当函数的参数不确定时，可以使用\*args 和\*\*kwargs，\*args 没有key值，\*\*kwargs有key值。

还是直接来代码吧，废话少说

1. **def** fun\_var\_args(farg, \*args):
2. **print** "arg:", farg
3. **for** value **in** args:
4. **print** "another arg:", value
6. fun\_var\_args(1, "two", 3) # \*args可以当作可容纳多个变量组成的list

result:

1. arg: 1
2. another arg: two
3. another arg: 3

\*\*kwargs:

1. **def** fun\_var\_kwargs(farg, \*\*kwargs):
2. **print** "arg:", farg
3. **for** key **in** kwargs:
4. **print** "another keyword arg: %s: %s" % (key, kwargs[key])

7. fun\_var\_kwargs(farg=1, myarg2="two", myarg3=3) # myarg2和myarg3被视为key， 感觉\*\*kwargs可以当作容纳多个key和value的dictionary

result:

1. arg: 1
2. another keyword arg: myarg2: two
3. another keyword arg: myarg3: 3

也可以用下面的形式：

1. **def** fun\_var\_args\_call(arg1, arg2, arg3):
2. **print** "arg1:", arg1
3. **print** "arg2:", arg2
4. **print** "arg3:", arg3
6. args = ["two", 3] #list
8. fun\_var\_args\_call(1, \*args)

result：

1. arg1: 1
2. arg2: two
3. arg3: 3
4. **def** fun\_var\_args\_call(arg1, arg2, arg3):
5. **print** "arg1:", arg1
6. **print** "arg2:", arg2
7. **print** "arg3:", arg3
9. kwargs = {"arg3": 3, "arg2": "two"} # dictionary
11. fun\_var\_args\_call(1, \*\*kwargs)

result：

1. arg1: 1
2. arg2:"two"
3. arg3:3