

# Defensive programming

It's like defensive driving



# Why?

- At many times you are unsure about the values of variables you will get.
- Either because it is provided from external source or it is generated by some complex system with random component.

# So what to look for?

- At critical points think of accepted ranges of variables.
- For both experimental and simulated data this is a very good exercise that lets you understand the problem you are working on.

# What to do?

- There are two possible outcomes if a variable gets an unexpected value. You can either:
  - Replace it with a different one
  - Terminate the whole program

# Exceptions

- Use when you know how to handle the “unexpected” for:
  - Alternative solution to solving the same issue
  - Maintaining an error log

# Exceptions syntax

- MATLAB

```
try
    throw(MException('foo')) ;
catch ME
    ME.stack
    ME.message
end
```

# Exceptions syntax

```
rethrow (ME) ;
```



# Exceptions syntax

- Python

```
try:
    except IOError:
...
    except ValueError:
...
    except:
...
    finally:
...
...
```

# Exceptions syntax

```
raise IOError
```

```
raise
```

# Assertions

- **MATLAB:**

```
assert (expression)
```

```
assert (expression, 'msgString')
```

- **Python:**

```
assert expression, 'msgString'
```

# MATLAB extras

- `error`
- `warning`
- `pause` or `keyboard`

# Naming conventions

# Loops

- $i, j, k \dots$
- Try something more meaningful to avoid confusion

```
for i in range(5):  
    weights[i] +=5
```

```
for neuronIndex in range(5):  
    weights[neuronIndex] +=5
```

# Be consistent

- `my_variable` vs `myVariable` vs `MyVariable`
- Does not really matter as long as you are consistent

# Temporary variables

- Avoid `tmp`, `temp` etc.
- Giving them sensible names does not cost too much but greatly improves readability



# Names to avoid

- Similar meaning
- Similar names
- Easily misspelled words
- Numerals in names
- L or I

# Naming conventions in maths

- $i, j, k$                       Array subscripts, loop counters
- $i, j \forall -1$                       with complex arithmetic
- $m, n$                       End of a sequence,  $i = 1, \dots, n$ ,  
number of rows ( $m$ ) and columns ( $n$ ) in a matrix
- $A, B$                       generic matrix
- $x, y, z$                       generic vectors