* KISS = Keep It Simple Stupid
* SRP: Single Responsibility Principle
* YAGNI: You Are Not Gonna Need It
* DRY: Don’t Repeat Yourself
* Meaningful and readable variable names
* Pascal naming…
* Do not change global variables inside functions
* Do not change an input variable inside functions
* Each task should have its specific function – avoid if inside functions in case like this:

def createFile(name, temp):

if temp:

….

else:

….

create two fucntions Instead:

createFile(name)

createTempFile(name)

* Small fucntions enhances “reusability”:

Function activeClients()

Function email()

Wrong:

Function emailActiveClients()

* Functions input should be fewer as possible, pass inputs as a json if they are attributes related to a same thing .
* Functions and if statements are better be Positive , it’s more readable and simple:

Function isDOMNodePresent():

* Encapsulate if statements:

For example: define a function called shouldShowSpinner and check the conditions whether spinner should be shown and everywhere you wanna check this just use that function.

* Use getter and setter: they are methods or functions which you define (inside of a class or anywhere else) to get and set a value for a variable and you can use it everywhere easily (even double the balance for instance). eg. get balance and set balance – balance is a bank account balance)
* Method chaining: if each method returns the object itself, you can chain these methods together in use. For eg. Car.setModel(“a8”).setColor(“red”).setMake(“Audi”)
* OCP : Open Close Principle
* Enough number of comments , not excessive – sometimes you can clarify a function’s utility by renaming it instead of leaving many step-by-step comments
* Write easy to remove functions (similar to decoupling)
* Decoupling and coheision