Type Contract Appendix

|  |  |  |
| --- | --- | --- |
| Data Type | Syntax | Example |
| Object | “obj” | """(obj) -> obj""" |
| String | “str” | """(str) -> str""" |
| Integer | “int” | """(int) -> int""" |
| Float | “float” | """(float) -> float""" |
| Boolean | “bool” | """(bool) -> bool""" |
| NoneType | “NoneType” | """() -> NoneType""" |
| List | “list” or “list of (<datatype>)” | """(list of (str)) -> list of (str or int)""" |
| Tuple | “tuple” or “tuple of (<datatype>)” | """(tuple of (str)) -> tuple of (str or int)""" |
| Set | “set” or “set of (<datatype>)” | """(set of (str)) -> set of (str or int)""" |
| Dictionary | “dict” or “dict of {<datatype>:<datatype>}” | """(dict of {str:int}) -> dict of {int:str}""" |
| Other Objects | “<class name>” | """(io.TextIOWrapper) -> NoneType""" |

Notes:

* Characters in blue are required and should not be changed
* Characters in orange are supposed to be replace. These can be replaced with any other data type that follows the formatting of this appendix
* To denote that an argument can be one of two types use the "or" operator as such: "<type1> or <type2>". This is acceptable with any of the above data types
* Containers (List, Tuple, Set, Dictionary) can contain other containers. This is illustrated by the following example: "list of (tuple of (set of (dict of {int:str})))"