## PyObject

* PyObject is an object structure that you use to define object types for Python.
* All Python objects share a small number of fields that are defined using the PyObject structure.
* All other object types are extensions of this type.
* PyObject tells the Python interpreter to treat a pointer to an object as an object.

For instance, setting the return type of the below function as PyObject defines the common fields that are required by the Python interpreter in order to recognize this as a valid Python type.

static PyObject \*method\_fputs(PyObject \*self, PyObject \*args) {
  
 char \*str, \*filename = NULL;
  
 int bytes\_copied = -1;
  
 /\* Snip \*/

* In line 2, you declare the argument types you wish to receive from your Python code:
  1. **char \*str** is the string you want to write to the file stream.
  2. **char \*filename** is the name of the file to write to.

## PyArg\_ParseTuple()

* PyArg\_ParseTuple() parses the arguments you’ll receive from your Python program into local variables:
* static PyObject \*method\_fputs(PyObject \*self, PyObject \*args) {
    
   char \*str, \*filename = NULL;
    
   int bytes\_copied = -1;
    
    
   /\* Parse arguments \*/
    
   if(!PyArg\_ParseTuple(args, "ss", &str, &filename)) {
    
   return NULL;
    
   }
* If you look at line 6, then you’ll see that PyArg\_ParseTuple() takes the following arguments:
  + **args** are of type PyObject.
  + *so hapa args ni tuple and its values should follow the type expected which in this case is "ss" that is representing two strings,,* ***str and filename***
  + **"ss"** is the format specifier that specifies the data type of the arguments to parse. (You can check out the [official documentation](https://docs.python.org/3/c-api/arg.html) for a complete reference.)
  + **&str and &filename** are pointers to local variables to which the parsed values will be assigned.