

```

int i;

double d;

char *s;

if (!PyArg_ParseTuple(args, "ids", &i, &d, &s)) {
    return NULL;
}

/* Do something interesting here. */

Py_RETURN_NONE;
}

```

Compiling the new version of your module and importing it enables you to invoke the new function with any number of arguments of any type:

```

module.func(1, s="three", d=2.0)
module.func(i=1, d=2.0, s="three")
module.func(s="three", d=2.0, i=1)

```

You can probably come up with even more variations.

The PyArg_ParseTuple Function

Here is the standard signature for **PyArg_ParseTuple** function:

```
int PyArg_ParseTuple(PyObject* tuple, char* format, ...)
```

This function returns 0 for errors, and a value not equal to 0 for success. *tuple* is the *PyObject** that was the C function's second argument. Here *format* is a C string that describes mandatory and optional arguments.

Here is a list of format codes for **PyArg_ParseTuple** function:

Code	C type	Meaning
------	--------	---------