Tutorial: Structures

To prepare for your homework and get accustomed to making structures, we'll make some structures that closely follow the examples in lecture.

Do this work in your `tutorials` directory, and add the results to git. Do your editing in `gvim` for this tutorial, even if you normally use `gedit` or `idlde` - there are some handy tricks for replicating code below.

First, open a file called `CarStruct__DEFINE.pro`. In it, create a structure definition.

```
pro CarStruct__DEFINE
    dummycar = {CarStruct,$
        miles: 0L}
end
```

Then, in IDL, create an instance of a `CarStruct`:
`car = {CarStruct}`

And set its mileage:
`car.miles = 300`

Now let's change the `CarStruct` definition. Add a new field called `year` to the structure definition. Remember, you may need to `.reset_session` to be allowed to override the structure definition.

This next step can be made easier if you use `gvim`'s `yy` and `p` commands to copy and paste whole lines rather than re-typing the whole command each time.

Make 5 `cars` with the following properties:

```
car1: 25000 miles, 2005
car2: 225000 miles, 1987
car3: 100000 miles, 2002
car4: 19000 miles, 1995
car5: 28000 miles, 2007
```

Create an array containing these cars.
HINT: This is a great time to make a script in order to save yourself some typing!
In `gvim`, enter 'command mode' and use `yy` to copy a line, then `p` to paste it. For example, type:
`carrarr[0] = car1`
then press `esc`, then `yyp`.
You should now have two copies of the line.
Next, move your cursor over the `[0]` and press `r1` to replace the `0` with a `1`.
One last trick: to replace the `car1` with `car2`, do this (make sure you are starting with your cursor hovering over the `1`though!):
Press `esc`, then type `wwwcw`
Now you are back in "insert" mode and the old 'word' `car1` has been deleted.
This hint is intended to save you a little time; try finishing the tutorial using `yy`, `p`, and `w`. Remember the VIM guide on the wall if you're lost.

Plot the miles driven versus the year of the cars. Use a reasonable plot symbol (i.e., don't use lines).

Using a `for` loop, add 1000 miles to each car's mileage. You can use `n_elements` to get the number of elements in the car array.

Use `oplot` to overplot the updated miles versus years in a different symbol.

Now *not* using a for loop, add 1 year to each car's date. Again, overplot the miles vs. years for the updated values in a different symbol. You should now have 3 symbols for each car.