- What is the relationship between def statements and lambda expressions ?

```
Keyword def that marks the start of the function header. A function name to uniquely identify the function.Function naming follows the same rules of writing identifiers in Python.
Lambdas are one-line methods without a name or we can say anonymous Func
They work practically the same as any other method in Python Lambdas differ from normal Python methods because they can have only one expression, can't contain any statements and their return type is a function object.

```

- What is the benefit of lambda

```
Fewer Lines of Code
Lambda functions are inline functions and thus execute comparatively faster Many times lambda functions make code much more readable by avoiding the logical jumps caused by function calls
```

- Compare and contrast map, filter, and reduce.

In [2]:

```
"""
1. map applies as a transformation to an element.
The map() function iterates through all items in the given iterable
and executes the function we passed as an argument on each of them.
Syntax : map(function, iterable(s))

2. filter accumulates only elements matching a condition.
filter() forms a new list that contains only elements that satisfy a
certain condition, i.e. the function we passed returns True Syntax :
filter(function, iterable(s))

3. reduce accumulates all elements to a single value, by using immutable values
reduce() works by calling the function we passed for the first two items
in the sequence. The result returned by the function is used in another call
to function alongside with the next (third in this case), element Syntax :
reduce(function, sequence[, initial])

"""
```

Out[2]:

```
'\n1. map applies as a transformation to an element.\nThe map() function iterates th
rough all items in the given iterable \nand executes the function we passed as an ar
gument on each of them.\nSyntax : map(function, iterable(s))\n\n2. filter accumulate
s only elements matching a condition.\nfilter() forms a new list that contains only
elements that satisfy a \ncertain condition, i.e. the function we passed returns Tru
e Syntax :\nfilter(function, iterable(s))\n\n3. reduce accumulates all elements to a
single value, by using immutable values\nreduce() works by calling the function we p
assed for the first two items \nin the sequence. The result returned by the function
is used in another call \nto function alongside with the next (third in this case),
element Syntax :\nreduce(function, sequence[, initial])\n\n'
```

In [3]:

```python
### Map functions
Animal = ["Anteater", "Dog", "Elephant", "Giraffe", "Cat"]
map_object = map(lambda s: s[0] == "A", Animal)


for i in map_object:
    print(i)
```

```
True
False
False
False
False
```

In [4]:

```python
## Reduce function
from functools import reduce

list = [4, 3, 8, 3]
print(reduce(lambda x, y: x + y, list))
print("With an initial value: " + str(reduce(lambda x, y: x + y, list, 10)))
```

```
18
With an initial value: 28
```

In [ ]:

```python
What are function annotations, and how are they used?
```