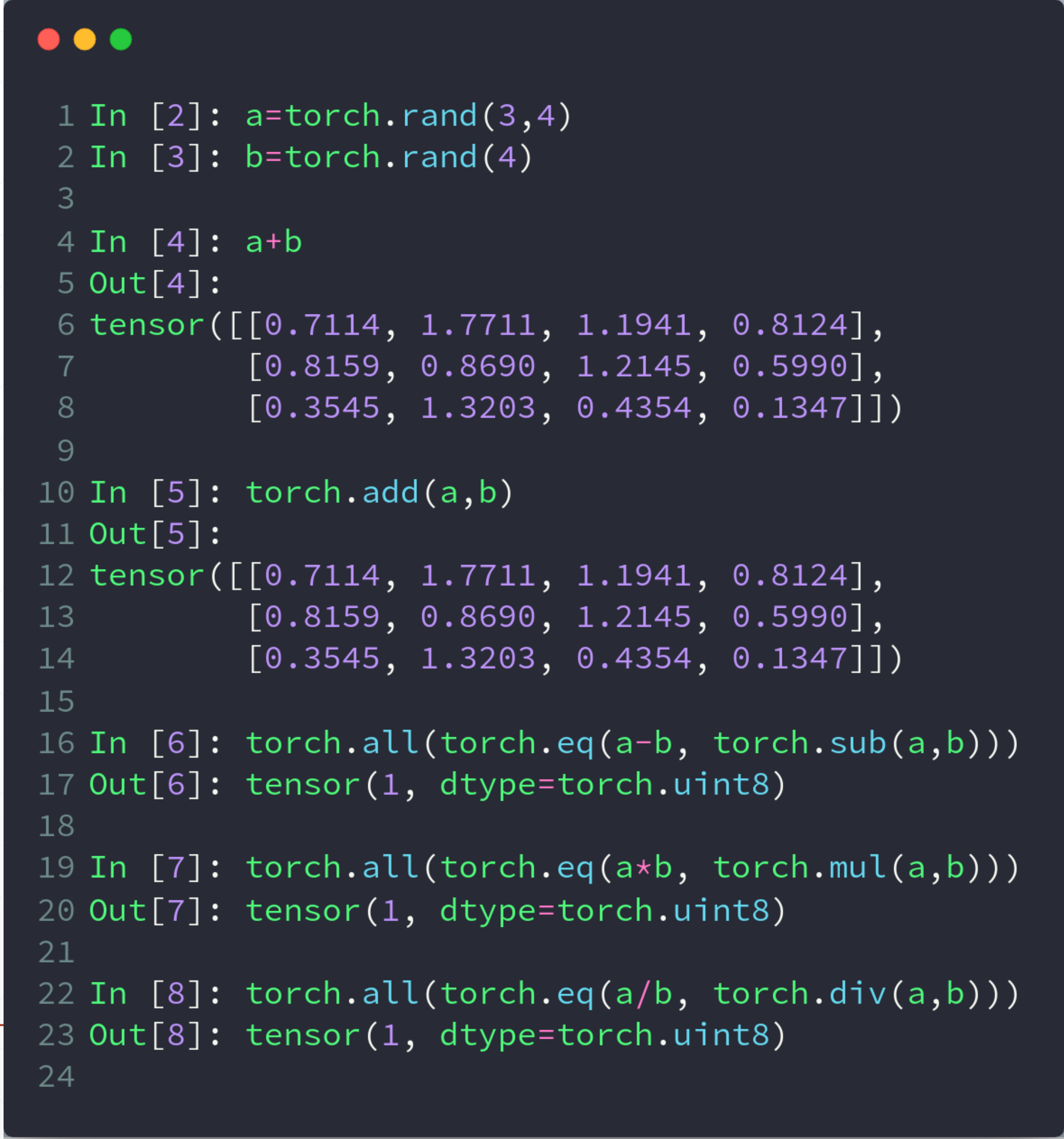
Math operation

- Add/minus/multiply/divide
- Matmul
- Pow
- Sqrt/rsqrt
- Round

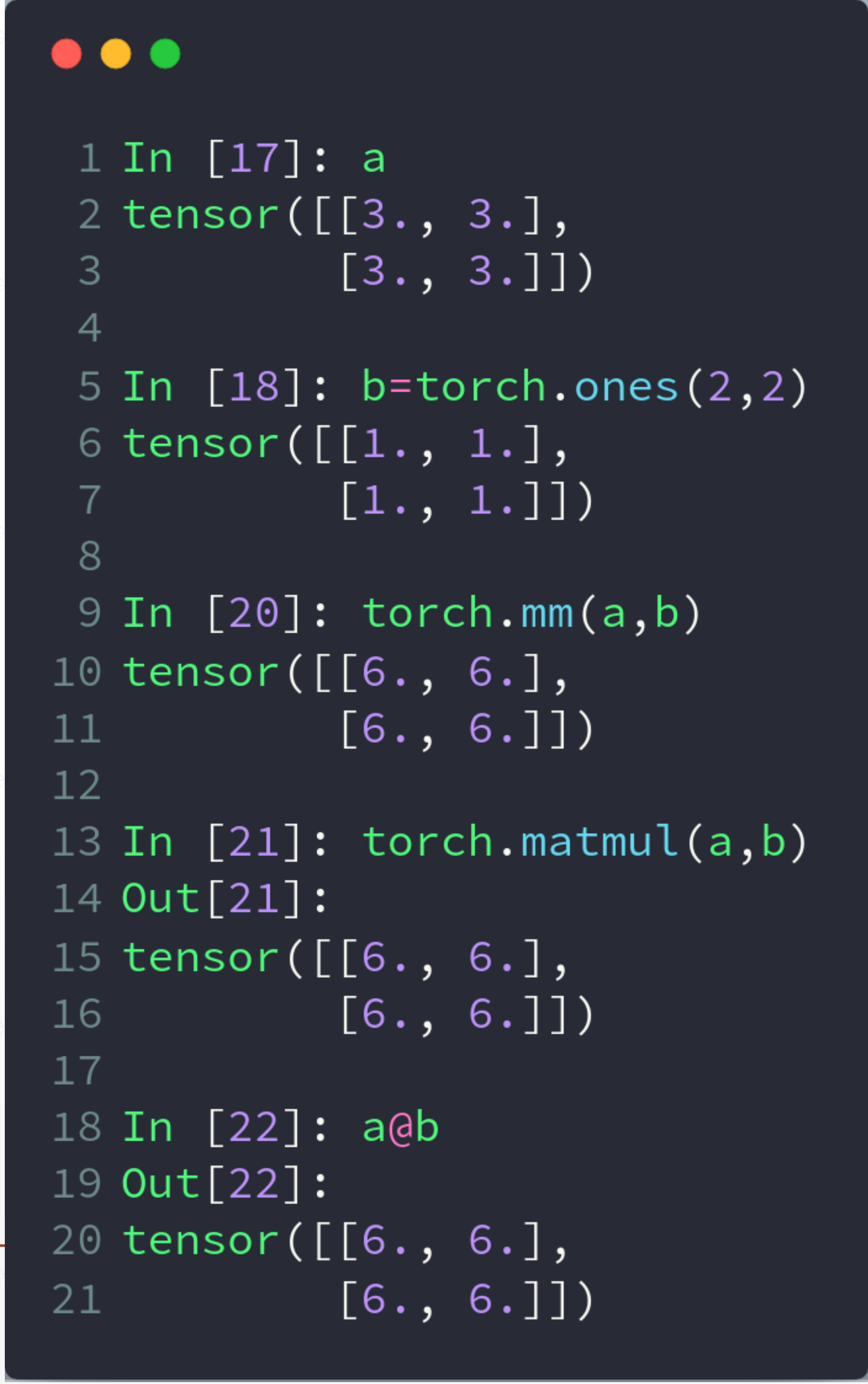
basic



```
In [2]: a=torch.rand(3,4)
In [3]: b=torch.rand(4)

In [4]: a+b
Out[4]:
tensor([[0.7114, 1.7711, 1.1941, 0.8124],
        [0.8159, 0.8690, 1.2145, 0.5990],
        [0.3545, 1.3203, 0.4354, 0.1347]])

In [5]: torch.add(a,b)
Out[5]:
tensor([[0.7114, 1.7711, 1.1941, 0.8124],
        [0.8159, 0.8690, 1.2145, 0.5990],
        [0.3545, 1.3203, 0.4354, 0.1347]])

In [6]: torch.all(torch.eq(a-b, torch.sub(a,b)))
Out[6]: tensor(1, dtype=torch.uint8)

In [7]: torch.all(torch.eq(a*b, torch.mul(a,b)))
Out[7]: tensor(1, dtype=torch.uint8)

In [8]: torch.all(torch.eq(a/b, torch.div(a,b)))
Out[8]: tensor(1, dtype=torch.uint8)

```

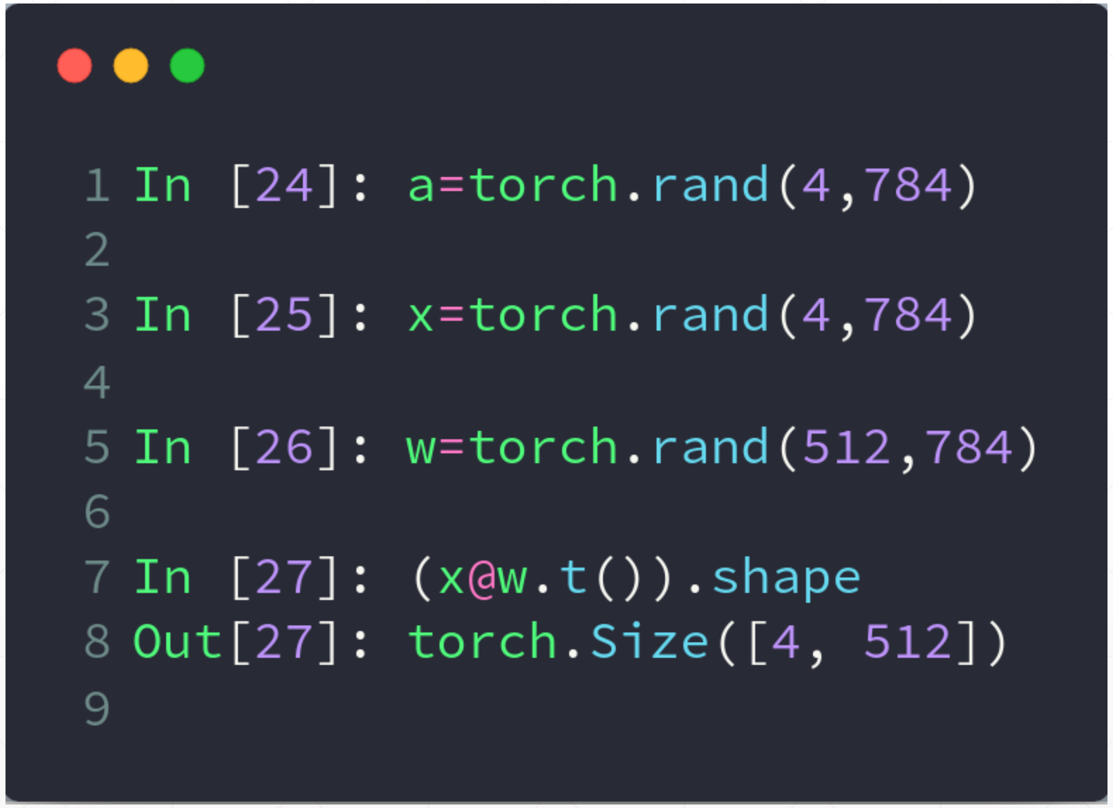
matmul

- Torch.mm
  - only for 2d
- Torch.matmul
- @



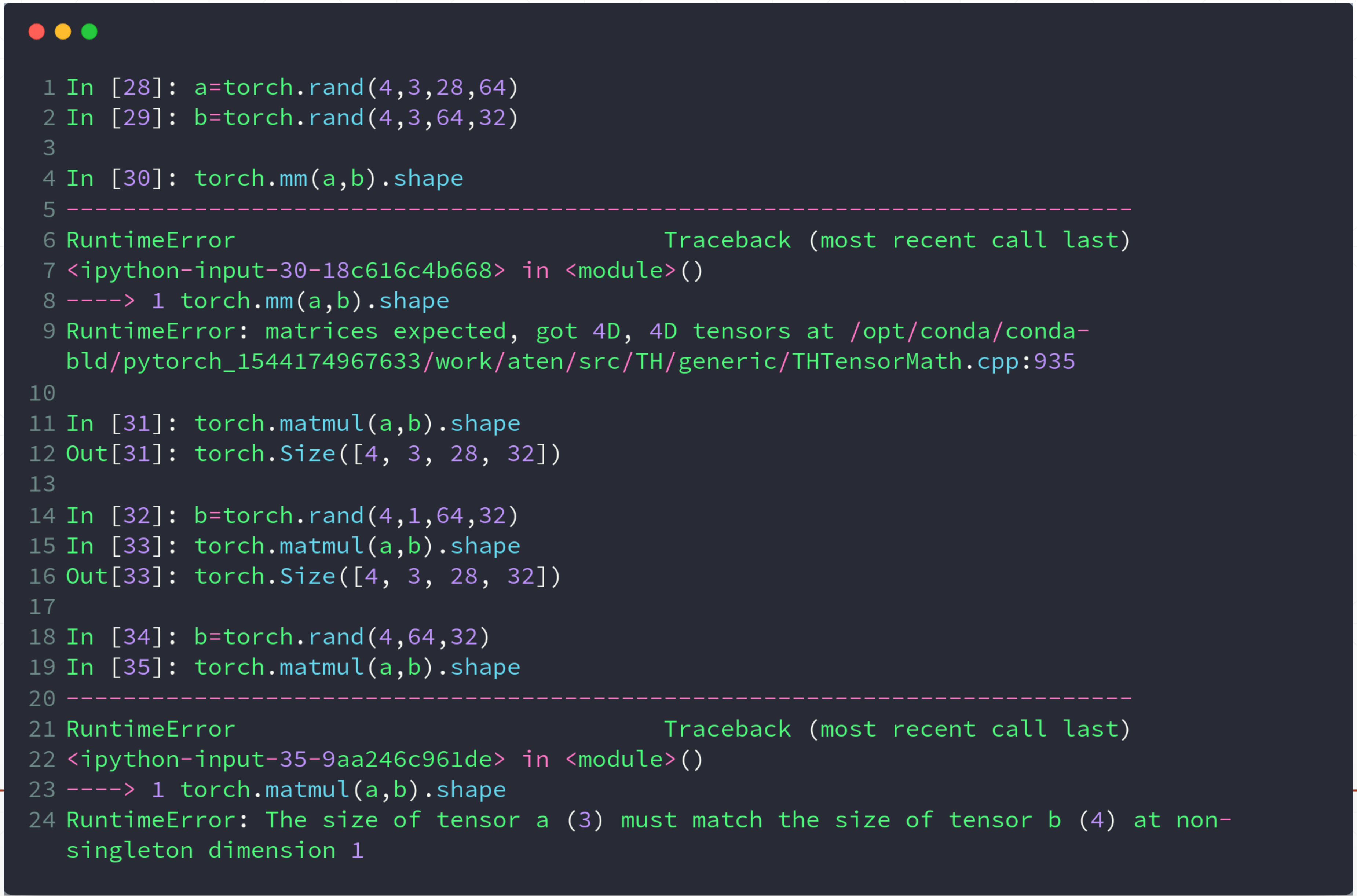
```
In [17]: a
tensor([[3., 3.],
        [3., 3.]])

In [18]: b=torch.ones(2,2)
tensor([[1., 1.],
        [1., 1.]])

In [20]: torch.mm(a,b)
tensor([[6., 6.],
        [6., 6.]])

In [21]: torch.matmul(a,b)
Out[21]:
tensor([[6., 6.],
        [6., 6.]])

In [22]: a@b
Out[22]:
tensor([[6., 6.],
        [6., 6.]])
```

An example



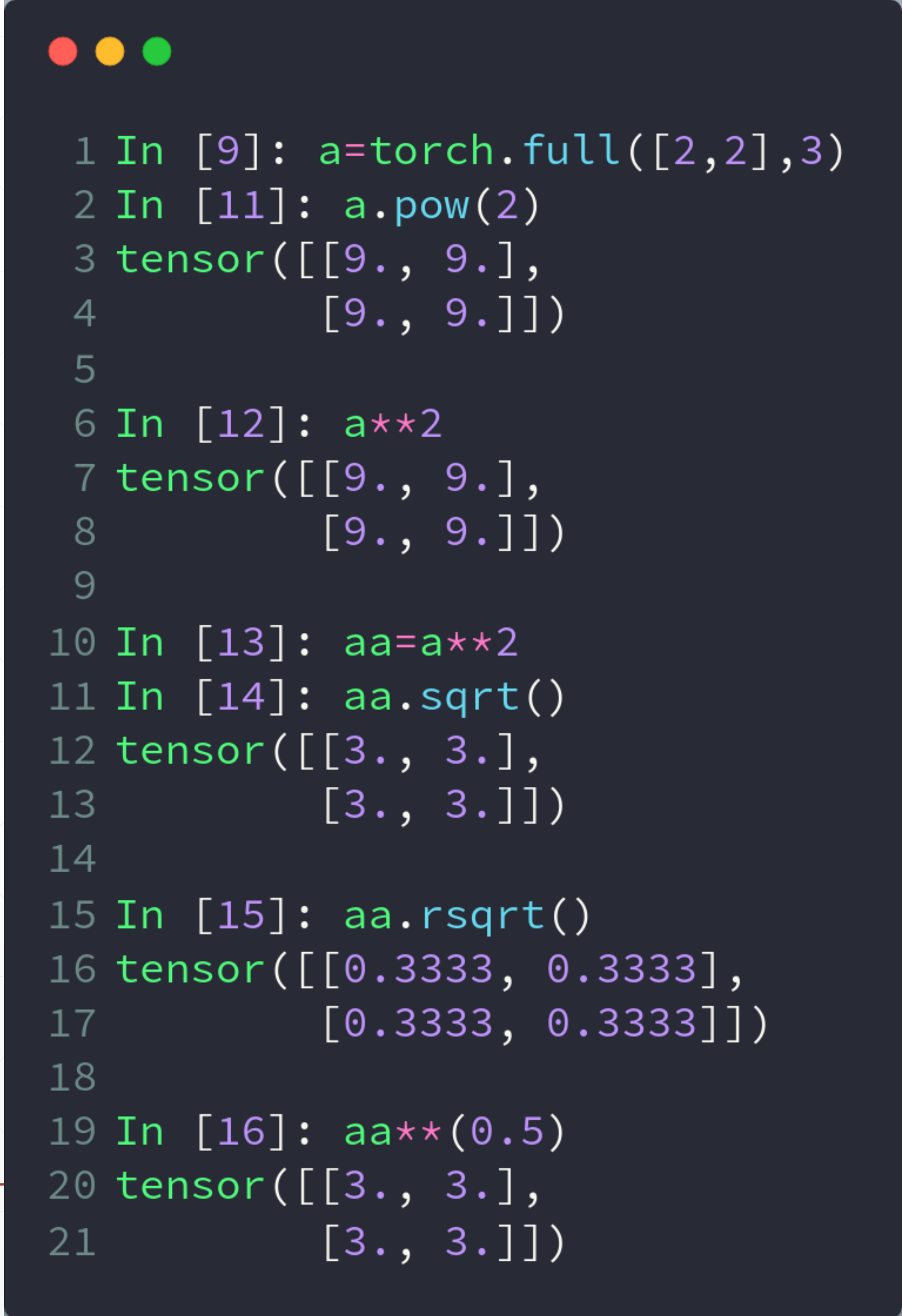
```
In [24]: a=torch.rand(4,784)

In [25]: x=torch.rand(4,784)

In [26]: w=torch.rand(512,784)

In [27]: (x@w.t()).shape
Out[27]: torch.Size([4, 512])
```

>2d tensor matmul?



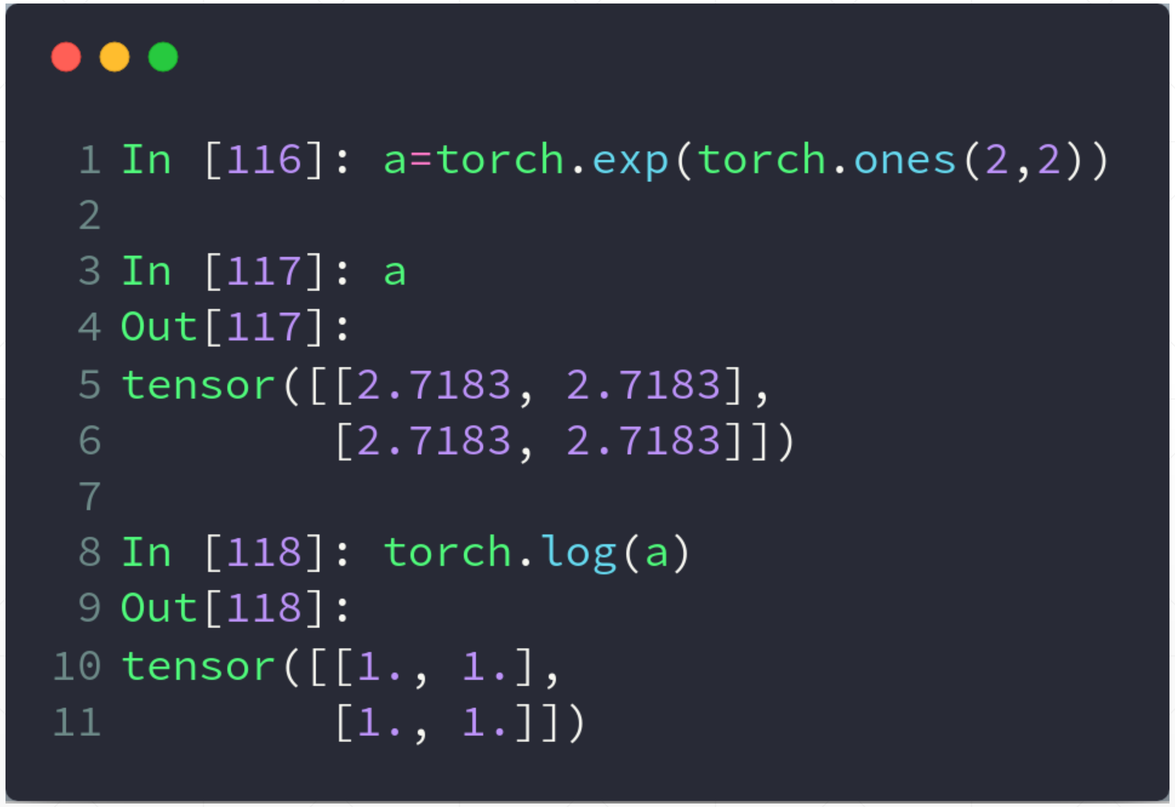
```
In [28]: a=torch.rand(4,3,28,64)
In [29]: b=torch.rand(4,3,64,32)

In [30]: torch.mm(a,b).shape
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-30-18c616c4b668> in <module>()
----> 1 torch.mm(a,b).shape
RuntimeError: matrices expected, got 4D, 4D tensors at /opt/conda/conda-
bld/pytorch_1544174967633/work/aten/src/TH/generic/THTensorMath.cpp:935

In [31]: torch.matmul(a,b).shape
Out[31]: torch.Size([4, 3, 28, 32])

In [32]: b=torch.rand(4,1,64,32)
In [33]: torch.matmul(a,b).shape
Out[33]: torch.Size([4, 3, 28, 32])

In [34]: b=torch.rand(4,64,32)
In [35]: torch.matmul(a,b).shape
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-35-9aa246c961de> in <module>()
----> 1 torch.matmul(a,b).shape
RuntimeError: The size of tensor a (3) must match the size of tensor b (4) at non-
singleton dimension 1
```

Power



```
In [9]: a=torch.full([2,2],3)
In [11]: a.pow(2)
tensor([[9., 9.],
        [9., 9.]])

In [12]: a**2
tensor([[9., 9.],
        [9., 9.]])

In [13]: aa=a**2
In [14]: aa.sqrt()
tensor([[3., 3.],
        [3., 3.]])

In [15]: aa.rsqrt()
tensor([[0.3333, 0.3333],
        [0.3333, 0.3333]])

In [16]: aa**(0.5)
tensor([[3., 3.],
        [3., 3.]])
```

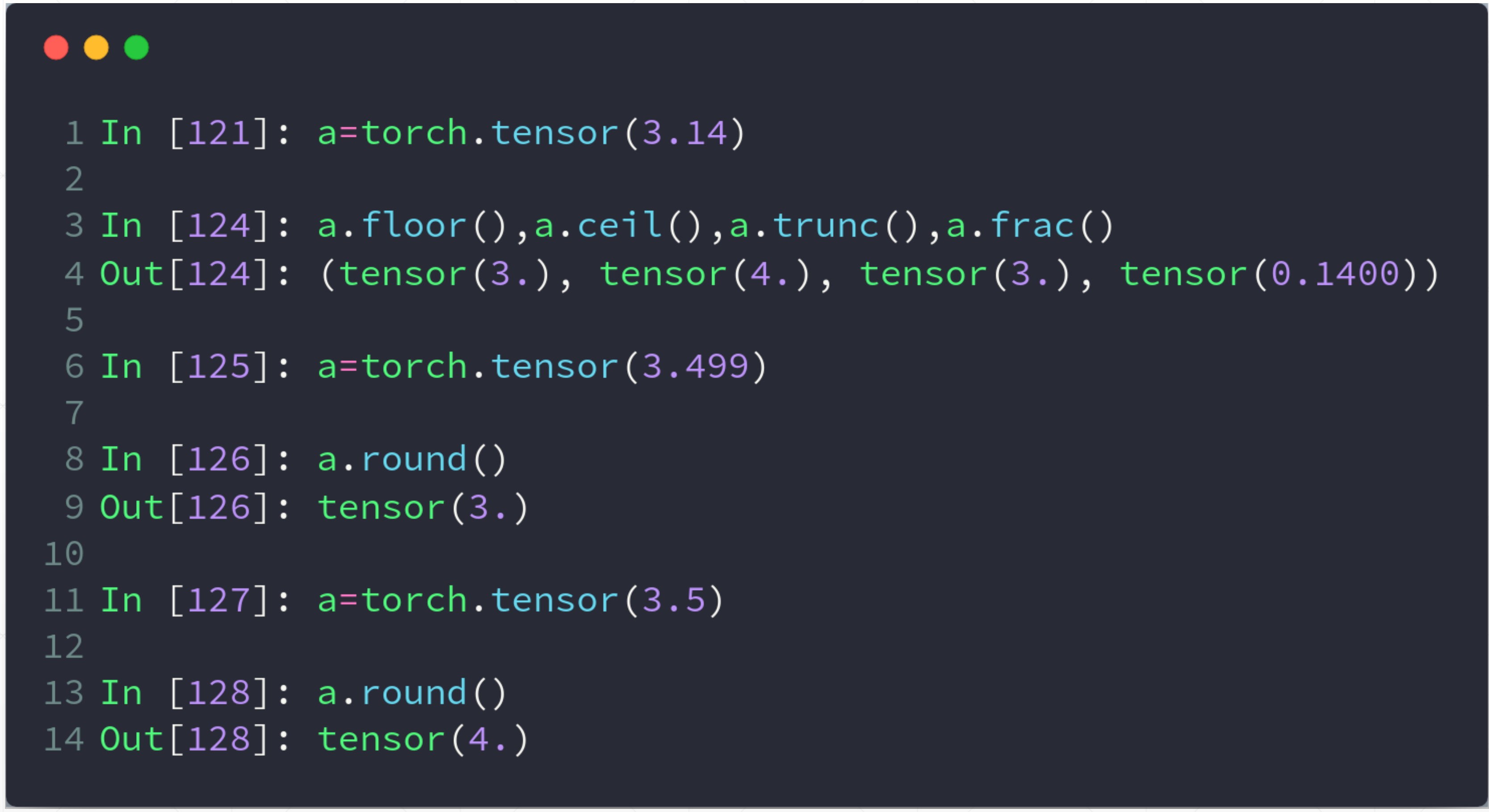
Exp log



```
In [116]: a=torch.exp(torch.ones(2,2))

In [117]: a
Out[117]:
tensor([[2.7183, 2.7183],
        [2.7183, 2.7183]])

In [118]: torch.log(a)
Out[118]:
tensor([[1., 1.],
        [1., 1.]])
```

Approximation

- .floor() .ceil()
- .round()
- .trunc() .frac()

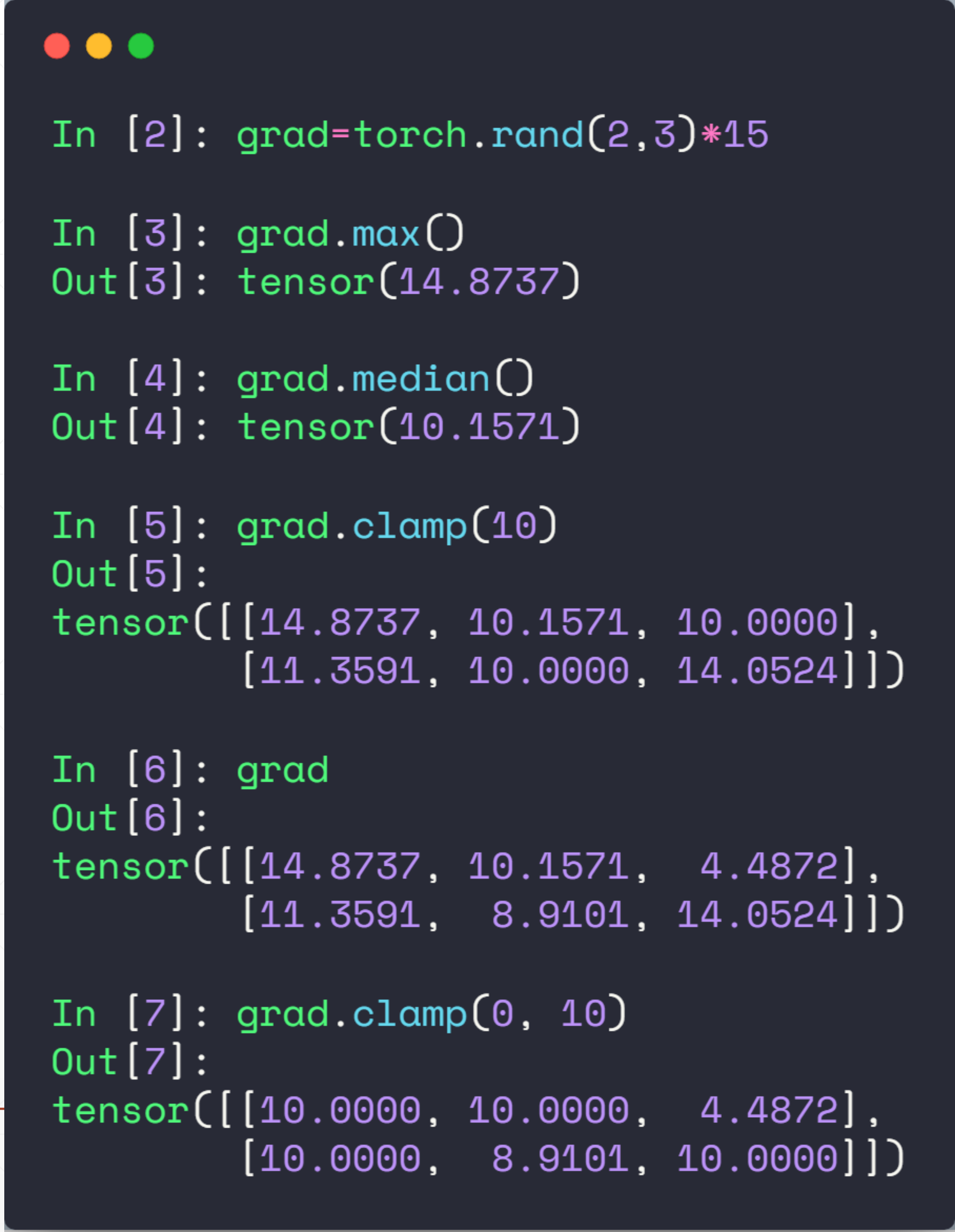


```
In [121]: a=torch.tensor(3.14)

In [124]: a.floor(),a.ceil(),a.trunc(),a.frac()
Out[124]: (tensor(3.), tensor(4.), tensor(3.), tensor(0.1400))

In [125]: a=torch.tensor(3.499)

In [126]: a.round()
Out[126]: tensor(3.)

In [127]: a=torch.tensor(3.5)

In [128]: a.round()
Out[128]: tensor(4.)
```

clamp

- gradient clipping

- (min)
- (min, max)



```
In [2]: grad=torch.rand(2,3)*15

In [3]: grad.max()
Out[3]: tensor(14.8737)

In [4]: grad.median()
Out[4]: tensor(10.1571)

In [5]: grad.clamp(10)
Out[5]:
tensor([[14.8737, 10.1571, 10.0000],
        [11.3591, 10.0000, 14.0524]])

In [6]: grad
Out[6]:
tensor([[14.8737, 10.1571,  4.4872],
        [11.3591,  8.9101, 14.0524]])

In [7]: grad.clamp(0, 10)
Out[7]:
tensor([[10.0000, 10.0000,  4.4872],
        [10.0000,  8.9101, 10.0000]])
```