

# AI作业 2

DDL: 2022-12-01前

## 任务0：继续学习python基础语法，并完成作业

python 语法参考资料: <https://www.liaoxuefeng.com/wiki/1016959663602400>

### 1. 学习 task3:AI 训练营-龙珠计划 python 训练营

链接: <https://tianchi.aliyun.com/>



### 2. 生成如下序列（必须使用列表生成式）

- (1) 10,20,30,40,50,60
- (2) 1,4,9,16,25,36,49
- (3) 10,-20,30,-40,50,60
- (4) 10,30,50（要求在 1 的基础上进行过滤）
- (5) 打印乘法口诀表（一行完成）

### 3. 完成计时器对象，完成如下定义

提示：该对象的实例有一个列表，start 开始记录时间,每次 stop 将 (现在的时间-start 的时间) 添加在列表中并将间隔的时间返回

```
class Timer :
    """ 记录多次运行时间。"""
    def __init__ ( self ):
    """ 启动计时器。"""
    def start ( self ):
    """ 停止计时器并将时间记录在列表中并返回"""
    def stop ( self ):
    """ 返回平均时间。"""
    def avg ( self ):
    """ 返回时间总和。"""
    def sum ( self ):
    """ 返回累计时间。"""
    def cumsum ( self ):
```

## 结果示例

```
timer.start()
d = a + b
f'{timer.stop():.5f} sec'

"""结果输出"""
'0.00020 sec'
```

## 4. 完成累加器 Accumulator 类

Accumulator 是一个实用程序类，用于对多个变量进行累加; 提示：例如定义 n=2 的累加器 a，其会定义一个大小为 2 的列表 [0,0]; 调用 a.add(2,3) 之后其列表的值对应累加为 [2,3]; a[idx] 则返回列表对应的索引的值; reset 重置这个列表为全零 [0,0]

```
class Accumulator :
    """ 在`n`个变量上累加。"""
    def __init__ ( self , n):
    def add ( self , * args ):
    def reset ( self ):
    def __getitem__ ( self , idx ):
```

## 5. 完成一个斐波那契数生成函数 (生成器)

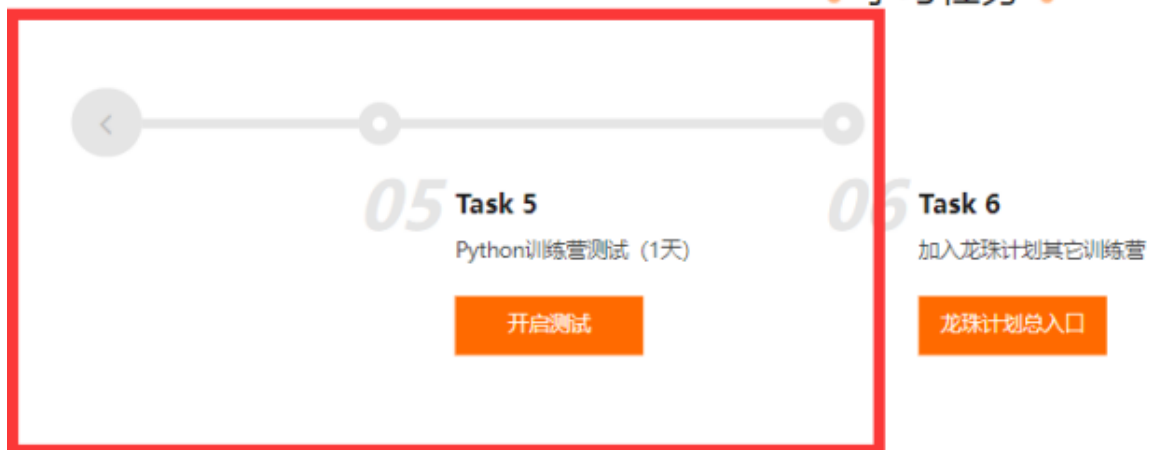
`f = fib()` 可以不断调用 `next(f)` 并不断返回下一个值

```
def fib():  
    ...  
    yield
```

## 任务1：完成测试

同任务一中龙珠计划 python 训练营链接，完成 Task5 测试。

学习任务



## 任务2：深度学习入门

### 1. 观看视频 00-18 并理解

结合相应书籍学习

参考书籍：《动手学深度学习》—动手学深度学习 2.0.0-alpha2 documentation (d2l.ai)

书籍 url: <http://zh.d2l.ai/>

参考学习视频：跟李沐学 AI 的个人空间 哔哩哔哩 bilibili

视频 url: <https://space.bilibili.com/1567748478/channel/seriesdetail?sid=358497>

可配合Pytorch入门一起学习

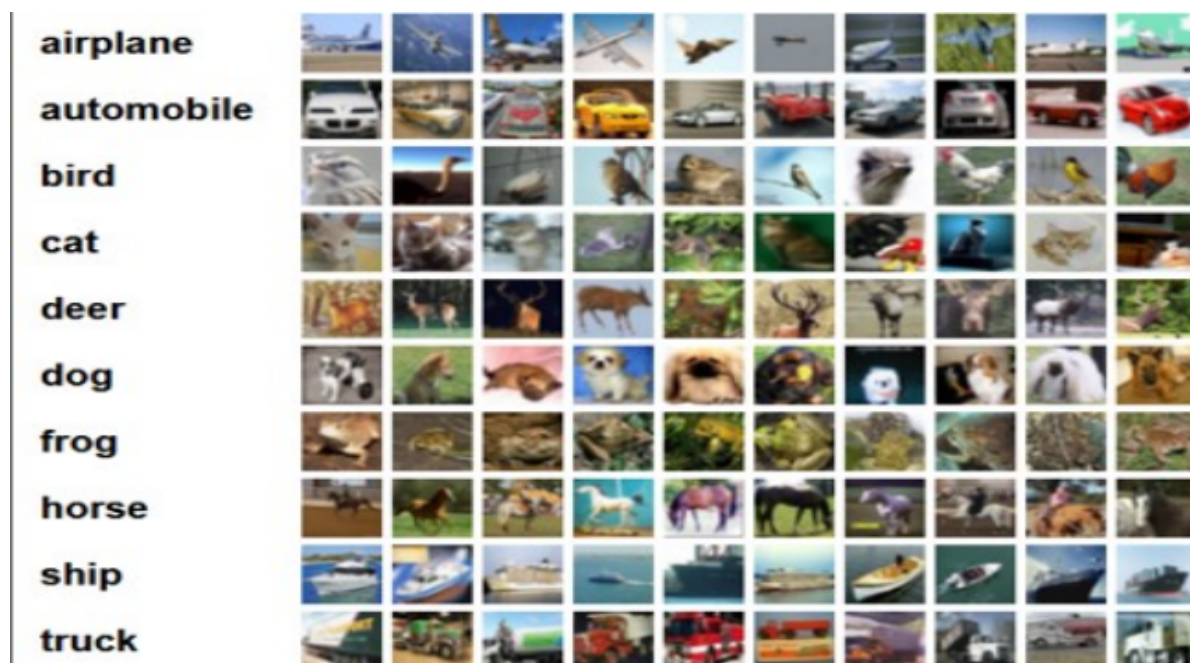
中文 url: [60分钟快速入门 PyTorch - 知乎 \(zhihu.com\)](https://www.zhihu.com/question/40461124/answer/104411124)

## 2. 对给定的图像数据集进行分类 (需要往下看视频)

### 数据集 1: MNIST数据集



### 数据集 2: CIFAR10



**不限定分类的方法，精度不作要求，主要体验过程 (选做不纳入考核)**

**提示：使用 00-18 的视频中的方法能够完成，但精度可能不高（可学习完卷积层再做）；在本次任务截止时间前，学有余力的同学可以顺着视频继续往下学，尝试其他方法以获取更高精度。**