**Python**

打包和解包：\*以及\*\*在形参中表示打包，可以在调用时传入多个参数，函数内部使用一个list或dict来处理。在实参中表示解包，将一个list或dict分成多个在函数中处理。

Python 赋值过程中不明确区分拷贝和引用，一般对静态变量的传递为拷贝，对动态变量的传递为引用。（注，对静态变量首次传递时也是引用，当需要修改静态变量时，因为静态变量不能改变，所以需要生成一个新的空间存储数据）。

* 字符串，数值，元组均为静态变量
* 列表，字典为动态变量。

变量有时比较复杂，存在组合现象，比如字典中包含列表，列表中包含字典，但赋值时，总是属于某个类型。如果实在不清楚状况，可以试验一下，用id()这个函数看看，如果是引用，两个变量指向的地址是相同的。

**Pytorch**

in-place operation：以tensor.add\_(x)的附加下划线形式，表示直接修改该tensor而不是进行复制。

Tensor变量的requires\_grad的属性默认为False,若一个节点requires\_grad被设置为True，那么所有依赖它的节点的requires\_grad都为True。

is\_leaf表示没有被操作的tensor

All Tensors that have [requires\_grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.requires_grad) which is False will be leaf Tensors by convention.

For Tensors that have [requires\_grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.requires_grad) which is True, they will be leaf Tensors if they were created by the user. This means that they are not the result of an operation and so grad\_fn is None.

Only leaf Tensors will have their [grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.grad) populated during a call to [backward()](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.backward). To get [grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.grad) populated for non-leaf Tensors, you can use [retain\_grad()](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.retain_grad).

不需要梯度的张量是叶子张量。（普通情况以及由多个叶子张量构成的张量，不需要梯度）

任何张量都需要考虑几个属性：数据域，计算图，梯度