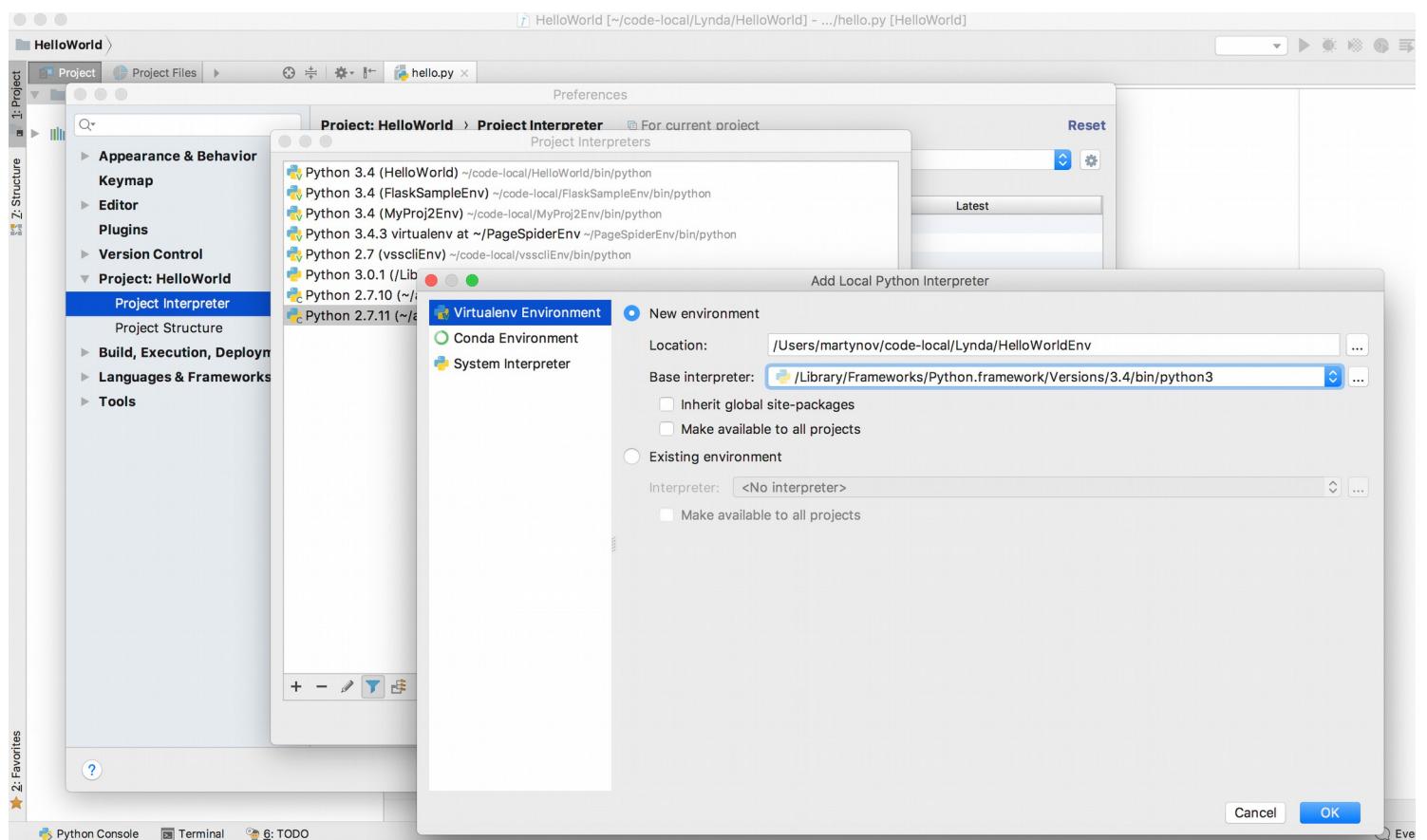


Adding new Python 3.4 virtual environment



PageSpider [~/code-local/PageSpider] - .../utilities/url_utilities.py [PageSpider]

Run Page Spider

PageSpider

Project Structure

PageSpider ~/code-local/PageSpider

- myutils
 - __init__.py
 - myurl_utils.py
- utilities
 - __init__.py
 - database_utils.py
 - url_utilities.py
 - input.txt
 - page_spider.py
- External Libraries
 - < Python 3.4.3 virtualenv at ~/Pa...
 - python3.4 library root
 - plat-darwin
 - site-packages
 - Extended Definitions
 - Binary Skeletons
 - python3.4 library root
 - lib-dyndload
 - plat-darwin
 - site-packages
 - Typedesh Stubs

```

1 import string
2 from urllib.request import urlopen
3
4 import re
5 from bs4 import BeautifulSoup
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```

New Project

Pure Python

Django

Flask

Google App Engine

Pyramid

Web2Py

Angular CLI

AngularJS

Foundation

HTML5 Boilerplate

React App

React Native

Twitter Bootstrap

Web Starter Kit

Location: /Users/martynov/code-local/MyProj3

Project Interpreter: New Virtualenv environment

New environment using Virtualenv

Location: /Users/martynov/code-local/MyProj3Env

Base interpreter: /Library/Frameworks/Python.framework/Versions/3.4/bin/python

Inherit global site-packages

Make available to all projects

Existing interpreter

Interpreter: Python 3.0.1 (/Library/Frameworks/Python.framework/Versions/3.0/python)

TODO: Project Current File Scope Based Default C

Found 5 TODO items in 2 files

- PageSpider (5 items in 2 files)
 - myutils (3 items in 1 file)
 - myurl_utils.py
 - (2, 6) #TODO: add the code needed to read the text file
 - (6, 6) #TODO: add the code that reads the url
 - (10, 6) #TODO:
 - utilities (2 items in 1 file)

Create

PageSpider [~/code-local/PageSpider] - .../utilities/url_utilities.py [PageSpider]

Project Structure

PageSpider ~/code-local/PageSpider

- myutils
 - __init__.py
 - myurl_utils.py
- utilities
 - __init__.py
 - database_utils.py
 - url_utilities.py
 - input.txt
 - page_spider.py
- External Libraries
 - < Python 3.4.3 virtualenv at ~/P...
 - python3.4 library root
 - plat-darwin
 - site-packages
 - Extended Definitions
 - Binary Skeletons
 - python3.4 library root
 - lib-dynload
 - plat-darwin
 - site-packages
 - Typeshed Stubs

Code Editor (url_utilities.py)

```
import string
from urllib.request import urlopen

import re
from bs4 import BeautifulSoup

def load_urls_from_file(file_path: str):
    try:
        with open(file_path) as f:
            contents = f.readlines()
        return contents
    except FileNotFoundError:
        print("The file " + file_path + " could not be found")
        exit(2)

def load_html_from_file(file_path: str) -> BeautifulSoup:
    res = requests.get(file_path)
    html = res.text
    return BeautifulSoup(html, "lxml")

def scrape_chicken_nuggets(url: str) -> list[dict]:
    for script in chicken_nuggets(url):
        yield script.extract()

    def load_html_from_file(file_path: str) -> BeautifulSoup:
        res = requests.get(file_path)
        html = res.text
        return BeautifulSoup(html, "lxml")

    def scrape_chicken_nuggets(url: str) -> list[dict]:
        for script in chicken_nuggets(url):
            yield script.extract()
```

New Package dialog

Enter new package name:

TODO: Project Current File Scope Based Default Changelist

Found 5 TODO items in 2 files

- PageSpider (5 items in 2 files)
 - myutils (3 items in 1 file)
 - myurl_utils.py
 - (2, 6) #TODO: add the code needed to read the text file
 - (6, 6) #TODO: add the code that reads the url
 - (10, 6) #TODO:
 - utilities (2 items in 1 file)
 - database_utils.py
 - (2, 7) # TODO: generate the database
 - (7, 7) # TODO: save words to the database

Run TODO Version Control Python Console Terminal Event Log

Subscription Validation: Your PyCharm subscription expires on 2018-01-08. // After this date you will no longer be able to use the product (23 minutes ago)

PageSpider - [C:\Users\Bruce Van Horn\PycharmProjects\PageSpider] - ...utilities\url_utilities.py - PyCharm 2017.1.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

PageSpider > utilities > url_utilities.py

Project page_spider.py url_utilities.py database_utilities.py

Run Page Spider

```
load_urls_from_...
1 def load_urls_from_file(file_path: str):
2     #TODO: add the code needed to read the text file with the urls in it
3     pass
4
5 def load_page(url: str):
6     #TODO: add the code that reads the url
7     pass
8
9 def scrape_page(page_contents: str):
10    #TODO: analyze the text
11    pass
12
```

TODO: Project Current File Scope Based

Found 5 TODO items in 2 files

- PageSpider (5 items in 2 files)
 - utilities (5 items in 2 files)
 - database_utilities.py
 - (2, 6) #TODO: generate the database
 - (6, 6) #TODO: save the words to the database
 - url_utilities.py
 - (2, 6) #TODO: add the code needed to read the text file with the urls in it
 - (6, 6) #TODO: add the code that reads the url
 - (10, 6) #TODO: analyze the text

Packages installed successfully: Installed packages: 'beautifulsoup4' (7 minutes ago)

LinkedIn

PageSpider [~/code-local/PageSpider] - .../utilities/url_utilities.py [PageSpider]

Project

Z-Structure

PageSpider ~/code-local/PageSpider

- myutils
 - __init__.py
 - myurl_utils.py
- utilities
 - __init__.py
 - database_utils.py
 - url_utilities.py
- input.txt
- page_spider.py

External Libraries

- < Python 3.4.3 virtualenv at ~/PageSpiderEnv
 - python3.4 library root
 - plat-darwin
 - site-packages
 - Extended Definitions
 - Binary Skeletons
 - python3.4 library root
 - lib-dynload
 - plat-darwin
 - site-packages
 - Typeshed Stubs

Run/Debug Configurations

Name: Run Page Spider

Configuration Logs

Script path: /Users/martynov/code-local/PageSpider/page_spider.py

Parameters: -i input.txt -db words.db

Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Python 3.4.3 virtualenv at ~/PageSpiderEnv

Interpreter options:

Working directory: /Users/martynov/code-local/PageSpider

Add content roots to PYTHONPATH (checked)

Add source roots to PYTHONPATH (checked)

Emulate terminal in output console (unchecked)

Show command line afterwards (unchecked)

Before launch: Activate tool window

There are no tasks to run before launch

Show this page (unchecked)

Activate tool window (checked)

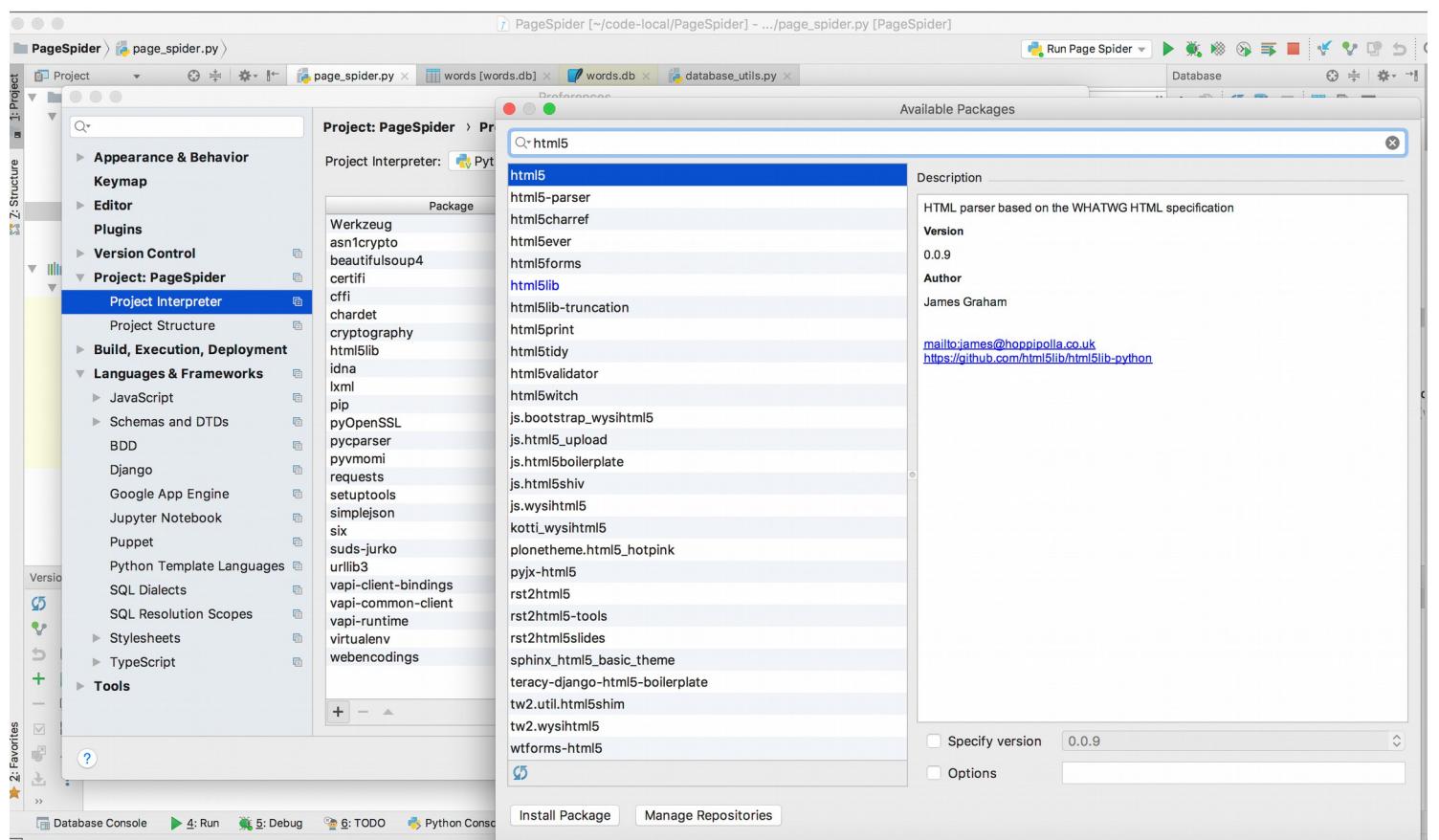
Cancel Apply OK

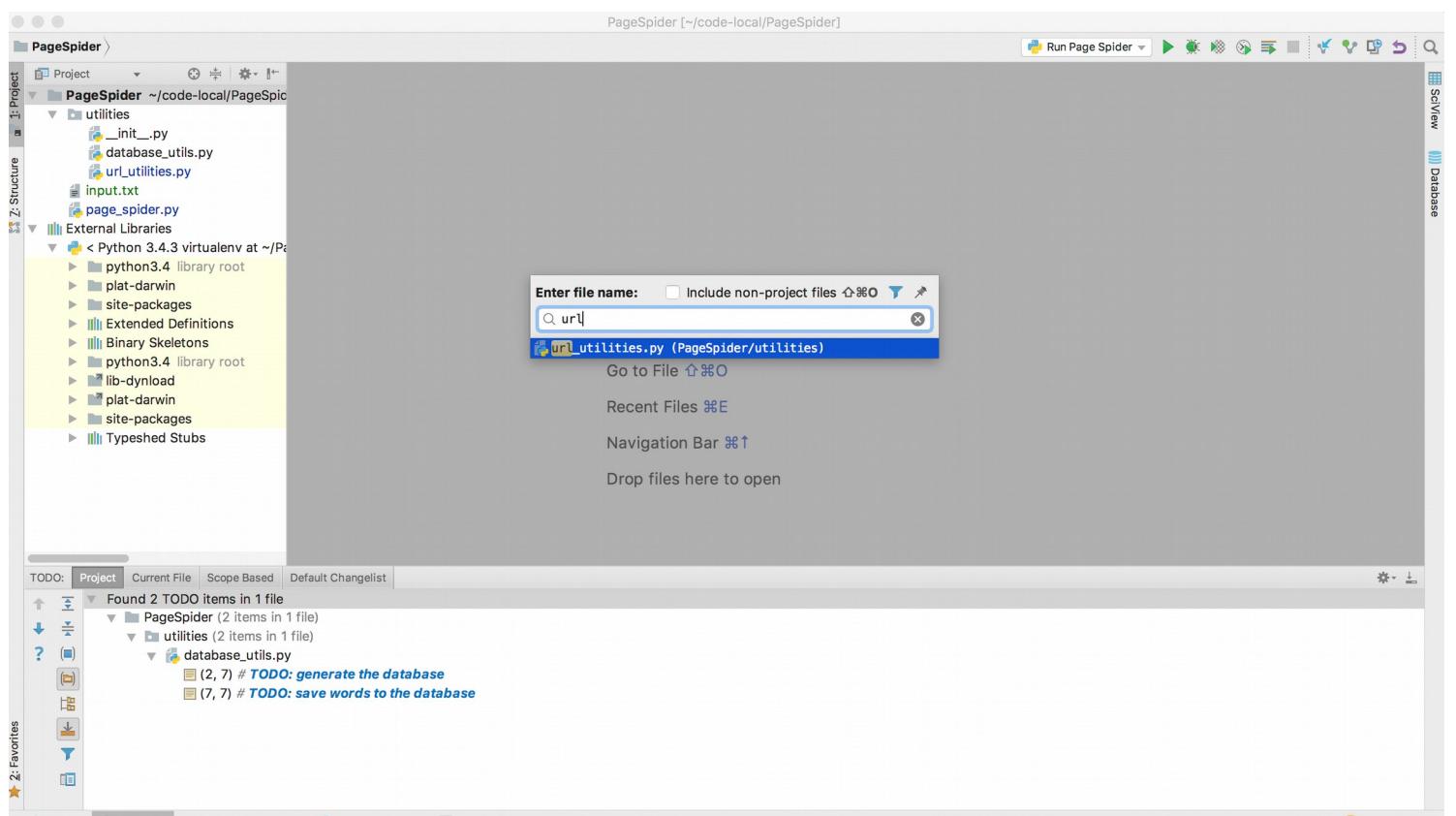
Subscription Validation: Your PyCharm subscription expires on 2018-07-05. After this date you will no longer be able to use the product (10 minutes ago)

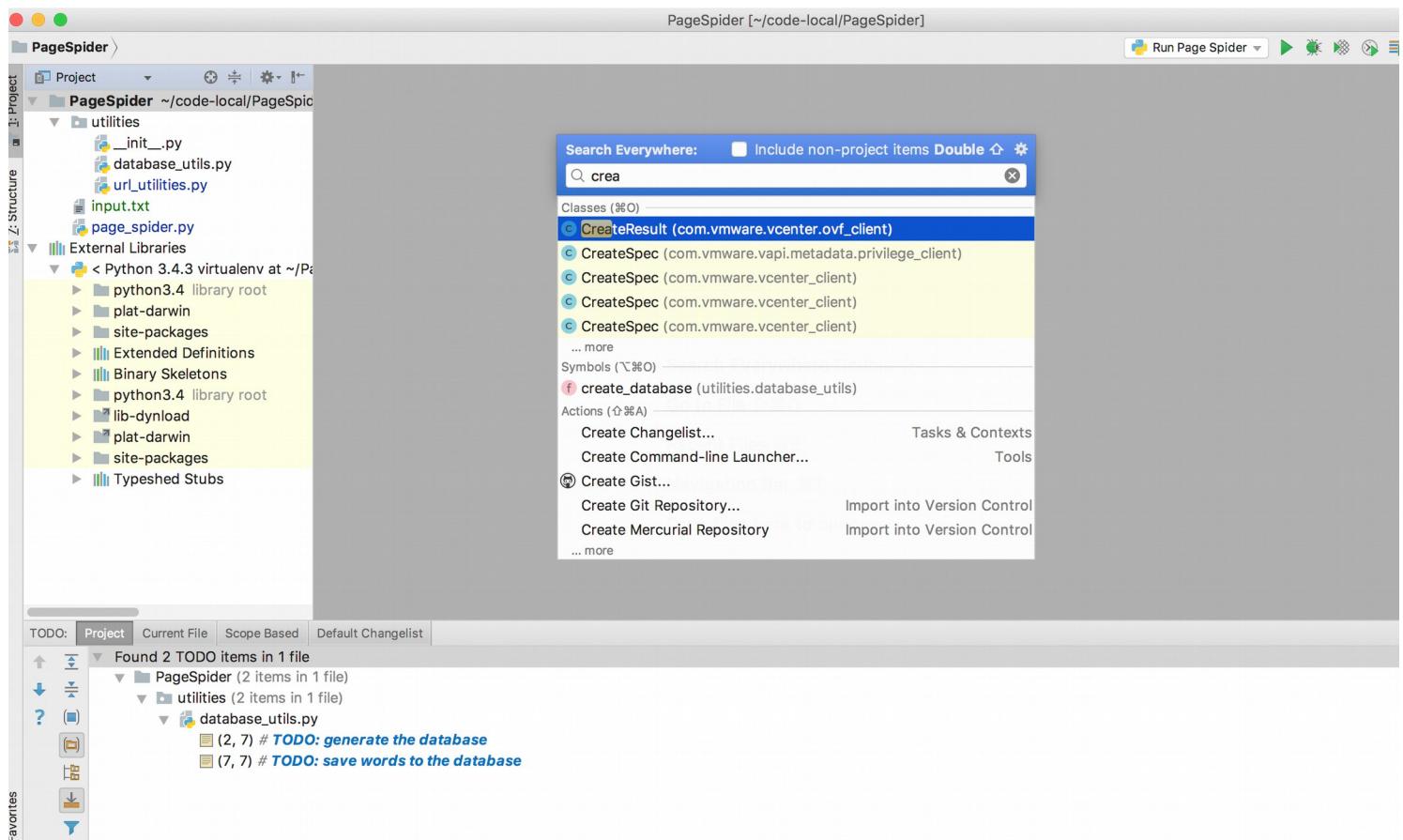
TODO: Project Current File Scope Based Default

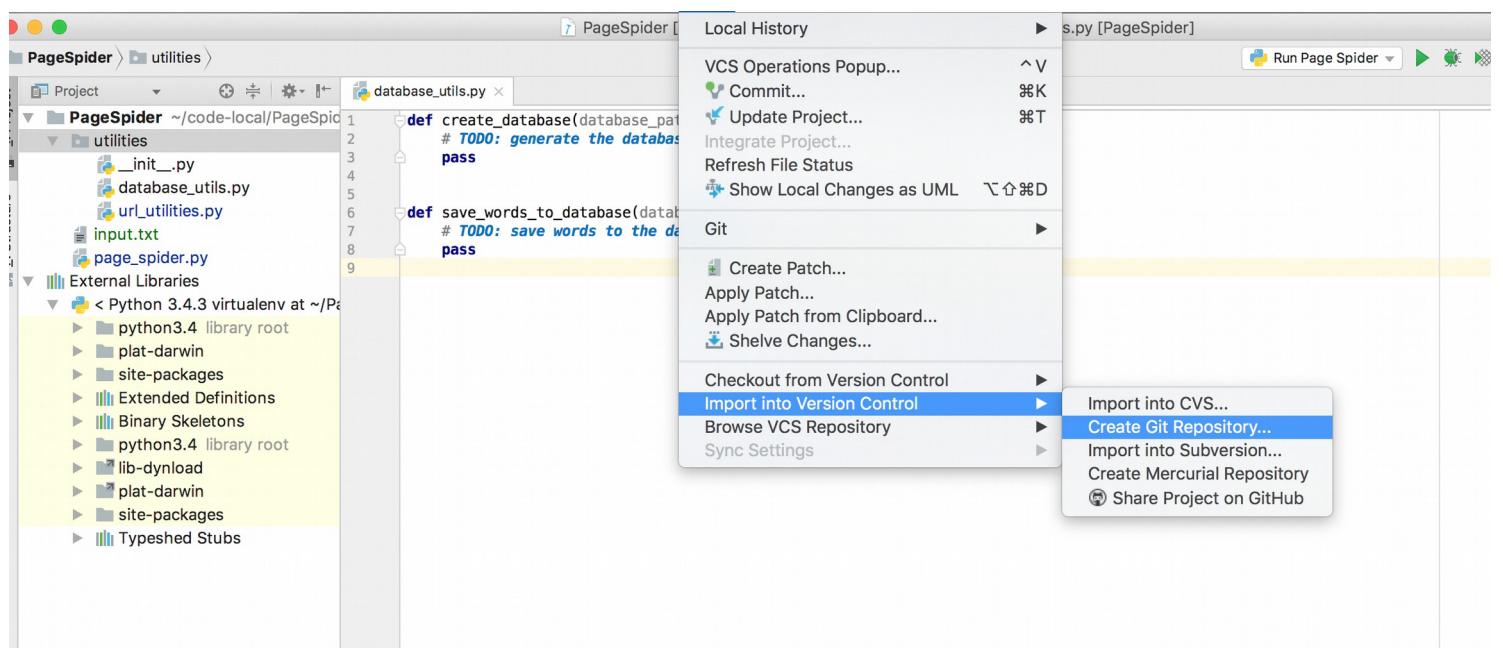
Found 5 TODO items in 2 files

- PageSpider (5 items in 2 files)
 - myutils (3 items in 1 file)
 - myurl_utils.py
 - (2, 6) #TODO: add the
 - (6, 6) #TODO: add the
 - (10, 6) #TODO:
 - utilities (2 items in 1 file)
 - database_utils.py
 - (2, 7) # TODO: generate
 - (7, 7) # TODO: save









PageSpider [~/code-local/PageSpider] - .../utilities/database_utils.py [PageSpider]

Project utilities database_utils.py

PageSpider ~/code-local/PageSpider

 └ utilities

- ├ _init__.py
- ├ database_utils.py
- └ url_utilities.py

 └ page_spider.py

External Libraries

 └ Python 3.4.3 virtualenv at ~/Pa

- ├ python3.4 library root
- ├ plat-darwin
- ├ site-packages
- ├ Extended Definitions
- ├ Binary Skeletons
- ├ python3.4 library root
- ├ lib-dyld
- ├ plat-darwin
- └ site-packages
- └ Typed stubs

database_utils.py

```
1 def create_database(database_path: str):
2     # TODO: generate the database
3     pass
4
5 def save_words_to_database(database_path: str, word_list: list):
6     # TODO: save words to the database
7     pass
8
9
```

Clone Repository

Git Repository URL: <https://github.com/oobii/2---Introduction-to-UIKit.git>

Parent Directory: /Users/martynov/code-local

Directory Name: 2---Introduction-to-UIKit

Version Control: Local Changes Log Console

Default

S V D L

Alt-Enter on the wiggly line

Reformat Code: Alt-Cmd-L

The screenshot shows the PyCharm IDE interface. The project navigation bar at the top indicates the current file is `page_spider.py`. The left sidebar displays the project structure under the `PageSpider` root, including a `utilities` folder containing `__init__.py`, `database_utils.py`, `url_utilities.py`, `input.txt`, and `page_spider.py`. Below this is an `External Libraries` section showing a virtual environment named `< Python 3.4.3 virtualenv at ~/Pa...` with sub-folders `python3.4 library root`, `plat-darwin`, `site-packages`, and `Extended Definitions`. The main editor window shows the following Python code:

```
1 import os
2 import argparse
3 from utilities import url_utilities
4
5 def main(database: str, url_list_file: str):
6     big_word_list = []
7     print("we are going to wor")
8     print("we are going to sca")
9     urls = url_utilities.load_
10    for url in urls:
11        print("reading " + url)
12        page_content = url_utilities.load_page(url=url)
13        words = url_utilities.scrape_page(page_contents=p)
14        big_word_list.extend(words)
15
16 if __name__ == "__main__":
17
```

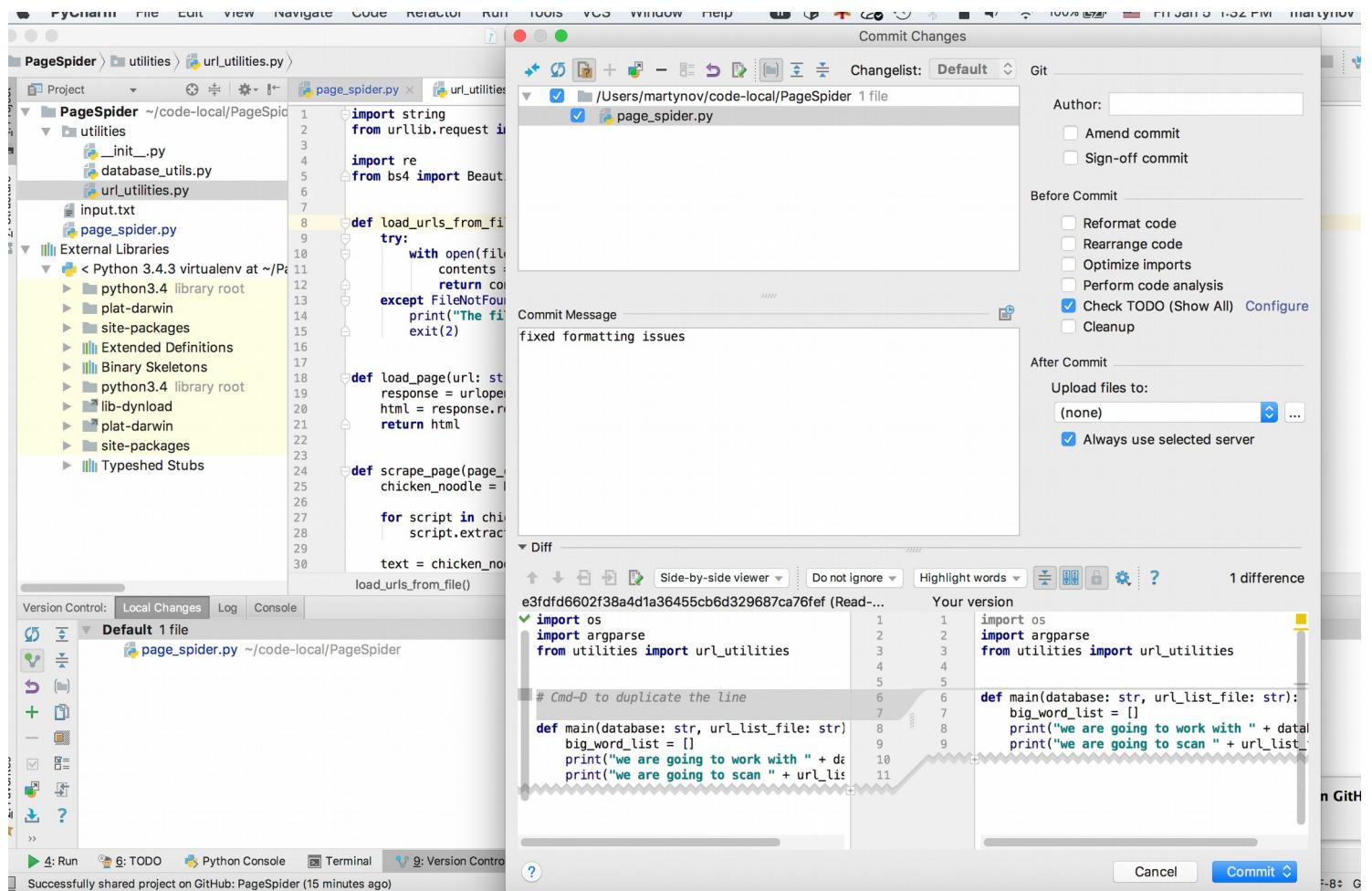
A context menu is open over the line `def main(database: str, url_list_file: str):`, listing options: `Reformat file`, `Edit inspection profile sett...`, `Ignore errors like this`, `Specify type for reference`, and `...`.

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar displays the project structure for "PageSpider". It includes files like `__init__.py`, `database_utils.py`, `url_utilities.py`, `input.txt`, and `page_spider.py`. A virtual environment for Python 3.4.3 is also listed.
- Code Editor:** The main window shows the content of `page_spider.py`. The code defines a `main` function that reads URLs from a file, loads pages, scrapes content, and prints words. It also handles command-line arguments for database and input files.
- Toolbars and Menus:** Standard PyCharm menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help are visible at the top.
- Status Bar:** The bottom status bar shows the current time as 1:20 PM and the date as Friday, January 17, 2014.

```
1 import os [Unused import statement]
2 import argparse
3 from utilities import url_utilities
4
5
6 def main(database: str, url_list_file: str):
7     big_word_list = []
8     print("we are going to work with " + database)
9     print("we are going to scan " + url_list_file)
10    urls = url_utilities.load_urls_from_file(url_list_file)
11    for url in urls:
12        print("reading " + url)
13        page_content = url_utilities.load_page(url=url)
14        words = url_utilities.scrape_page(page_contents=page_content)
15        big_word_list.extend(words)
16
17    if __name__ == "__main__":
18        parser = argparse.ArgumentParser()
19        parser.add_argument("-db", "--database", help="SQLite File Name")
20        parser.add_argument("-i", "--input", help="File containing urls to read")
21
```

Select “Default” to Commit all at once



Alt-Enter on squiggly line to get suggestion how to fix the error, it will add import line at the top of the file

The screenshot shows the PyCharm IDE interface with the title bar "PageSpider [~/code-local/PageSpider] - .../utilities/url_utilities.py [PageSpider]". The project tree on the left shows a "PageSpider" project with a "utilities" directory containing files like __init__.py, database_utils.py, url_utilities.py, input.txt, and page_spider.py. The "External Libraries" section lists Python 3.4.3 virtualenv at ~/Path, python3.4 library root, plat-darwin, site-packages, Extended Definitions, Binary Skeletons, and python3.4 library root. The code editor window displays "url_utilities.py" with the following content:

```
1 import string
2
3 import re
4 from bs4 import BeautifulSoup
5
6
7 def load_urls_from_file(file_path: str):
8     try:
9         with open(file_path) as f:
10             contents = f.readlines()
11             return contents
12     except FileNotFoundError:
13         print("The file " + file_path + " could not be found")
14         exit(2)
15
16
17 def load_page(url: str):
18     response = urlopen(url)
19     html = response.read()
20     return html
21
22
23 def scrape_page(page):
24     chicken_noodle =
25
26     for script in chicken_noodle:
27         script.extract()
28
29     text = chicken_noodle.get_text()
30
load_page()
```

A code completion dropdown is open over the line "response = urlopen(url)". The suggestions include:

- Import 'urllib.request.urlopen()'
- Import 'urllib.request.urlopen()' locally
- Create function 'urlopen'
- Create parameter 'urlopen'
- Rename reference
- Ignore unresolved reference 'utilities.url_utilities.urlopen'
- Mark all unresolved attributes of 'utilities.url_utilities' as ignored

PageSpider [~/code-local/PageSpider] - .../utilities/url_utilities.py [PageSpider]

Run Page Spider

Project PageSpider utilities url_utilities.py

PageSpider ~/code-local/PageSpider

utilities

- _init_.py
- database_utils.py
- url_utilities.py

External Libraries

- < Python 3.4.3 virtualenv at ~/Pe
- python3.4 library root
- plat-darwin
- site-packages
- Extended Definitions
- Binary Skeletons
- python3.4 library root
- lib-dyld
- plat-darwin
- site-packages
- Typeshed Stubs

url_utilities.py

```
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
```

for script in chicken_noodle(["script", "style"]):
 script.extract()

text = chicken_noodle.get_text()
lines = (line.strip() for line in text.splitlines())
chunks = (phrase.strip() for line in lines for phrase in line.split(" "))

text = ' '.join(chunk for chunk in chunks if chunk)
plain_text = '.join(chunk for chunk in chunks if chunk)'
clean_words = []
words = plain_text.split()
for word in words:
 clean = True
 # no punctuation
 for punctuation in punctuation:
 if punctuation in word:
 clean = False
 break
 # no numbers
 if any(char.isdigit() for char in word):
 clean = False
 # at least one letter
 if len(word) < 2:
 clean = False
 if clean:
 clean_words.append(word)

scrape_page() > for word in words:
 clean = True
 for punctuation in punctuation:
 if punctuation in word:
 clean = False
 break
 if any(char.isdigit() for char in word):
 clean = False
 if len(word) < 2:
 clean = False
 if clean:
 clean_words.append(word)

Copy Reference ⌘C

Paste ⌘V

Paste from History... ⌘⇧V

Paste Simple ⌘⇧V

Column Selection Mode ⌘B

Find Usages ⌘F7

Refactor ▶

- Rename... ⌘F6
- Change Signature... ⌘F6
- Move... F6
- Copy... F5
- Safe Delete... ⌘⌫
- Extract ▶
- Inline... ⌘N
- Pull Members Up...
- Push Members Down...
- Convert to Python Package
- Convert to Python Module

Local Changes Log Console

Default 1 file

url_utilities.py ~/code-local/PageSpider/utilities

Version Control

File Project

PageSpider [~/code-local/PageSpider] - .../page_spider.py [PageS

Project

PageSpider ~/code-local/PageSpider

utilities

- __init__.py
- database_utils.py
- url_utilities.py
- input.txt
- page_spider.py

External Libraries

- < Python 3.4.3 virtualenv at ~/Pa
- python3.4 library root
- plat-darwin
- site-packages
- Extended Definitions
- Binary Skeletons

page_spider.py

```
1 import os
2 import argparse
3 from utilities import url_utilities
4
5
6 def main(database: str, url_list_file: str):
7     big_word_list = []
8     print("we are going to work with " + database)
9     print("we are going to scan " + url_list_file)
10    urls = url_utilities.load_urls_from_file(url_list_file)
11    for url in urls:
12        print("reading " + url)
13        page_content = url_utilities.load_page(url=url)
14        words = url_utilities.scrape_page(page_contents=page_content)
15        big_word_list.extend(words)
16
17    .. . . .
```

Example how to work with Python packages
See “from ... import ...”
And the selected code

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'PageSpider' located at '~/code-local/PageSpider'. It contains a 'utilities' folder with '_init_.py', 'database_utils.py', and 'url_utilities.py', along with files 'input.txt' and 'page_spider.py'. The right panel shows the code editor for 'page_spider.py'. The code is as follows:

```
1  import os
2  import argparse
3  from utilities import url_utilities
4
5
6  def main(database: str, url_list_file: str):
7      big_word_list = []
8      print("we are going to work with " + database)
9      print("we are going to scan " + url_list_file)
10     urls = url_utilities.load_urls_from_file(url_list_file)
11     for url in urls:
12         print("reading " + url)
13         page_content = url_utilities.load_page(url=url)
14         words = url_utilities.scrape_page(page_contents=page_content)
15         big_word_list.extend(words)
16
17
18 if __name__ == "__main__":
19     parser = argparse.ArgumentParser()
20     parser.add_argument("-db", "--database", help="SQLite File Name")
21     parser.add_argument("-i", "--input", help="File containing urls to read")
22     args = parser.parse_args()
23     database_file = args.database
24     input_file = args.input
25     main(database=database_file, url_list_file=input_file)
26
```

The code editor highlights the section from line 10 to line 15, which corresponds to the selected code in the question. The PyCharm interface includes tabs for 'page_spider.py', 'url_utilities.py', and 'database_utils.py'.

Debugging: click the bug at the top

The screenshot shows the PyCharm IDE interface with the following details:

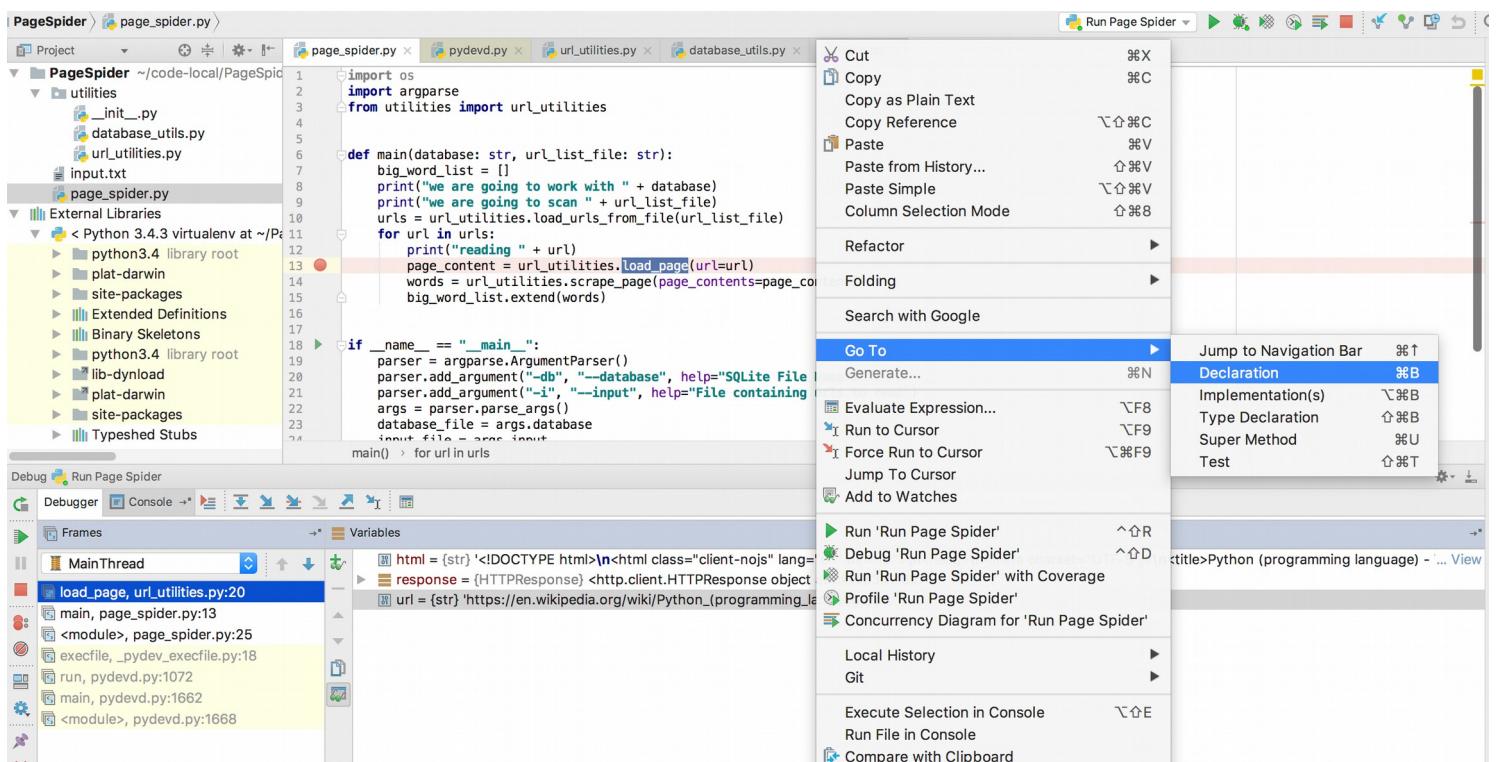
- Project Structure:** The project is named "PageSpider" located at "/code-local/PageSpider". It contains files: __init__.py, database_utils.py, url_utilities.py, input.txt, and page_spider.py.
- Code Editor:** The file "page_spider.py" is open. A red circular breakpoint icon is visible on line 12, and a blue lightbulb icon is on line 13. The code is as follows:

```
1 import os
2 import argparse
3 from utilities import url_utilities
4
5
6 def main(database: str, url_list_file: str):
7     database: 'words.db' url_list_file: 'input.txt'
8     big_word_list = [] big_word_list: <class 'list'>: []
9     print("we are going to work with " + database)
10    print("we are going to scan " + url_list_file)
11    urls = url_utilities.load_urls_from_file(url_list_file) urls: <class 'list'>: ['https://en.wikipedia.org/wiki/Python_(programming_language)\n', 'http://en.wikipedia.org/wiki/Python_(programming_language)\n']
12    for url in urls: url: 'https://en.wikipedia.org/wiki/Python_(programming_language)\n'
13        print("reading " + url)
14        page_content = url_utilities.load_page(url=url)
15        words = url_utilities.scrape_page(page_contents=page_content)
16        big_word_list.extend(words)
17
18 if __name__ == "__main__":
19     parser = argparse.ArgumentParser()
20     parser.add_argument("--db", "--database", help="SQLite File Name")
21     parser.add_argument("-i", "--input", help="File containing urls to read")
22     args = parser.parse_args()
23     database_file = args.database
24     input_file = args.input
25     main(database_file, url_list_file=input_file)
```

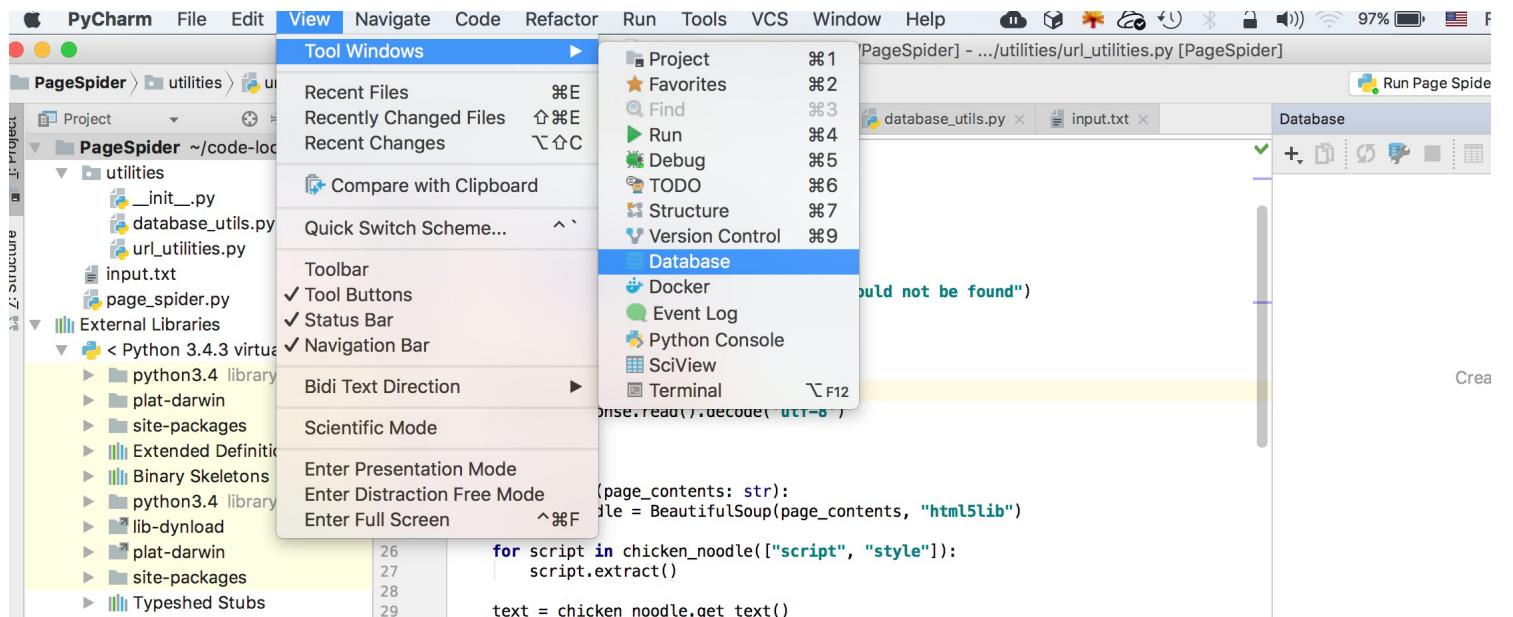
- Breakpoint:** A red circular breakpoint icon is on line 12.
- Variables:** The "Variables" tool window shows the following variables:
 - big_word_list: []
 - database: 'words.db'
 - url: 'https://en.wikipedia.org/wiki/Python_(programming_language)\n'
 - url_list_file: 'input.txt'
 - urls: ['https://en.wikipedia.org/wiki/Python_(programming_language)\n', 'https://en.wikipedia.org/wiki/Guido_van_Rossum\n', 'https://en.wiki...
- Bottom Bar:** The "Run Page Spider" button is visible.

GoTo function declaration: Cmd-B

Cmd-[Back
Cmd-] Forward



Working with Databases



Select PageSpider, CopyPath and use it for file path
Press Download to download driver

The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays a file structure for a 'PageSpider' project. The 'url_utilities.py' file is open in the main code editor, showing Python code for URL handling. On the right, the Database tool window is open, showing a list of available database drivers. The 'Sqlite (Xerial)' driver is selected and highlighted in blue.

```
6     def load_urls_from_file(file_path: str):
7         try:
8             with open(file_path) as f:
9                 contents = f.readlines()
10            return contents
11        except FileNotFoundError:
12            print("The file " + file_path + " could not be found")
13            exit(2)
14
15
16     def load_page(url: str):
17         response = urlopen(url)
18         html = response.read().decode('utf-8')
19         return html
20
21
22     def scrape_page(page_contents: str):
23         chicken_noodle = BeautifulSoup(page_contents, "html5lib")
24
25         for script in chicken_noodle(["script", "style"]):
26             script.extract()
27
28             text = chicken_noodle.get_text()
29             lines = (line.strip() for line in text.splitlines())
30             chunks = (phrase.strip() for line in lines for phrase in line.split(" "))
31
32             text = '\n'.join(chunk for chunk in chunks if chunk)
33             plain_text = ''.join(filter(lambda x: x in string.printable, text))
34
35
load_page()
```

Database

- + Data Source >
- DDL Data Source
- Data Source from URL
- Data Source from Path
- Import from Sources...

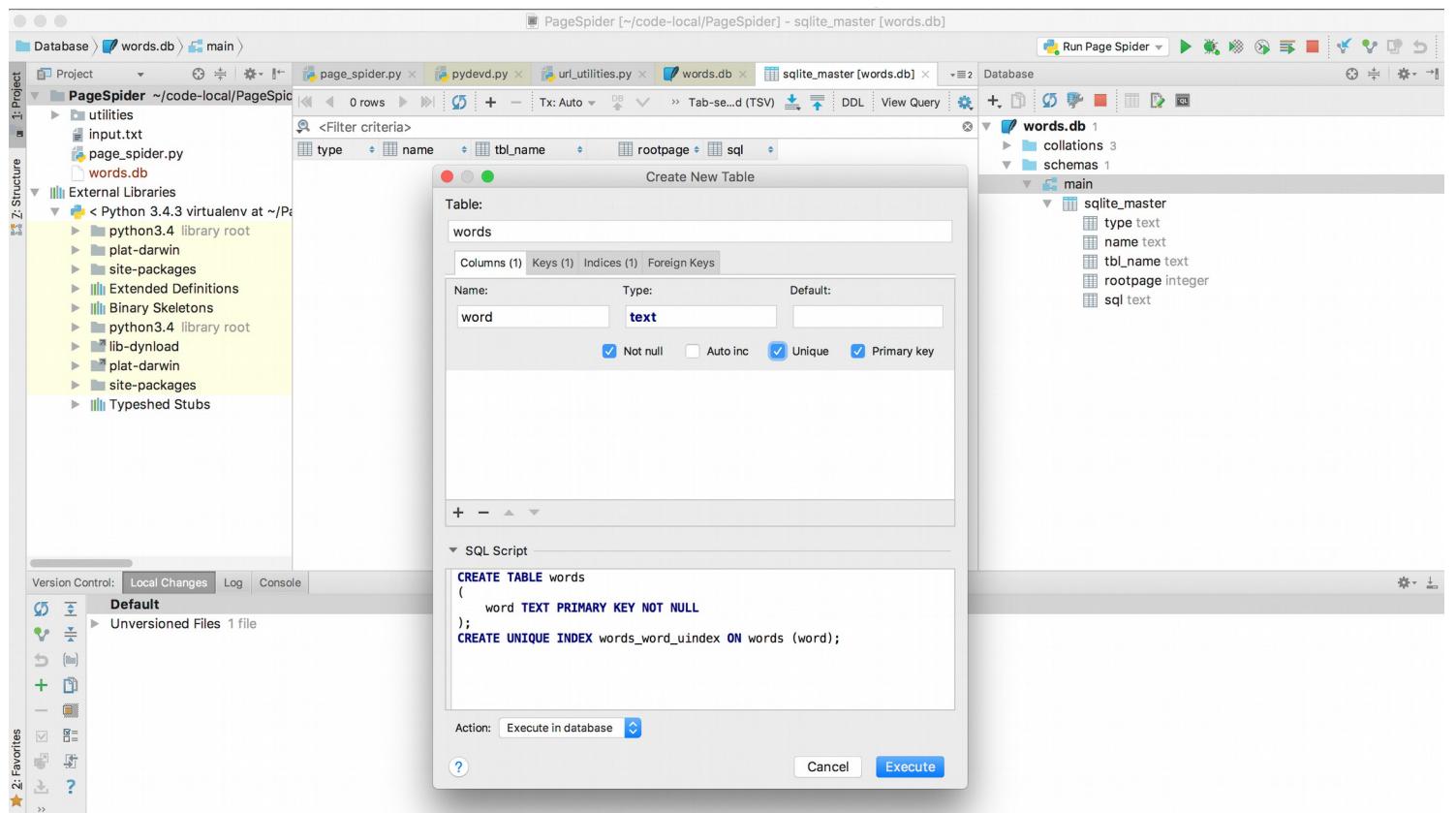
Create a new data source

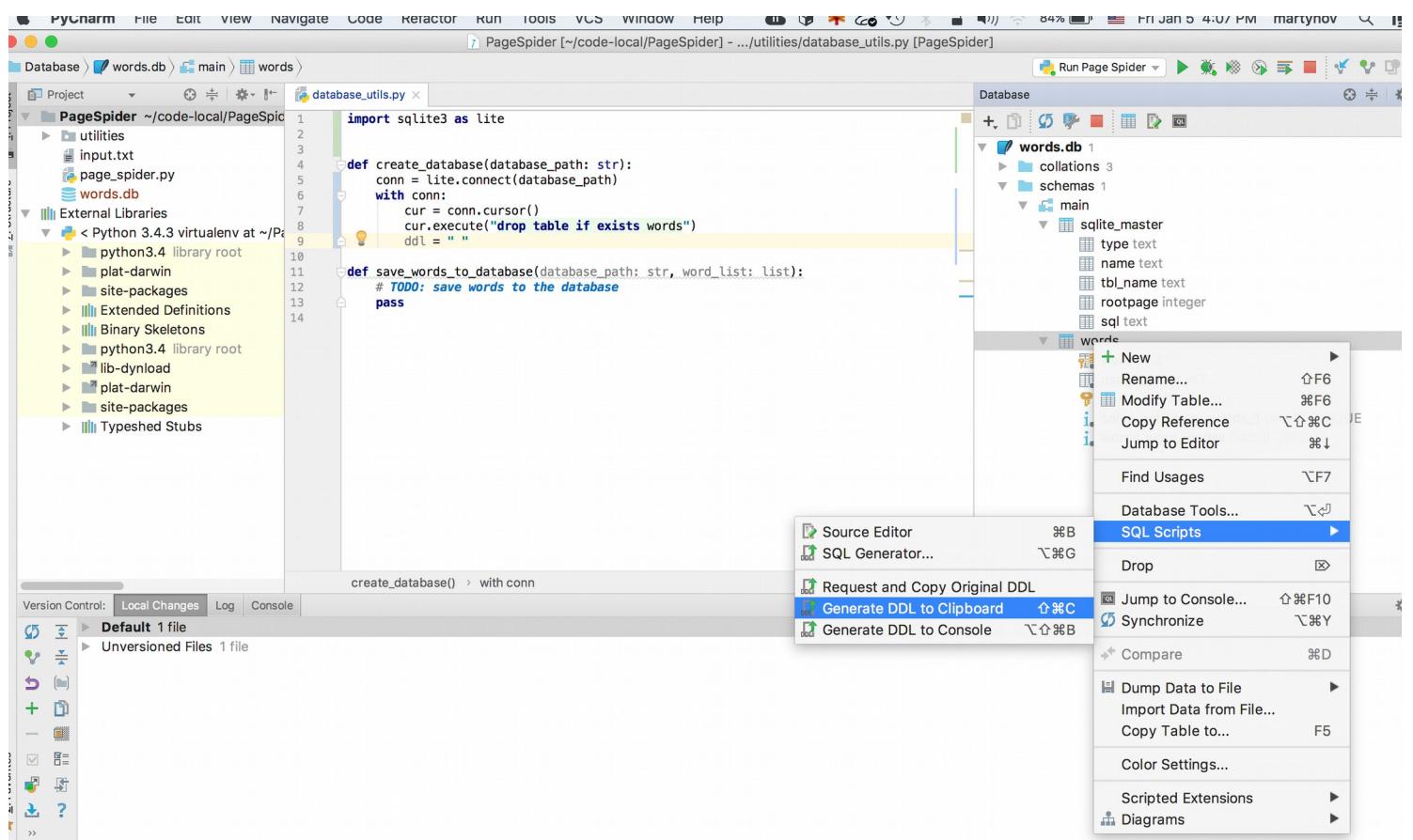
- Amazon Redshift
- Azure (Microsoft)
- DB2 (JTOpen)
- DB2 (LUW)
- Derby (Embedded)
- Derby (Remote)
- Exasol
- H2
- HSQLDB (Local)
- HSQLDB (Remote)
- MySQL
- Oracle
- PostgreSQL
- SQL Server (jTds)
- SQL Server (Microsoft)
- SQLite (Xerial) **Selected**
- Sybase (jTds)
- Sybase (Native)

The screenshot shows the PyCharm IDE interface with the following details:

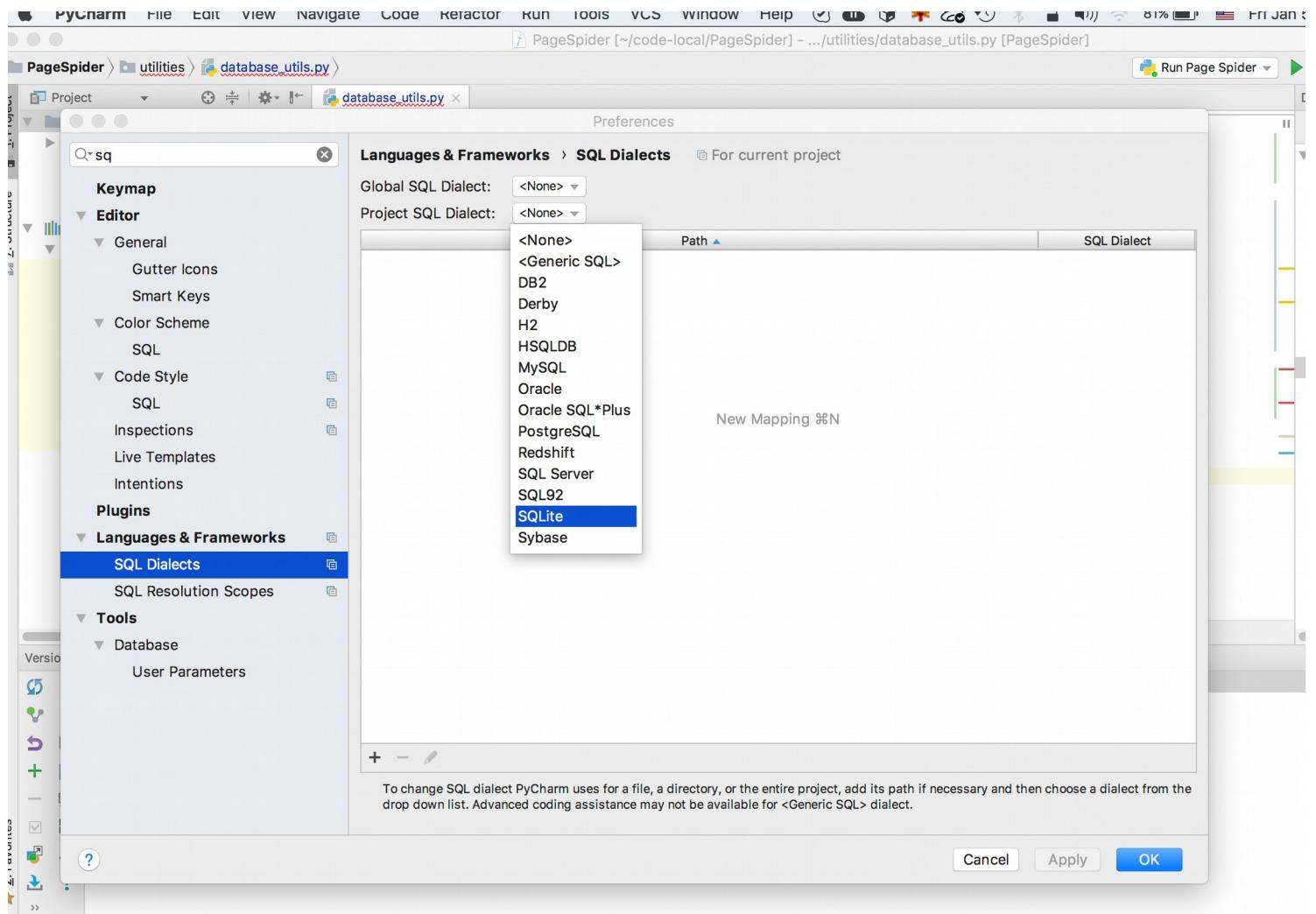
- Project:** PageSpider (~/code-local/PageSpider)
- Code Editor:** Opened file page_spider.py containing Python code for loading URLs from a file.
- Database Configuration Dialog:** A modal window titled "Data Sources and Drivers" is open, showing the configuration for a database named "words.db".
- General Tab:** The "Name" field is set to "words.db". The "File" field contains the path "/Users/martynov/code-local/PageSpider/words.db". The "URL" field shows "jdbc:sqlite:/Users/martynov/code-local/PageSpider/words.db".
- Driver Selection:** The "Driver" is set to "Sqlite (Xerial)".
- Drivers List:** A list of available drivers is shown, including Amazon Redshift, Azure (Microsoft), DB2 (JTOpen), DB2 (LUW), Derby (Embedded), Derby (Remote), Exasol, H2, HSQLDB (Local), HSQLDB (Remote), MySQL, Oracle, PostgreSQL, SQL Server (jTds), SQL Server (Microsoft), Sqlite (Xerial), Sybase (jTds), and Sybase (Native).
- Buttons:** The dialog includes "Cancel", "Apply", and "OK" buttons at the bottom right.

Creating Table and columns





Setting Project wide SQL dialect



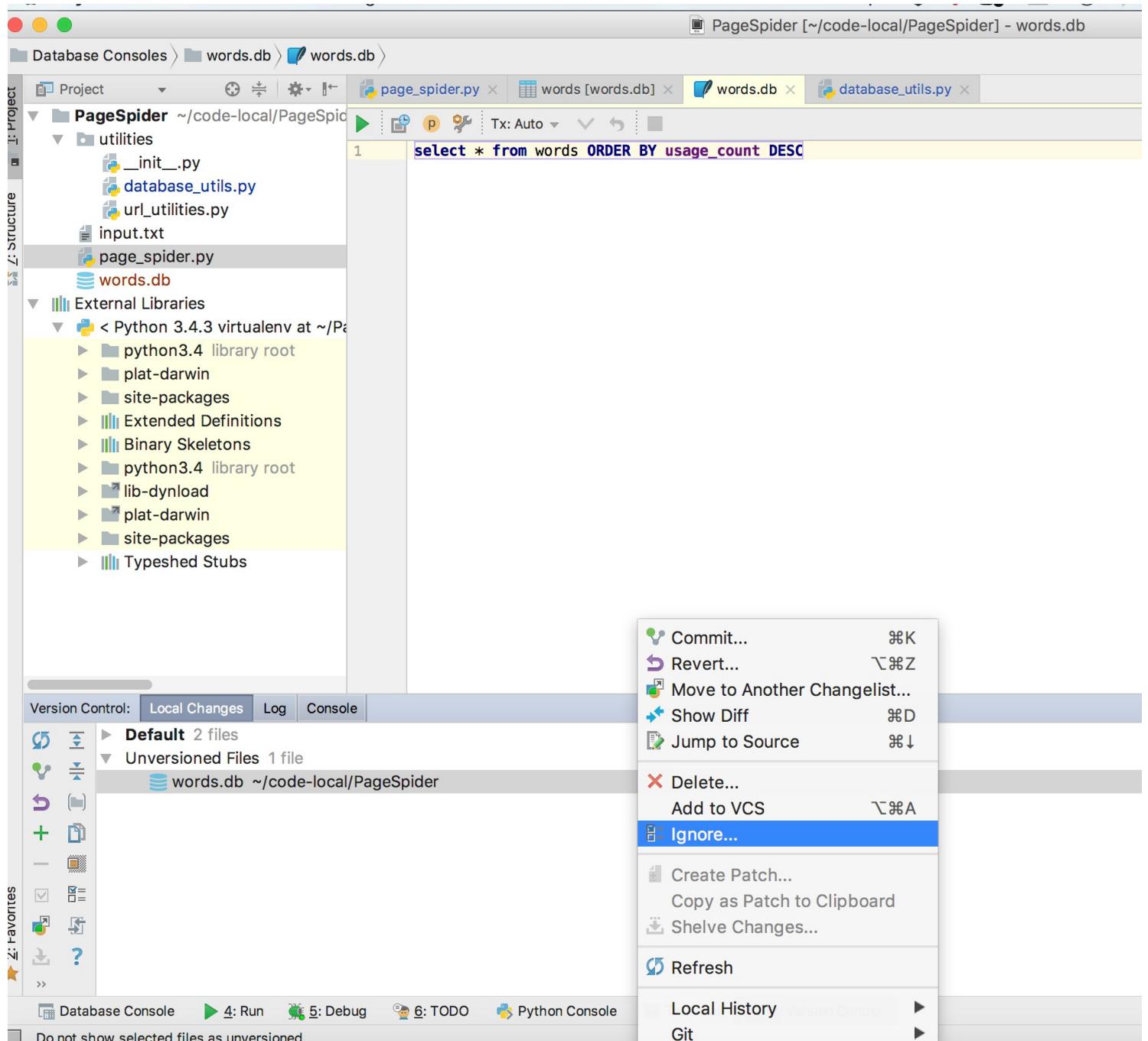
Open SQL console to run query

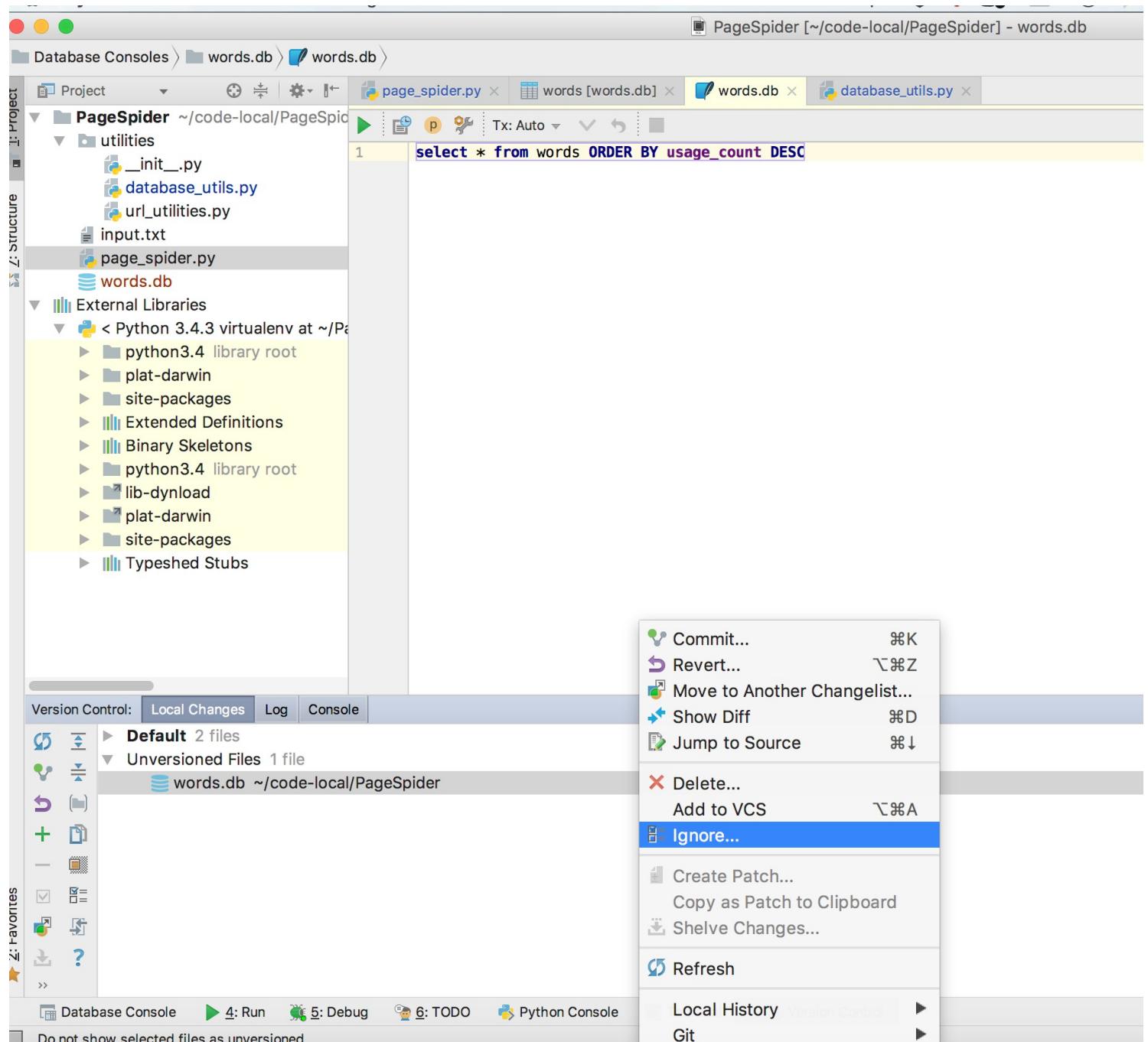
The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays a Python project named "PageSpider" with files like `__init__.py`, `database_utils.py`, `url_utilities.py`, `input.txt`, and `page_spider.py`. A yellow selection bar highlights the `words.db` file. In the center, the Database Console window shows the results of the query `select * from words ORDER BY word`. The results table has two columns: `word` and `usage_count`. The data is as follows:

word	usage_count
__add__	26
__future__	26
abc	160
abilities	32
ability	32
about	448
academic	32
access	32
according	96
...	...

On the right, the Database tool window shows the structure of the `words.db` database. It contains a single table named `words` with columns `word` (TEXT), `usage_count` (INT), and `sqlite_autoindex_words` (an index). The `sqlite_master` table is also visible.

Ignore file in the Git version control





This is done through Terminal
pip freeze > requirements.txt
Then add it to Git repo and commit

