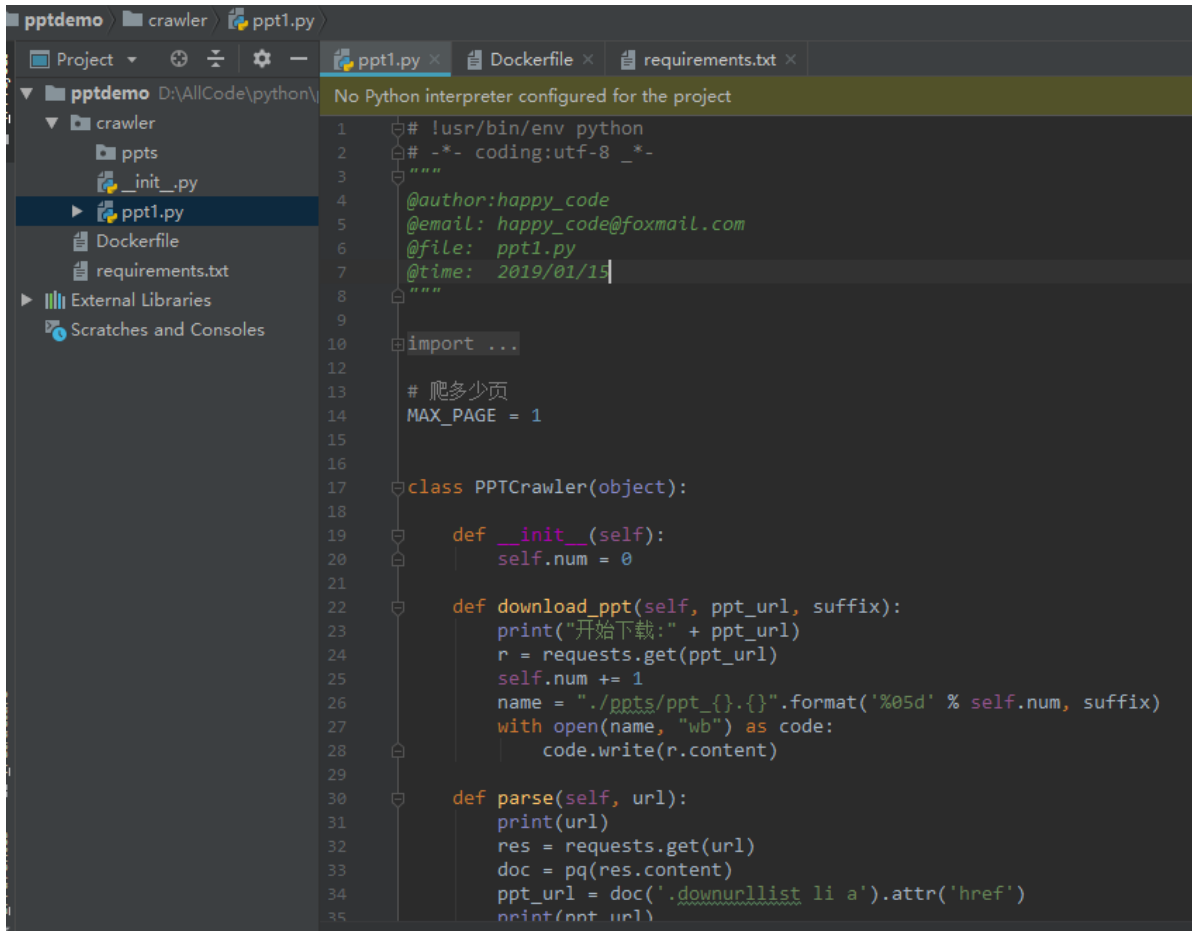


# Docker的基本使用(部署python项目)

今天开始利用docker来部署项目，当然，首先，需要安装好Docker，这个在之前的PPT讲过。

## 一、准备项目

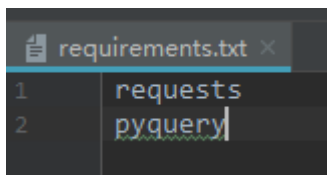


```
1 #!/usr/bin/env python
2 # -*- coding:utf-8 -*-
3 """
4 @author: happy_code
5 @email: happy_code@foxmail.com
6 @file: ppt1.py
7 @time: 2019/01/15
8 """
9
10 import ...
11
12 # 爬多少页
13 MAX_PAGE = 1
14
15
16
17 class PPTCrawler(object):
18
19     def __init__(self):
20         self.num = 0
21
22     def download_ppt(self, ppt_url, suffix):
23         print("开始下载:" + ppt_url)
24         r = requests.get(ppt_url)
25         self.num += 1
26         name = "./ppts/ppt_{0}.{1}".format('%05d' % self.num, suffix)
27         with open(name, "wb") as code:
28             code.write(r.content)
29
30     def parse(self, url):
31         print(url)
32         res = requests.get(url)
33         doc = pq(res.content)
34         ppt_url = doc('.downurl li a').attr('href')
35         print(ppt_url)
```

我写的是一个爬取某ppt网站的代码，就一个ppt1.py是爬虫，然后，ppts是存放下载的ppt的

## 二、准备requirement.txt文件

这个是需要哪些python库支持，写好

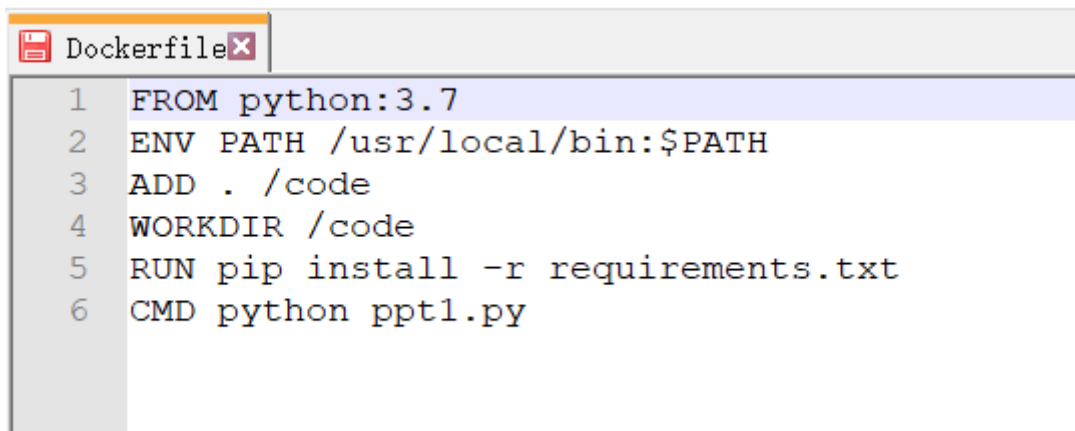


```
1 requests
2 pyquery
```

## 三、准备Dockerfile文件

需要一个名为Dockerfile的文件，没有后缀，这个创建docker镜像的配置文件

```
FROM python:3.7 --基础镜像根据自己的实际情况去调整
ENV PATH /usr/local/bin:$PATH
ADD . /code
WORKDIR /code
RUN pip install -r requirements.txt
CMD python ppt1.py
```



```
1 FROM python:3.7
2 ENV PATH /usr/local/bin:$PATH
3 ADD . /code
4 WORKDIR /code
5 RUN pip install -r requirements.txt
6 CMD python ppt1.py
```

FROM: 需要什么环境

ENV: 修改path, 即增加/usr/local/bin这个环境变量

ADD: 将本地代码放到虚拟容器中, 它有两个参数, 第一个是., 代表本地当前路径; 第二个参数是/code, 代表虚拟容器中的路径, 即将本地项目的所有内容放到虚拟容器的/code目录下, 以便在虚拟容器中运行代码

WORKDIR: 指定工作目录, 也就是刚才的/code, 在虚拟容器中的目录

RUN: 执行某些环境准备工作, docker容器中只有python3环境, 还需要python的库, 这里安装那些库

CMD: 运行项目的命令行命令

## 四、开始创建镜像

```
docker build -t pptdemo:latest .
```

```
选择MINGW64/d/AllCode/python/pptdemo
C:\code\python\pptdemo
$ docker build -t pptdemo:latest .
Sending build context to Docker daemon 23.55kB
Step 1/6 : FROM python:3.6
--> 55fb8aca33df
Step 2/6 : ENV PATH /usr/local/bin:$PATH
--> Using cache
--> 97e82715b8ee
Step 3/6 : ADD . /code
--> 9d2d253015ee
Step 4/6 : WORKDIR /code
Removing intermediate container 25ccdad420a0
--> ec462b723417
Step 5/6 : RUN pip install -r requirements.txt
--> Running in 83e607d0bc06
Collecting requests (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/7d/e3/20f3d364d6c8e5d2353c72a67778eb189176f08e873c9900e10c0287b84b/requests-2.21.0-py2.py3-none-any.whl (57kB)
Collecting pyquery (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/09/c7/ce8c9c37ab8ff8337faad3335c088d60bed4a35a4bed33a64f0e64fbcf29/pyquery-1.4.0-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl (58kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting certifi>=2017.4.17 (from requests->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/9f/e0/accfc1b56b57e9750eba272e24c4dddeac86852c2bebd1236674d7887e8a/certifi-2018.11.29-py2.py3-none-any.whl (154kB)
Collecting urllib3<1.25,>=1.21.1 (from requests->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/62/00/eeld7de624db8a7090d1226aebefab96a2c71cd5cfa7629d6ad3f61b79e/urllib3-1.24.1-py2.py3-none-any.whl (118kB)
Collecting lxml>=2.1 (from pyquery->-r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/5d/d4/e81be10be160a6323cf5f29f1eabc9693080cb16780a2e19c96091ee37ee/lxml-4.3.0-cp36-cp36m-manylinux1_x86_64.whl (5.7MB)
Collecting cssselect>0.7.9 (from pyquery->-r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/7b/44/25b7233e50585f0b4156960691d951b05d061abf4a714078393e51929b30/cssselect-1.0.3-py2.py3-none-any.whl
Installing collected packages: idna, chardet, certifi, urllib3, requests, lxml, cssselect, pyquery
Successfully installed certifi-2018.11.29 chardet-3.0.4 cssselect-1.0.3 idna-2.8 lxml-4.3.0 pyquery-1.4.0 requests-2.21.0 urllib3-1.24.1
Removing intermediate container 83e607d0bc06
--> 22244632da67
Step 6/6 : CMD python ppt1.py
--> Running in c5ff77a9f680
Removing intermediate container c5ff77a9f680
--> 07cfc786f1a
Successfully built 07cfc786f1a
Successfully tagged pptdemo:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
```

这样应该就没错了

```
$ docker build -t pptdemo:latest .
Sending build context to Docker daemon 23.55kB
Step 1/6 : FROM python:3.7
--> 55fb8aca33df
Step 2/6 : ENV PATH /usr/local/bin:$PATH
--> Using cache
--> 97e82715b8ee
Step 3/6 : ADD . /code
--> 9d2d253015ee
Step 4/6 : WORKDIR /code
Removing intermediate container 25ccdad420a0
--> ec462b723417
Step 5/6 : RUN pip install -r requirements.txt
--> Running in 83e607d0bc06
Collecting requests (from -r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/7d/e3/20f3d364d6c8e5d2353c72a67778eb189176f08e873c9900e10c0287b84b/requests-2.21.0-py2.py3-none-any.whl (57kB)
Collecting pyquery (from -r requirements.txt (line 2))
  Downloading
https://files.pythonhosted.org/packages/09/c7/ce8c9c37ab8ff8337faad3335c088d60bed4a35a4bed33a64f0e64fbcf29/pyquery-1.4.0-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl (58kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting certifi>=2017.4.17 (from requests->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/9f/e0/accfc1b56b57e9750eba272e24c4dddeac86852c2bebd1236674d7887e8a/certifi-2018.11.29-py2.py3-none-any.whl (154kB)
Collecting urllib3<1.25,>=1.21.1 (from requests->-r requirements.txt (line 1))
```

```
Downloading
https://files.pythonhosted.org/packages/62/00/ee1d7de624db8ba7090d1226aebefab96a
2c71cd5cfa7629d6ad3f61b79e/urllib3-1.24.1-py2.py3-none-any.whl (118kB)
Collecting lxml>=2.1 (from pyquery->-r requirements.txt (line 2))
Downloading
https://files.pythonhosted.org/packages/5d/d4/e81be10be160a6323cf5f29f1eabc96930
80cb16780a2e19c96091ee37ee/lxml-4.3.0-cp36-cp36m-manylinux1_x86_64.whl (5.7MB)
Collecting cssselect>0.7.9 (from pyquery->-r requirements.txt (line 2))
Downloading
https://files.pythonhosted.org/packages/7b/44/25b7283e50585f0b4156960691d951b05d
061abf4a714078393e51929b30/cssselect-1.0.3-py2.py3-none-any.whl
Installing collected packages: idna, chardet, certifi, urllib3, requests, lxml,
cssselect, pyquery
Successfully installed certifi-2018.11.29 chardet-3.0.4 cssselect-1.0.3 idna-2.8
lxml-4.3.0 pyquery-1.4.0 requests-2.21.0 urllib3-1.24.1
Removing intermediate container 83e607d0bc06
---> 22244632da67
Step 6/6 : CMD python ppt1.py
---> Running in c5ff77a9f680
Removing intermediate container c5ff77a9f680
---> 07cfec786f1a
Successfully built 07cfec786f1a
Successfully tagged pptdemo:latest
SECURITY WARNING: You are building a Docker image from windows against a non-
windows Docker host. All files and directories added to build context will have
'-rwxr-xr-x' permissions. It is recommended to double check and reset permissions
for sensitive files and directories.
```

继续输入代码查看是否创建成功

```
docker images
```

## 五、运行

```
docker run pptdemo
```

## 六、docker的导入和导出

docker 镜像导入导出有两种方法:

一种是使用 `save` 和 `load` 命令

使用例子如下:

```
docker save pptdemo:latest>/root/pptdemo.tar
docker load<pptdemo.tar
```

一种是使用 `export` 和 `import` 命令

使用例子如下:

```
docker export pptdemo:latest> pptdemo.tar
cat pptdemo.tar | docker import - pptdemo:latest
```

`export` 和 `import` 导出的是一个容器的快照，不是镜像本身，也就是说没有 `layer`。

你的 `dockerfile` 里的 `workdir`, `entrypoint` 之类的所有东西都会丢失，`commit` 过的话也会丢失。

快照文件将丢弃所有的历史记录和元数据信息（即仅保存容器当时的快照状态），而镜像存储文件将保存完整记录，体积也更大。

`docker save` 保存的是镜像（`image`），`docker export` 保存的是容器（`container`）；  
`docker load` 用来载入镜像包，`docker import` 用来载入容器包，但两者都会恢复为镜像；  
`docker load` 不能对载入的镜像重命名，而 `docker import` 可以为镜像指定新名称。