这是什么

这是一个简单的python脚本，用于分析python代码的基本情况，包括代码行数、注释行数、空白行数、有效代码行数、引用包情况、缩进层级、循环、条件结构计数以及变量名统计等功能。

如何使用

直接使用`python python_parser.py FILE` 即可使用。

代码结构

将所有功能集成到唯一一个函数`code_analysis`中，输入是待分析的python文件，输出是一系列的代码信息。

- 总行数统计的功能使用了`readlines()` 函数进行统计。
- 单行注释行数统计的功能使用了关键词`#`进行标识统计。
- 空白行数统计的功能使用了空白字符串检测实现。
- 有效代码行统计的功能使用了总行数减空白行数实现。
- 引用包统计的功能使用了字符串替换与切分的方式实现。
- `for, if ,while, try`等关键词统计的功能使用字符串检测的方式实现。
- 变量名统计使用了去除按行读取单词并用关键词过滤得到，并最总以哈希表的方法清洗重复的变量。

运行实例

对`testfile.py`文件进行测试。该文件内容如下：

```
import os
import re
import shutil
import pandas as pd
import requests
import lxml
import glob
import tarfile
import time
from bs4 import BeautifulSoup

url_base = 'https://arxiv.org/e-print/'
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.54'
}
DOWNLOAD_DIR = 'Downloads/'

def download_tar(file_name, dir):
    print("\nDownloading: \033[1;31;40m{}.tar.gz\033[0m".format(file_name))
    url = url_base + file_name
    file_name += '.tar.gz'
    file_path = os.path.join(dir, file_name)
    if os.path.exists(file_path):
        print("File \033[1;31;40m{}\033[0m Already Exisist.\n\033[1;36;40mSize :
{} Bytes\033[0m.".format(file_name,os.path.getsize(file_path)))
        return file_path
    else:
        req = requests.get(url,headers=headers)
        if req.status_code == 200:
            with open(file_path, 'wb') as f:
                f.write(req.content)
            print('Finish Download: {}. \n\033[1;36;40m\033[1;36;40mSize : {}
Bytes\033[0m.'.format(file_name, os.path.getsize(file_path)))
            return file_path
        else:
            print('\033[1;36;40mDownload {} Failed.\033[0m'.format(file_name))
            return None

def un_tar(file):
    tar = tarfile.open(file)
    names = tar.getnames()
    if os.path.isdir(file + '_files'):
        pass
    else:
        os.mkdir(file + '_files')
        for name in names:
            tar.extract(name, file + '_files')
    # print("Unzip File Finished.")
    return file + '_files'

def find_texs(untar_file):
    texs = glob.glob('{}/*.tex'.format(untar_file))
    return texs

def parse_tex(file):
    rule = re.compile(r'\\begin{equation}(.*?)\\end{equation}',re.S)
    with open(file, 'rb') as f:
        content = f.read().decode('utf-8')
        content = content.replace('\n','')
        results = rule.findall(content)
    return results

def parse_texs(files):
    results = []
    rule = re.compile(r'\\begin{equation}(.*?)\\end{equation}',re.S)
    for file in files:
        with open(file, 'rb') as f:
            content = f.read().decode('utf-8')
            content = content.replace('\n','')
            result = rule.findall(content)
        results += result
    print("\033[1;32;40mTotal {} Formula Found.\033[0m".format(len(results)))
    return {index:value for index, value in enumerate(results)}

def clean_files(tar_file, folder):
    shutil.rmtree(folder)
    os.remove(tar_file)
```

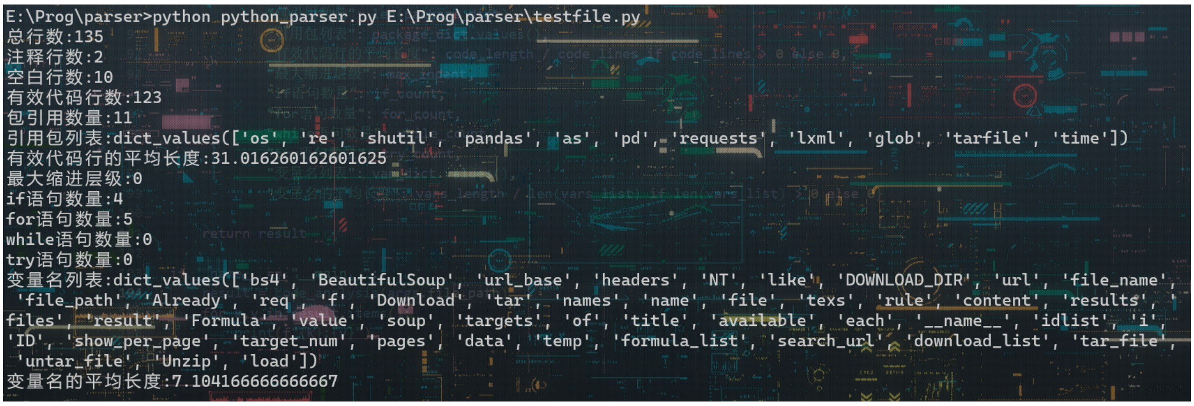
```
def get_paper_num(url):
    req = requests.get(url, headers=headers)
    soup = BeautifulSoup(req.text, 'xml')
    targets = soup.findAll('small')
    return int(re.findall('total of (.*?) entries',targets[0].text)[0])

def get_index_per_page(search_url):
    req = requests.get(search_url,headers=headers)
    soup = BeautifulSoup(req.text, 'xml')
    results = soup.find_all('a', title = 'Other formats' )
    print("Find {} available files.".format(len(results)))
    return [each.get('href')[8:] for each in results]

if __name__ == '__main__':
    idlist = [i for i in range(2001,2013)]
    for ID in idlist:
        url = 'https://arxiv.org/list/cs/{}'.format(ID)
        show_per_page = 50
        target_num = get_paper_num(url)
        pages = target_num//show_per_page
        data = pd.DataFrame()
        # data = pd.read_csv('Outputs/{}.csv'.format(ID), encoding='utf-8')
        for i in range(pages):
            temp = pd.DataFrame()
            formula_list = []
            search_url = 'https://arxiv.org/list/cs/{}?skip={}&show=
{}'.format(ID, i * show_per_page,show_per_page)
            download_list = get_index_per_page(search_url)
            temp['paper_id'] = download_list
            for each in download_list:
                try:
                    tar_file = download_tar(each, DOWNLOAD_DIR)
                except:
                    print("\033[1;31;40mCDownload Failed.\033[0m")
                    formula_list.append("Download Failed.")
                    continue
                try:
                    untar_file = un_tar(tar_file)
                except:
                    print("\033[1;31;40mCould not Unzip .tar file.\033[0m")
                    formula_list.append("Unzip Failed.")
                    continue
                try:
                    texs = find_texs(untar_file)
                except:
                    print("\033[1;31;40mCounld not load texes.\033[0m")
                    formula_list.append("Counld not load texes")
                    continue
                try:
                    content = str(parse_texs(texs))
                except:
                    print("\033[1;31;40mParsing Failed.\033[0m")
                    formula_list.append("Parsing Failed.")
                    continue
```

```python
                formula_list.append(content)
                clean_files(tar_file, untar_file)
            temp['formula_json'] = formula_list
            data = pd.concat([data, temp], axis=0)
            data.to_csv("Outputs/{}.csv".format(ID),index=False)
        print("Task {} Finished.".format(ID))
```

运行结果如图所示:



```
E:\Prog\parser>python python_parser.py E:\Prog\parser\testfile.py
总行数:135
注释行数:2
空白行数:10
有效代码行数:123
包引用数量:11
引用包列表:dict_values(['os', 're', 'shutil', 'pandas', 'as', 'pd', 'requests', 'lxml', 'glob', 'tarfile', 'time'])
有效代码行的平均长度:31.016260162601625
最大缩进层级:0
if语句数量:4
for语句数量:5
while语句数量:0
try语句数量:0
变量名列表:dict_values(['bs4', 'BeautifulSoup', 'url_base', 'headers', 'NT', 'like', 'DOWNLOAD_DIR', 'url', 'file_name',
 'file_path', 'Already', 'req', 'f', 'Download', 'tar', 'names', 'name', 'file', 'texs', 'rule', 'content', 'results', '
files', 'result', 'Formula', 'value', 'soup', 'targets', 'of', 'title', 'available', 'each', '__name__', 'idlist', 'i',
'ID', 'show_per_page', 'target_num', 'pages', 'data', 'temp', 'formula_list', 'search_url', 'download_list', 'tar_file',
 'untar_file', 'Unzip', 'load'])
变量名的平均长度:7.104166666666667
```