Naming conventions

*Scope*

Variables, functions, and classes should be kept as local as appropriate.

|  |  |
| --- | --- |
| **Concept** | **Prefix** |
| Global variables | gbl |
| Global functions | gblF |
| Global Classes | gblC |
| Local variables | [function name] |
| Local functions | [function name]F |
| Local classes | [function name]C |

I’ve given things prefixes so that it is easier to distinguish between global and local variables, this is because it is important to keep track of global variables. Also local variables shall be given a prefix of the function where they were defined, this is in an effort to help keep track of scope while nesting functions. The prefix for local variables does not include the prefix of the function, only the core name.

Exception: while using a for loop, the temporary variable does not need to conform to these rules.

*Variables*

|  |  |
| --- | --- |
| **Concept** | **Example** |
| All variables must have their type as a suffix | gbl\_counter\_int |
| All constants must have the main part of the name in all caps | gbl\_PI\_flt |
| Multi-word variable names must use camel casing | gbl\_multiWord\_str |
| Class variables do not have to use the local prefix. | Self.myString\_str |

|  |  |
| --- | --- |
| **Variable suffixes** | |
| **Type** | **Suffix** |
| Integer | int |
| String | str |
| Float | flt |
| Boolean | boo |
| Array | ary |
| #d Array | #dA |
| Tuple | tup |
| Dictionary | dic |
| Class instance | ins |
| Function | fun |

*Global code*

There should be as few global lines as possible, if something is global it should be imperative that it is.

There should only be this block of code that is run globally.

if \_\_name\_\_ == ‘\_\_main\_\_’:

Main()

All other functions that need to be ran should be called in the Main() function, this is to keep as much local as possible. I want to keep the amount of global code down because you can accidentally mess with other code if you’re not careful.