The airspeed of an unladen European swallow is 11 meters per second There are 1609.34 meters in a mile, and 3600 seconds in an hour. Write a short line of code that calculates the airspeed of an unladen European swallow in miles per hour. Hint: if you get 5 mph, check your logic!

Okay, you can also write it in a more general way by leaving your air-speed input unchanged and writing below that:

air_speed = air_speed*3600/1609.34

In [2]:
```
air_speed = 24.606 #miles per hour;

#there are a few ways to print out a number in Python - let's see some of t
hem
print("The air speed of an unladen European swallow is {} mph".format(air_s
peed)) #format puts the arguments in place of the {}
print("The air speed of an unladen {0} swallow is {1} mph".format("European
", air_speed)) #format puts the arguments in, {0} means first argument, {1}
is the second, etc.
print(f'The air speed of an unladed European swallow is {air_speed} mph') #
here's another way to do formatting
#you can use :.xf to specify x digits after the decimal place
print(f'The air speed of an unladed European swallow is {air_speed:.0f}
mph')
print(f'The air speed of an unladed European swallow is {air_speed:.1f}
mph')
```

```
The air speed of an unladen European swallow is 24.606 mph
The air speed of an unladen European swallow is 24.606 mph
The air speed of an unladed European swallow is 24.606 mph
The air speed of an unladed European swallow is 25 mph
The air speed of an unladed European swallow is 24.6 mph
```

Congrats, programmer!

We'll be writing substantially more complicated and interesting programs in this course, but you should pat yourself on the back for reaching your first programming milestone: writing your version of the famous "Hello, world!" program that has helped introduce nearly every modern programmer to the world of coding.

Getting help

To learn more about any Python command or function, you can type `?<function name>` into a code cell (relacing "`<function name>`" with the actual function name, e.g. `print`) and press shift + return to execute it. Try getting help using this approach for the following functions: `print`, `input`, and `open`. Click the `x` to close the help documentation once you're done scrolling through it. If you're already familiar with reading API documentation, you may notice some interesting features of these functions. For example, together `print`, `input`, and `open` constitute a basic but powerful set of functions that allow you to read and write to files-- including remotely (e.g., web pages)! And don't worry if the documentation seems inscrutible right now. As you learn more about Python programming you'll start to build up a sense of the contexts in which different parts of the documentation apply, and you'll also build up a vocabulary that will allow you to communicate (e.g., by reading and writing documentation asking questions, etc.) with other programmers. Try running (and tweaking) the following commands to get a sense of how the built-in help functions work:

```
>> ?print
>> help(print)
```

In [ ]:
```
?print #note that the help box opens on the right ---->
```