1. Function Design for Unit Conversions:

In [1]:
```
import unit_conversions as uc
```

Miles and Kilometers:

In [16]:
```
x=3
a= uc.miles_to_kilos(x)
print(f" {x} miles is equal to {a:.2f} kilos")
```

```
 3 miles is equal to 4.83 kilos
```

In [3]:
```
#### Kilos to Miles:
y=5
b=uc.kilos_to_miles(y)
print(f" {y} kilometers is equal to {b:.2f} miles")
```

```
 5 kilometers is equal to 3.11 miles
```

Farenheit and Celsius:

In [4]:
```
### Farenheit to Celsius:
x=98.6
c= uc.far_to_cel(x)
print(f" {x} degrees Farenheit is equal to {c:.2f} degrees Celsius")
```

```
 98.6 degrees Farenheit is equal to 37.00 degrees Celsius
```

In [5]:
```
### Celsius to Farenheit:
y=3
d= uc.cal_to_far(y)
print(f" {y} degrees Celsius is equal to {d:.2f} degrees Farenheit")
```

```
 3 degrees Celsius is equal to 37.40 degrees Farenheit
```

Kilograms and Pounds

In [6]:
```
### Kilo to Pounds:
x=2.4
e= uc.kilo_to_lbs(x)
print(f" {x} kilos is equal to {e:.2f} pounds")
```

```
 2.4 kilos is equal to 5.29 pounds
```

In [7]:
```
### Pounds to Kilo:
y=3.5
f= uc.lbs_to_kilo(y)
print(f" {y} pounds is equal to {f:.2f} kilos")
```

```
 3.5 pounds is equal to 1.59 kilos
```

Namespaces:

In [15]:
```
mylist=[1,2,"dogs",23,45]
type(mylist)
len(mylist) #outputs 5
```

Out[15]:
```
42
```

In [12]:
```
import mylen1
mylen1.len(mylist)
```

Out[12]:
```
42
```

In [14]:
```
from mylen1 import len

len(mylist)
```

Out[14]:
```
42
```

Now, Python considers the len() function as my module's function- not the base Python function. When I ran len(mylist) again, it outputted 42, rather than 5. This is because when I ran "from mylen import len", it imported my len function into the current namespace. Basically, it overwrote the built-in len function.