Python

Python Lambda Function

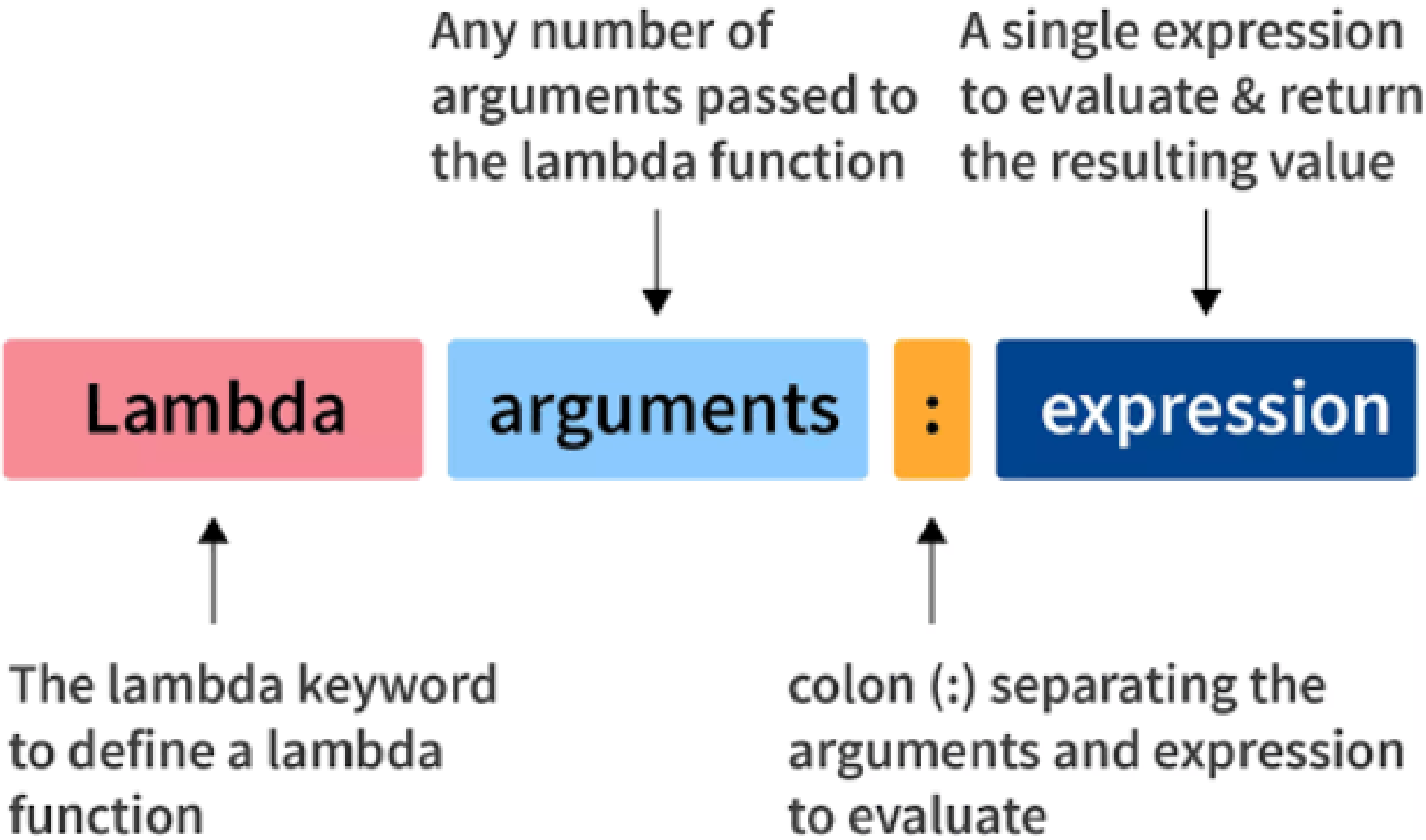
Python Lambda Functions

Python Lambda Functions are anonymous function means that the function is without a name. As we already know that the *def* keyword is used to define a normal function in Python. Similarly, the *lambda* keyword is used to define an anonymous function in Python.

Python Lambda Function Syntax

Syntax: lambda arguments: expression

- This function can have any number of arguments but only one expression, which is evaluated and returned.
- One is free to use lambda functions wherever function objects are required.
- You need to keep in your knowledge that lambda functions are syntactically restricted to a single expression.
- It has various uses in particular fields of programming, besides other types of expressions in functions.


Any number of arguments passed to the lambda function
A single expression to evaluate & return the resulting value
Lambda
arguments
:
expression
The lambda keyword to define a lambda function
colon (:) separating the arguments and expression to evaluate

Python Lambda Function Example

Python3

```
str1 = 'GeeksforGeeks'

# lambda returns a function object
rev_upper = lambda string: string.upper()[::-1]
print(rev_upper(str1))
```

Output:

```
SKEEGROFSKEEG
```

Explanation: In the above example, we defined a lambda function(**rev_upper**) to convert a string to it's upper-case and reverse it.

Example 2: Difference Between Lambda functions and def defined function

Python3

```
def cube(y):
    return y*y*y

lambda_cube = lambda y: y*y*y

# using function defined
# using def keyword
print("Using function defined with `def` keyword, cube:", cube(5))

# using the lambda function
print("Using lambda function, cube:", lambda_cube(5))
```

Output:

```
Using function defined with `def` keyword, cube: 125

Using lambda function, cube: 125
```

As we can see in the above example, both the **cube**() function and **lambda_cube**() function behave the same and as intended. Let's analyze the above example a bit more:

With lambda function	**Without lambda function**
Supports single line statements that returns some value.	Supports any number of lines inside a function block
Good for performing short operations/data manipulations.	Good for any cases that require multiple lines of code.
Using lambda function can sometime reduce the readability of code.	We can use comments and function descriptions for easy readability.

Practical Uses of Python lambda function

Example 1: Python Lambda Function with List Comprehension

In this example, we will use the lambda function with list comprehension.

Python3

```
is_even_list = [lambda arg=x: arg * 10 for x in range(1, 5)]

# iterate on each lambda function
# and invoke the function to get the calculated value
for item in is_even_list:
    print(item())
```

Output:

```
10
20
30
40
```

Example 2: Python Lambda Function with if-else

Here we are using **Max** lambda function to find the maximum of two integers.

Python3

```
# Example of lambda function using if-else
Max = lambda a, b : a if(a > b) else b

print(Max(1, 2))
```

Output:

```
2
```