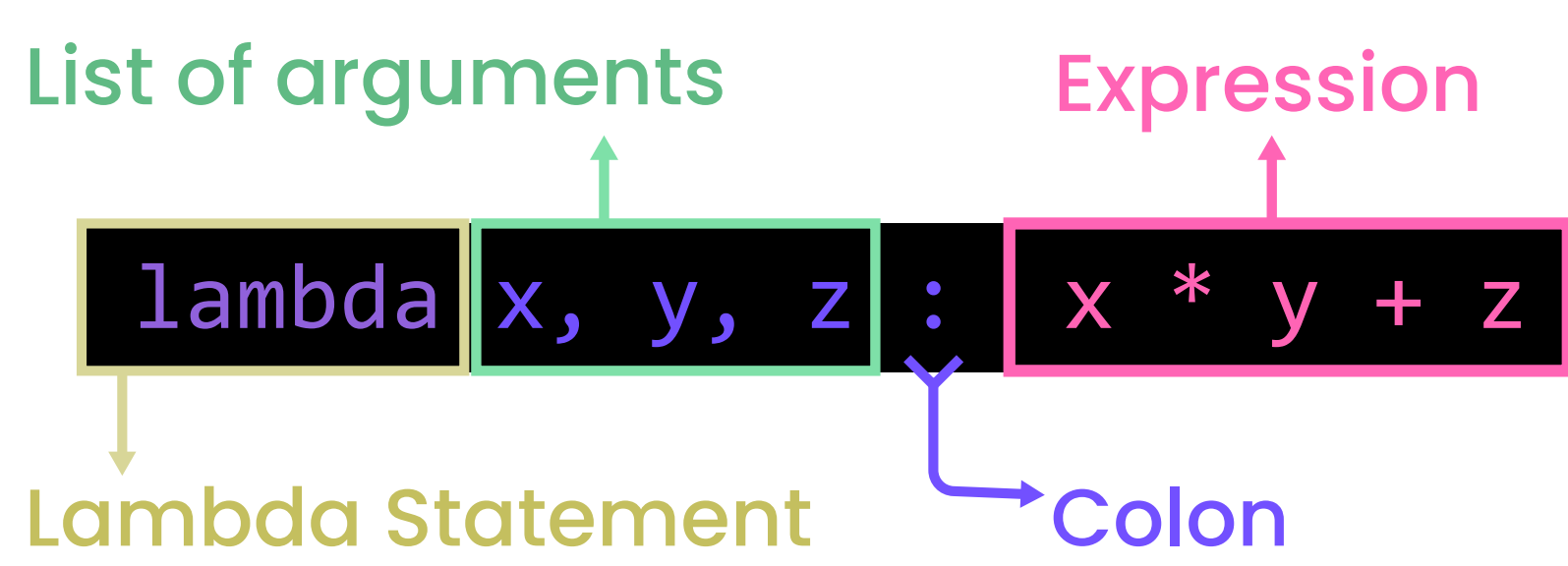
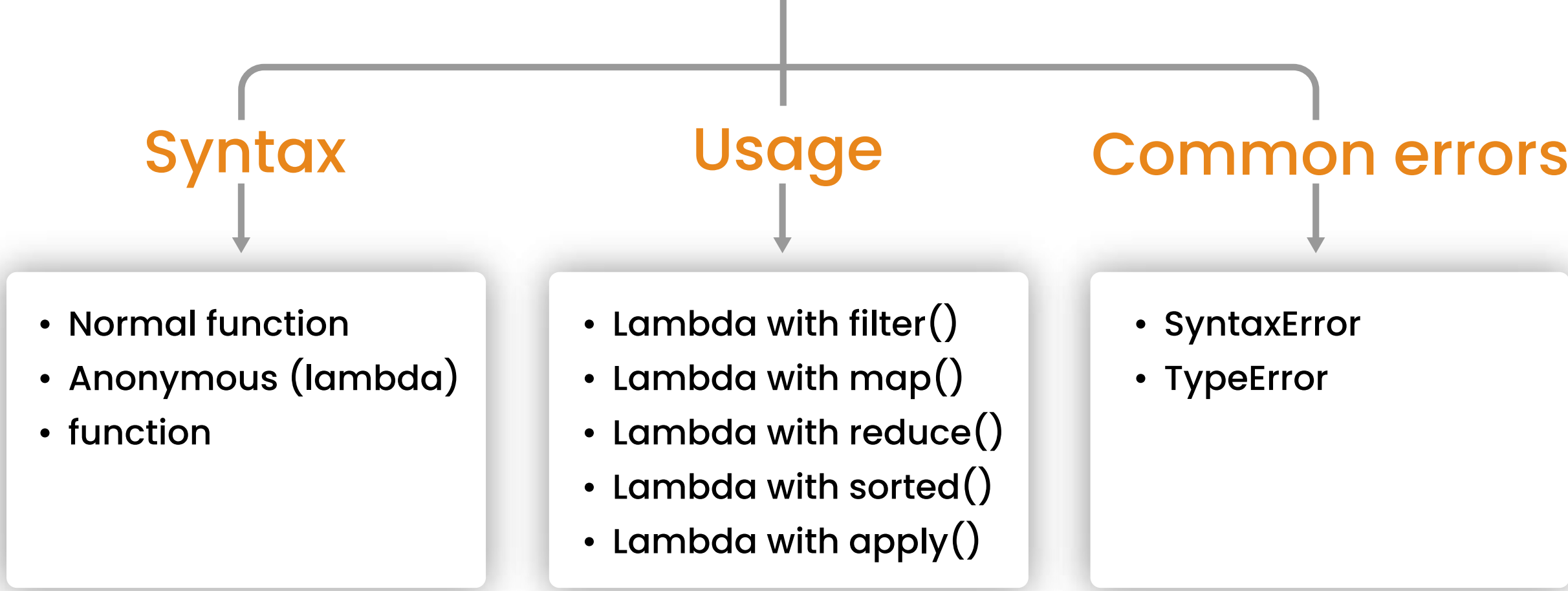




- A lambda function is a small anonymous function.
- A lambda function can take any number of arguments, but can only have one expression.
- The expression is evaluated and returned.
- Lambda functions can be used wherever function objects are required.
- Lambda expressions (or lambda functions) are essentially blocks of code that can be assigned to variables, passed as an argument, or returned from a function call, in languages that support high-order functions.
- They have been part of programming languages for quite some time.
- The main role of the lambda function is better described in the scenarios when we employ them anonymously inside another function.
- In Python, the lambda function can be utilized as an argument to the higher order functions as arguments



Lambda Functions in python



```
# Define a function using 'def'
def f(x):
    return x + 6
print(f(3.14))
```

9.14

```
# Define the same function using 'lambda'
(lambda x: x+6) (3.14)
```

9.14

```
# Define a function using 'def'
def f(x, y):
    return x + y

print('The sum of {} and {} is'.format(3.14, 2.718), f(3.14, 2.718))
```

The sum of 3.14 and 2.718 is 5.8580000000000005

```
# Define the same function using 'lambda'
print(f'The sum of pi number and euler number is {(lambda x, y: x+y)(3.14, 2.718)}.'.)
```

The sum of pi number and euler number is 5.8580000000000005.



```
# Calculate the volume of a cube using def and lambda functions
# def function
def cube_volume_def(a):
    return a*a*a
print(f'The volume of a cube using def function is {cube_volume_def(3.14)}.')
```

The volume of a cube using def function is 30.959144000000002.



```
# lambda function
print(f'The volume of a cube using lambda function is {(lambda a: a*a*a)(3.14)}.')
```

The volume of a cube using lambda function is 30.959144000000002.

Multiplication table



```
def mult_table(n):
    return lambda x:x*n
n = int(input('Enter a number: '))
y = mult_table(n)
print(y)
```

Enter a number: 5

<function mult_table.<locals>.<lambda> at 0x000002AAFA6F7160>



```
print(f'The entered number is {n}.')
for i in range(11):
    print('%d x %d = %d' %(n, i, y(i)))
```

The entered number is 5.

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

filter()



```
# This program returns a new list when the special numbers in the
list are divided by 2 and the remainder is equal to 0
special_nums = [0.577, 1.618, 2.718, 3.14, 6, 28, 37, 1729]
list(filter(lambda x:(x%2==0), special_nums))
```

[6, 28]

map()



```
# This program will multiply each element of the list with 5
and followed by power of 2.
special_nums = [0.577, 1.618, 2.718, 3.14, 6, 28, 37, 1729]
print(f'Non special numbers are {list(map(lambda x: x*5, special_nums))}')
print(f'Non special numbers list is {list(map(lambda x: pow(x, 2), special_nums))}')
```

Non special numbers are [2.885, 8.09, 13.59, 15.700000000000001, 30, 140, 185, 8645]

Non special numbers list is [0.332929, 2.6179240000000004, 7.387524, 9.8596, 36, 784, 1369, 2989441]

lambda function with if/else



```
age = int(input('Enter an age: '))
print(f'The entered age is {age}.')
(lambda age: print('Therefore, you can use a vote.')if (age>=18)
else print('Therefore, you do not use a vote.'))(age)
```

Enter an age: 6

The entered age is 6.

Therefore, you do not use a vote.

Some examples



```
def func(n):
    return lambda x: x*n
mult_pi_number = func(3.14)
mult_euler_constant = func(0.577)
print(f'The multi plicati on of euler number and pi number is equal
to {mult_pi_number(2.718)}.')
print(f'The multi plicati on of euler number and euler constant is
equal to {mult_euler_constant(2.718)}.'
```

The multi plicati on of euler number and pi number is equal to 8.53452.

The multi plicati on of euler number and euler constant is equal to 1.5682859999999998.



```
text = 'Python is a programming language.'
print(lambda text: text)
```

<function <lambda> at 0x000002AAF8BFB8B0>



```
text = 'Python is a programming language.'
(lambda text: print(text))(text)
```

Python is a programming language.



```
import itertools
from itertools import product
from numpy import sqrt
X=[1]
X1=[2]
Y=[1,2,3]
print(list(product(Y,X,X1)))
print(list(map(lambda x: sqrt(x[1]+x[0]**x[2]),product(Y,X,X1))))
```

[(1, 1, 2), (2, 1, 2), (3, 1, 2)]

[1.4142135623730951, 2.23606797749979, 3.1622776601683795]