

PyCharm



Error404

Что такое PyCharm?

PyCharm — интегрированная среда разработки, с полным набором средств предназначенных для разработки на языке Python



Необходимое программное и аппаратное обеспечение

Операционная система	<input type="checkbox"/>	64-битная версия Microsoft Windows 10, 8 macOS 10.14 или выше Любой дистрибутив Linux, поддерживающий Gnome, KDE или Unity DE
Процессор	<input type="checkbox"/>	Любой современный многоядерный CPU
RAM	<input type="checkbox"/>	Не менее 4 ГБ, рекомендуется 8 ГБ
Место на диске	<input type="checkbox"/>	Не менее 10 ГБ

Функции и возможности

1

Встроенный визуальный отладчик

2

Возможность запуска в любом окружении

3

Интерактивную консоль и встроенный терминал

4

Инструменты для работы с базами данных

5

Инструменты для работы веб-разработки

Функции и возможности

PyCharm поддерживает все основные реализации языка Python: Python 2.x и 3.x, Jython, IronPython, PyPy и Cython, что позволяет ему делать следующее:

1

Подсвечивать синтаксис

2

Проверять код на ошибки с их указанием ещё до компиляции и запуска

3

Проводить автодополнение кода

4

Обеспечивать навигацию по коду и возможность просматривать структуру кода

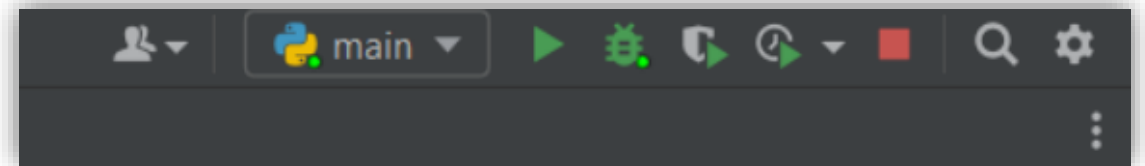
5

Проводить рефакторинг кода

Также важно отметить, что PyCharm - это кроссплатформенная IDE, возможности которой пользователь всегда может расширить в зависимости от своих требований, благодаря большому количеству различных плагинов

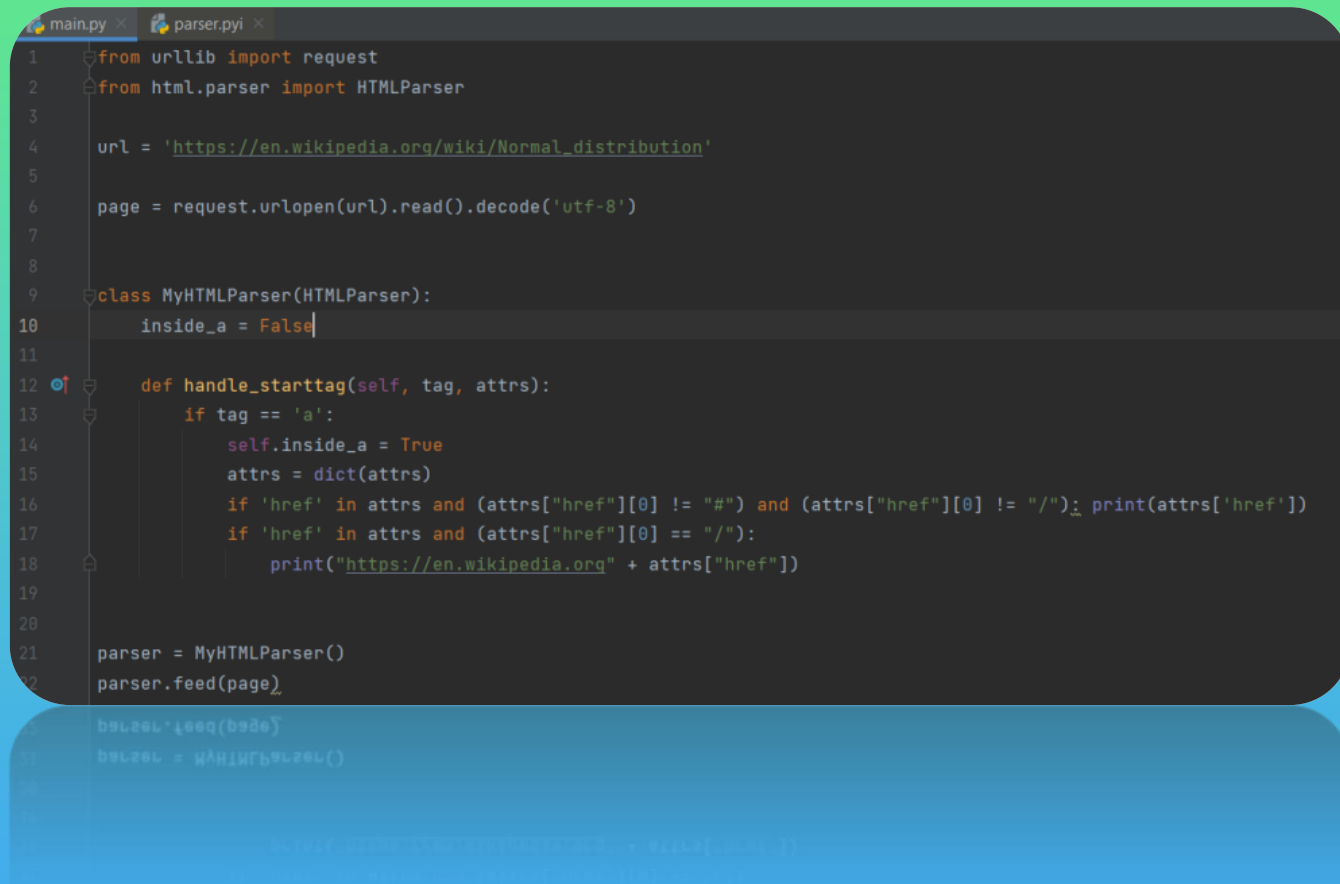
Запуск и компиляция кода

Как и во многих других IDE компиляция кода происходит сразу после его непосредственного запуска, без необходимости в каком либо стороннем аппаратном или пользовательском вмешательстве, что обеспечивает простоту использования данной IDE



Форматирование кода на PyCharm

- Автоматическая табуляция и расстановка пробелом в различных элементах кода, которая выставляется, в зависимости от используемых структур
- Подсветка синтаксиса и возможность просмотра начала и конца используемых структур
- Автозамена текста, в случае ошибочного написания кода на русском языке



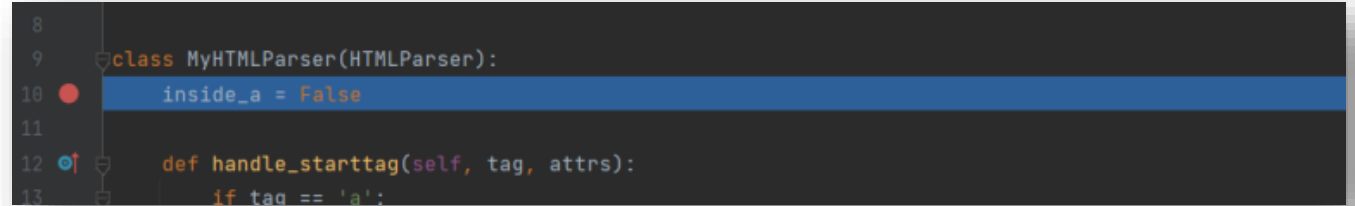
```

1  from urllib import request
2  from html.parser import HTMLParser
3
4  url = 'https://en.wikipedia.org/wiki/Normal_distribution'
5
6  page = request.urlopen(url).read().decode('utf-8')
7
8
9  class MyHTMLParser(HTMLParser):
10     inside_a = False
11
12     def handle_starttag(self, tag, attrs):
13         if tag == 'a':
14             self.inside_a = True
15             attrs = dict(attrs)
16             if 'href' in attrs and (attrs["href"][0] != "#" and attrs["href"][0] != "/"): print(attrs['href'])
17             if 'href' in attrs and attrs["href"][0] == "/":
18                 print("https://en.wikipedia.org" + attrs["href"])
19
20
21  parser = MyHTMLParser()
22  parser.feed(page)
  
```


Отладка кода

Отладка программы — это пошаговое выполнение вашей программы с целью обнаружения ошибок и проверки правильности кода.

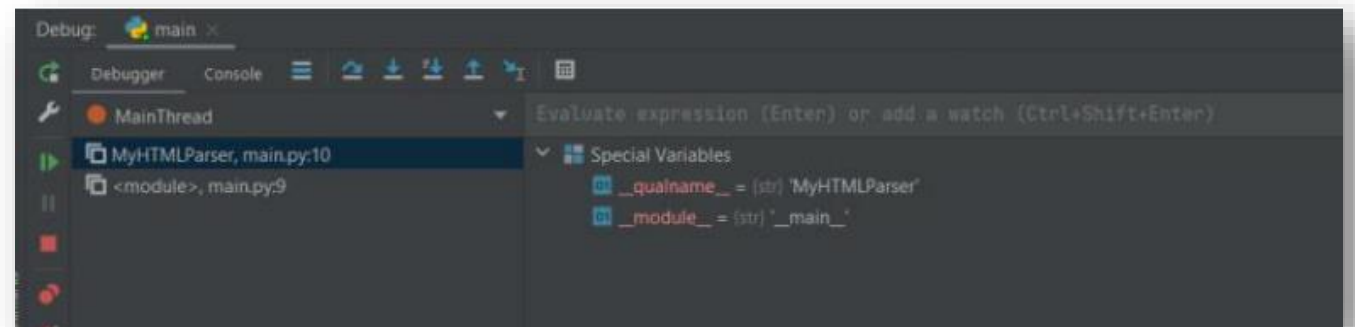
Чтобы начать отладку необходимо указать строку кода



```

8
9 class MyHTMLParser(HTMLParser):
10     inside_a = False
11
12     def handle_starttag(self, tag, attrs):
13         if tag == 'a':
  
```

Затем либо через кнопку запуска отладки либо через комбинацию клавиш shift+F9 запускается сам процесс отладки



Debug: main

Debugger Console

Main Thread

MyHTMLParser, main.py:10

<module>, main.py:9

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

Special Variables

- `__qualname__ = (str) 'MyHTMLParser'`
- `__module__ = (str) '__main__'`

Некоторой особенностью и большим плюсом является возможность работы отладчика с несколькими точками остановки, что позволяет работать с конкретным куском кода

```

11
12 def handle_starttag(self, tag, attrs):
13     if tag == 'a':
14         self.inside_a = True
15         attrs = dict(attrs)
16         if 'href' in attrs and (attrs["href"][0] != "#") and (attrs["href"][0] != "/"): print(attrs['href'])
17         if 'href' in attrs and (attrs["href"][0] == "/"):
18             print("https://en.wikipedia.org" + attrs["href"])
19

```

Кодирование

Умный редактор PyCharm предназначен для максимально продуктивной разработки на Python, JavaScript, CoffeeScript, TypeScript, CSS и популярных языках шаблонов.

1

Код читается легко

2

Варианты автодополнения
для ключевых слов, классов и
переменных

3

Используйте сочетания
клавиш

4

Многочисленные инспекции
проверяют код прямо в
режиме редактирования

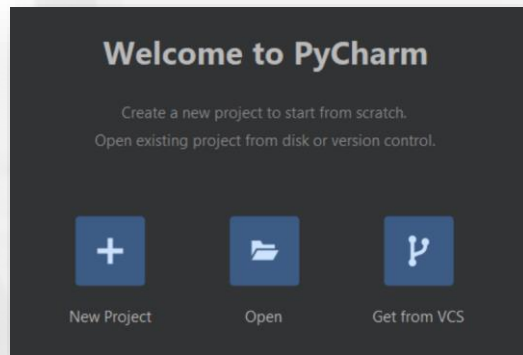
5

PyCharm проверяет код на
наличие дублирующихся
фрагментов

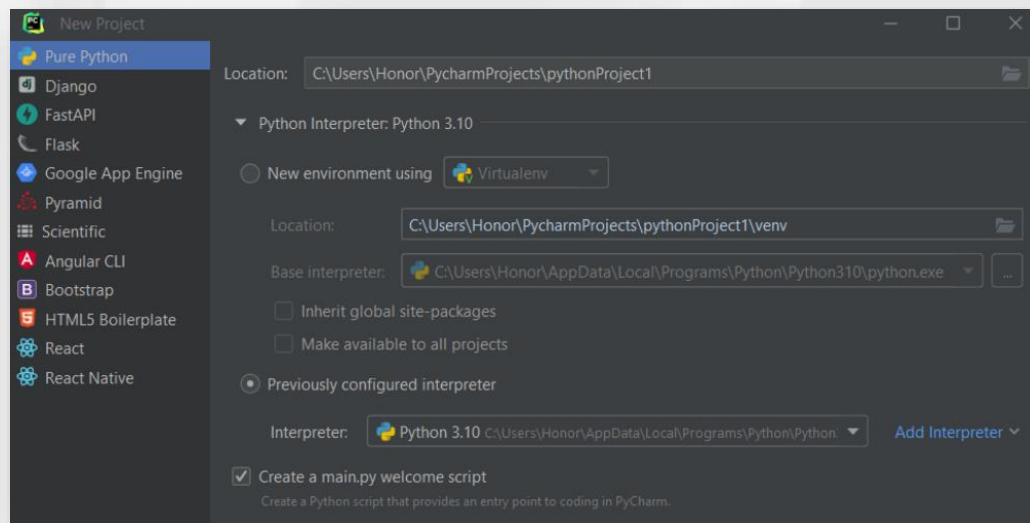
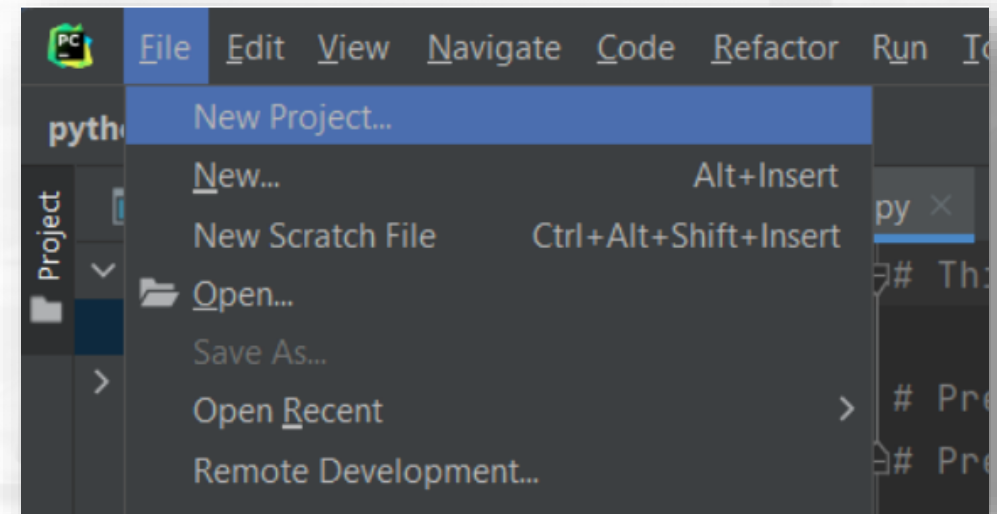
6

Редактируйте код,
написанный не на Python,
внутри строковых литералов

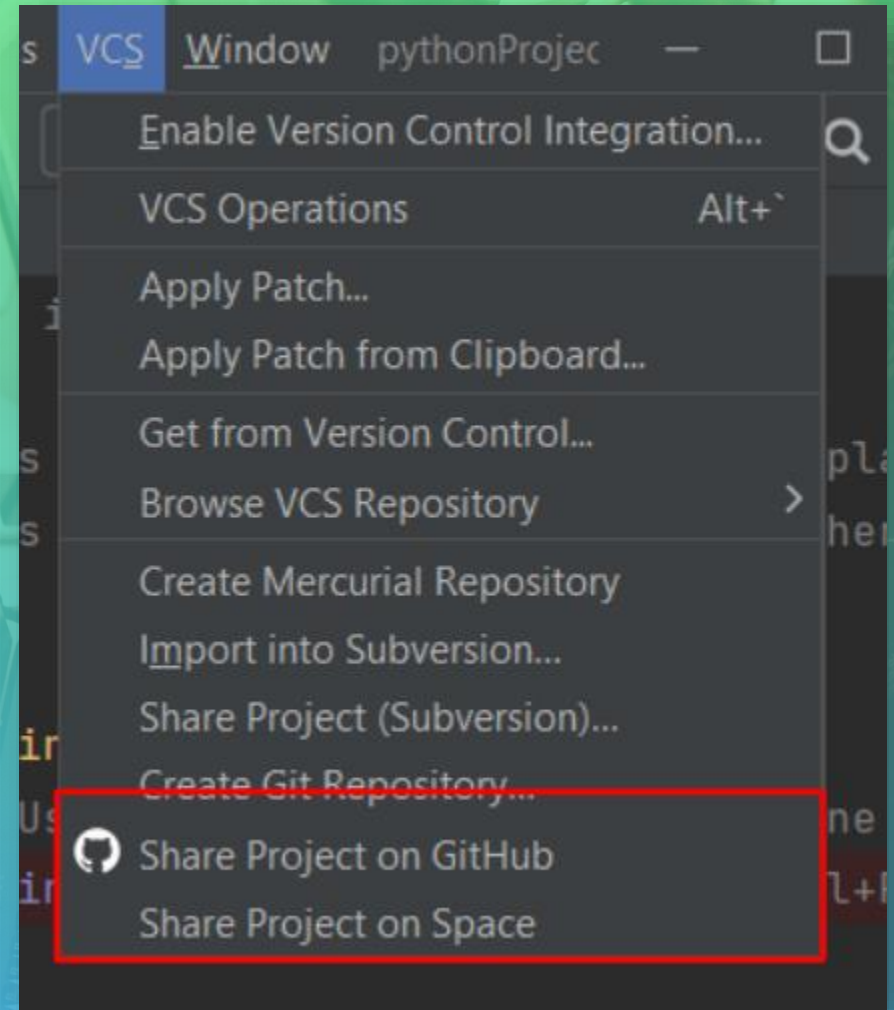
Создание проекта

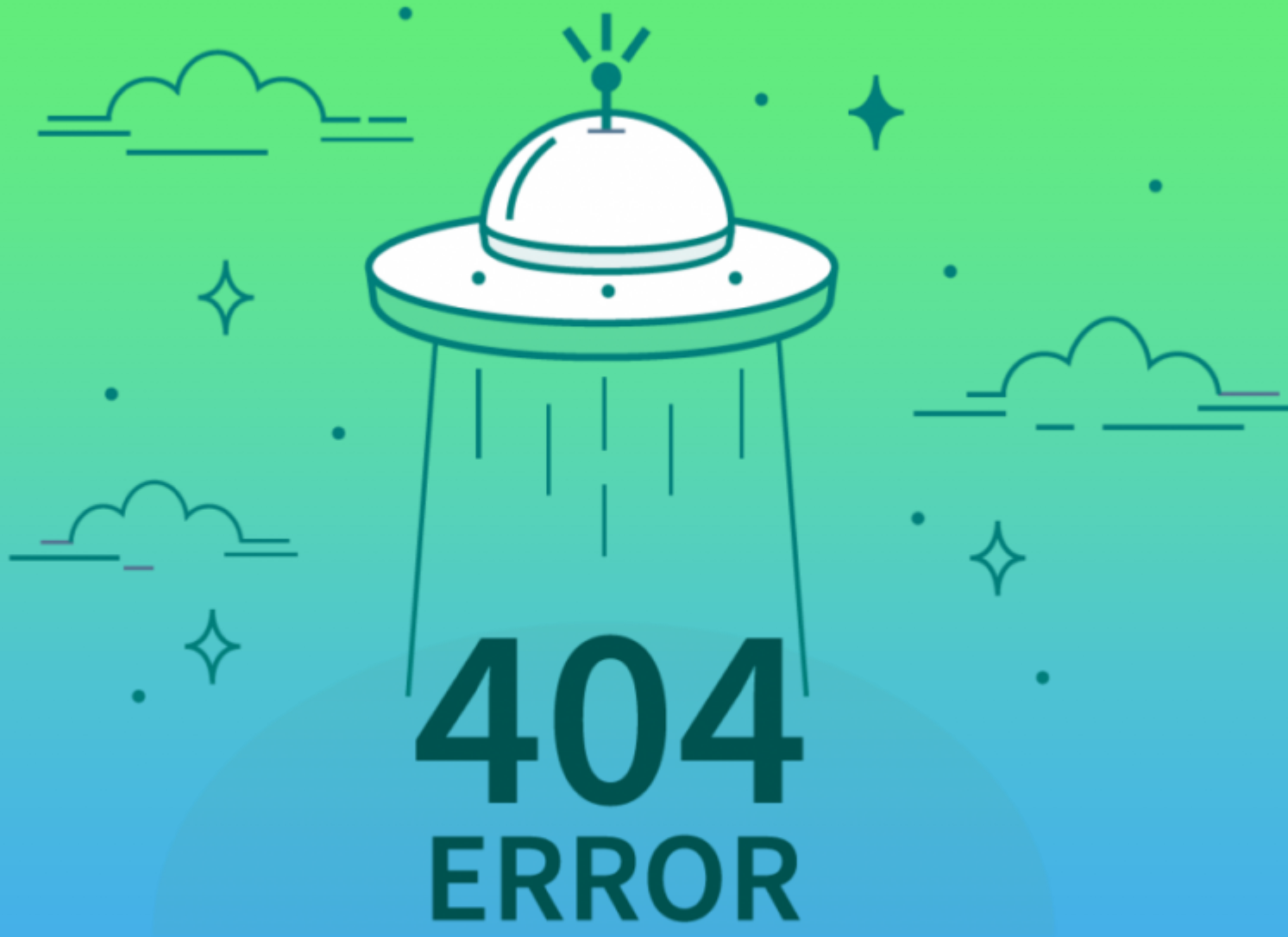


или



Публикация в репозитории





This page is lost