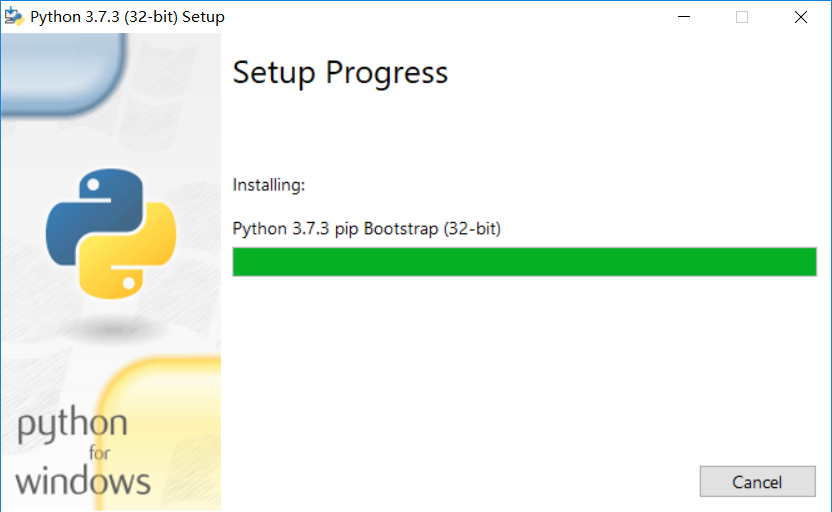
# 实训第一次报告

——16340312周泽昊

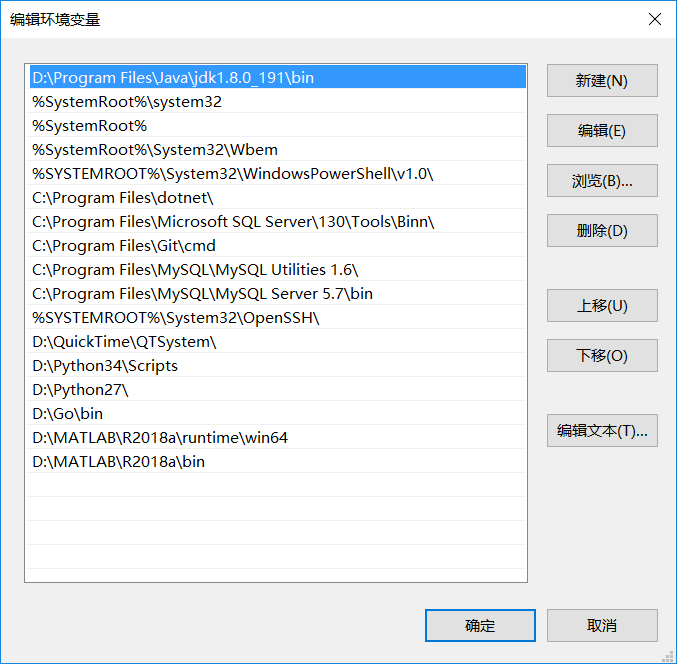
1. **配置python环境**

下载安装python3

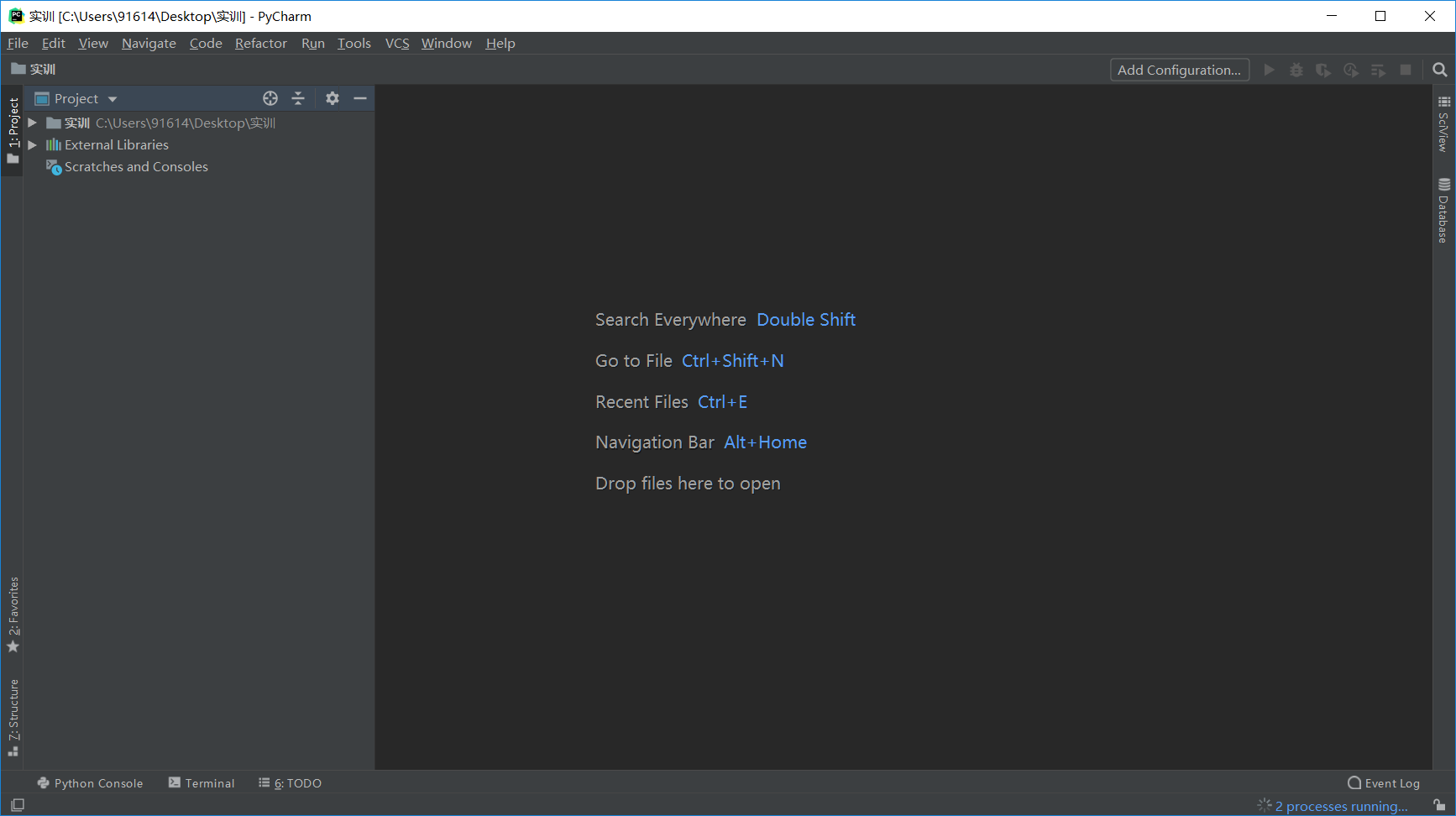




环境变量



下载安装IDE：PyCharm



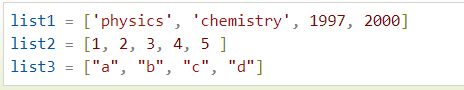
可以开始了

1. **简略学习python语言**

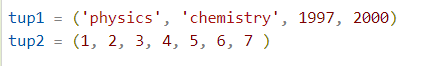
之前没有用过，所以了解一下python的基本操作

常用数据类型如下

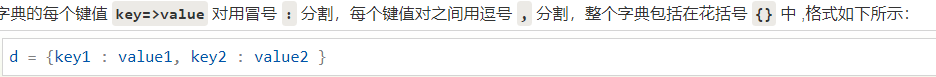
列表：



元祖：

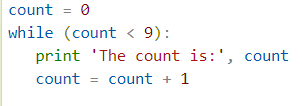


字典：

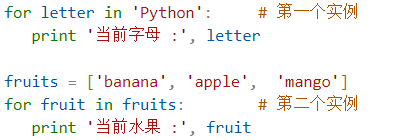


循环语句与C++等都类似

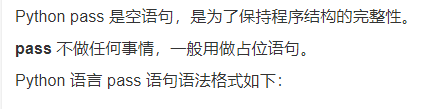
While：



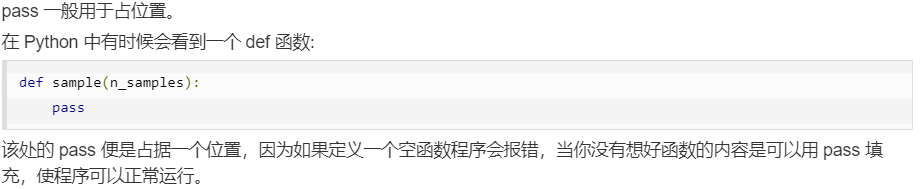
For：



不太一样的东西：pass



一开始名明白这个关键字的意义，直到看到



1. **学习策略和API**

**小市值策略**：

筛选出市值介于20-30亿的股票，选取其中市值最小的三只股票，

每天开盘买入，持有五个交易日，然后调仓。

**双线策略**：

双均线策略，当五日均线位于十日均线上方则买入，反之卖出。

## 初始化函数，设定要操作的股票、基准等等  
def initialize(context):  
 # 定义一个全局变量, 保存要操作的股票  
 # 要操作的股票为000002(股票:万科A)  
 g.security = '000002.XSHE'  
 # 设定沪深300作为基准  
 set\_benchmark('000300.XSHG')  
 # True为开启动态复权模式，使用真实价格交易  
 set\_option('use\_real\_price', True)   
 # 设定成交量比例  
 set\_option('order\_volume\_ratio', 1)  
 # 股票类交易手续费是：买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税, 每笔交易佣金最低扣5块钱  
 set\_order\_cost(OrderCost(open\_tax=0, close\_tax=0.001, \  
 open\_commission=0.0003, close\_commission=0.0003,\  
 close\_today\_commission=0, min\_commission=5), type='stock')  
 # 运行函数  
 run\_daily(trade, 'every\_bar')  
  
## 交易程序  
def trade(context):  
 security = g.security  
 # 设定均线窗口长度  
 n1 = 5  
 n2 = 10  
 # 获取股票的收盘价  
 close\_data = attribute\_history(security, n2+2, '1d', ['close'],df=False)  
 # 取得过去 ma\_n1 天的平均价格  
 ma\_n1 = close\_data['close'][-n1:].mean()  
 # 取得过去 ma\_n2 天的平均价格  
 ma\_n2 = close\_data['close'][-n2:].mean()  
 # 取得当前的现金  
 cash = context.portfolio.cash  
  
 # 如果当前有余额，并且n1日均线大于n2日均线  
 if ma\_n1 > ma\_n2:  
 # 用所有 cash 买入股票  
 order\_value(security, cash)  
 # 记录这次买入  
 log.info("Buying %s" % (security))  
  
 # 如果n1日均线小于n2日均线，并且目前有头寸  
 elif ma\_n1 < ma\_n2 and context.portfolio.positions[security].closeable\_amount > 0:  
 # 全部卖出  
 order\_target(security, 0)  
 # 记录这次卖出  
 log.info("Selling %s" % (security))  
  
 # 绘制n1日均线价格  
 record(ma\_n1=ma\_n1)  
 # 绘制n2日均线价格  
 record(ma\_n2=ma\_n2)

**银行轮滚动策略**：

始终持有沪深300银行指数成分股中市净率最低的股份制银行，每周检查一次，

如果发现有新的股份制银行市净率低于原有的股票，则予以换仓。

## 初始化函数，设定要操作的股票、基准等等  
def initialize(context):  
 # 设定沪深300银行指数作为基准  
 set\_benchmark('399951.XSHE')  
 # True为开启动态复权模式，使用真实价格交易  
 set\_option('use\_real\_price', True)  
 # 设定成交量比例  
 set\_option('order\_volume\_ratio', 1)  
 # 股票类交易手续费是：买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税, 每笔交易佣金最低扣5块钱  
 set\_order\_cost(OrderCost(open\_tax=0, close\_tax=0.001, \  
 open\_commission=0.0003, close\_commission=0.0003, \  
 close\_today\_commission=0, min\_commission=5), type='stock')  
 # 运行函数, 按周运行，在每周第一个交易日运行  
 run\_weekly(chenk\_stocks, weekday=1, time='before\_open') # 选股  
 run\_weekly(trade, weekday=1, time='open') # 交易  
  
  
## 得到沪深300银行指数成分股,找到市净率最低的股票  
def chenk\_stocks(context):  
 # 得到沪深300银行指数成分股  
 g.stocks = get\_index\_stocks('399951.XSHE')  
  
 # 查询股票的市净率，并按照市净率升序排序  
 if len(g.stocks) > 0:  
 g.df = get\_fundamentals(  
 query(  
 valuation.code,  
 valuation.pb\_ratio  
 ).filter(  
 valuation.code.in\_(g.stocks)  
 ).order\_by(  
 valuation.pb\_ratio.asc()  
 )  
 )  
  
 # 找出最低市净率的一只股票  
 g.code = g.df['code'][0]  
  
  
## 交易  
def trade(context):  
 if len(g.stocks) > 0:  
 code = g.code  
 # 如持仓股票不是最低市净率的股票，则卖出  
 for stock in context.portfolio.positions.keys():  
 if stock != code:  
 order\_target(stock, 0)  
  
 # 持仓该股票  
 if len(context.portfolio.positions) > 0:  
 return  
 else:  
 order\_value(code, context.portfolio.cash)

**低估价值选股策略：**

1.市净率小于2；

2.负债比例高于市场平均值;

3.企业的流动资产至少是流动负债的1.2倍;

4.每年四次调仓，即在1/4/7/10月调仓；

5.可加入止损(十天HS300跌幅达10%清仓)；