values and then for our function, we decided that we would multiple those integers. So when we print the result, we get 10 times 2 which is 20. If we want to do something else with those integers, we can simply change the * symbol between x and y to any operator like “+” or “-”.

Now we have learned how to use higher order function when we want to use function as parameters.

An example of a popular higher order function in Scala is the map function. Here’s how it works:

Map is a function used for list that can do something to each element in the list. Let’s try to illustrate this idea with an example using map.

First let’s initialize a list we can use

```
val mylist = List(1,2,3,4,5,6)
```

Then, lets use the map function in the main function to do something in this list.

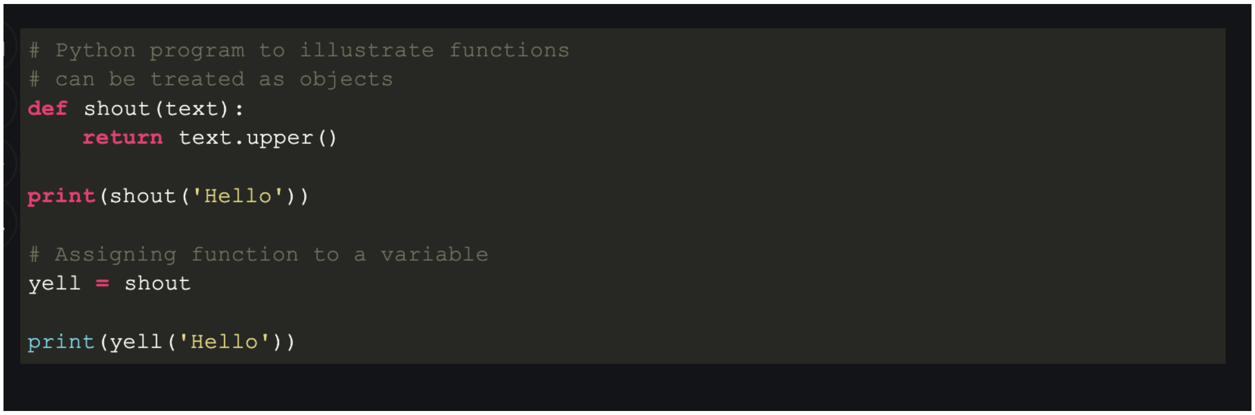
```
def main(args: Array[String]) {
        println(mylist.map(x => x*2) //returns List[2,4,6,8,10,12]
        println(mylist.map(x => x + “foo”) // List(1foo,2foo,3foo,4foo,5foo,6foo)
}
```

The first print statement multiplies every element in the list by 2. And the second print statement appends foo to every element in the list.

We are using map which “changes” a list by doing something to it. It is important to keep in mind that this doesn’t actually change the original list, “mylist”, but rather creates a new list and print it. Since, map is a function and we put another function with it, this is a higher order function. This is one of the many example of higher order methods.

Python higher order functions:
https://docs.python.org/3/library/functools.html
https://www.geeksforgeeks.org/higher-order-functions-in-python/
Properties of higher-order functions

- A function is an instance of the Object type.
- You can store the function in a variable.
- You can pass the function as a parameter to another function.
- You can return the function from a function.
- You can store them in data structures such as hash tables, lists, …

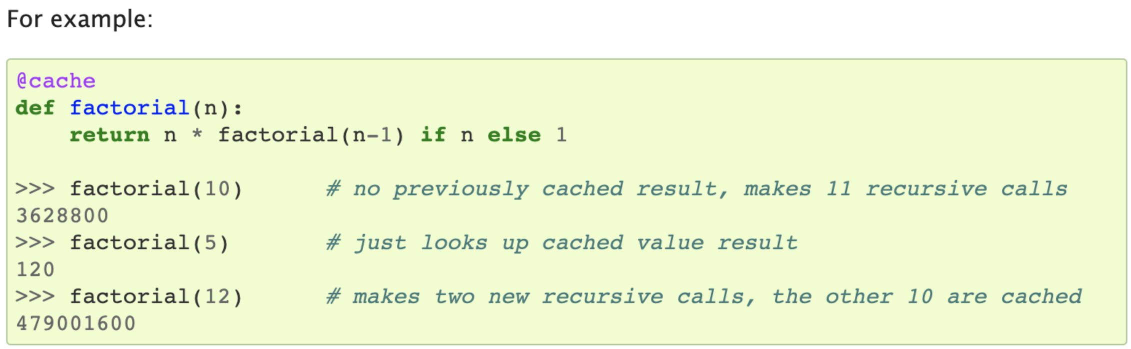


```
# Python program to illustrate functions
# can be treated as objects
def shout(text):
    return text.upper()

print(shout('Hello'))

# Assigning function to a variable
yell = shout

print(yell('Hello'))
```



For example:

```
@cache
def factorial(n):
    return n * factorial(n-1) if n else 1

>>> factorial(10)      # no previously cached result, makes 11 recursive calls
3628800
>>> factorial(5)       # just looks up cached value result
120
>>> factorial(12)      # makes two new recursive calls, the other 10 are cached
479001600
```

Similarities between python higher order functions and scala higher order functions:
They both are functions that act or return other functions, but it's easier to visualize it with a coding language that most people are more familiar with. (compare and contrast examples of scala higher order functions vs. python higher order functions)

Sources:
https://www.geeksforgeeks.org/higher-order-functions-in-scala/