**开发环境**

**Anaconda**

Anaconda是一个方便的python包管理和环境管理软件，一般用来配置不同的项目环境。  
 我们常常会遇到这样的情况，正在做的项目A和项目B分别基于python2和python3，而电脑只能安装一个环境，这个时候Anaconda就派上了用场，它可以创建多个互不干扰的环境，分别运行不同版本的软件包，以达到兼容的目的。  
 Anaconda通过管理工具包、开发环境、Python版本，大大简化了你的工作流程。不仅可以方便地安装、更新、卸载工具包，而且安装时能自动安装相应的依赖包，同时还能使用不同的虚拟环境隔离不同要求的项目。

**数据库**

1. **Microsoft SQL Server**

SQL Server 是[Microsoft](https://baike.baidu.com/item/Microsoft/125917" \t "https://baike.baidu.com/item/Microsoft%20SQL%20Server/_blank) 公司推出的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "https://baike.baidu.com/item/Microsoft%20SQL%20Server/_blank)。具有使用[方便](https://baike.baidu.com/item/%E6%96%B9%E4%BE%BF)可伸缩性好与相关[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053)集成[程度](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%A6/70052)高等[优点](https://baike.baidu.com/item/%E4%BC%98%E7%82%B9/884953)，可跨越从[运行](https://baike.baidu.com/item/%E8%BF%90%E8%A1%8C/5480697)Microsoft Windows 98 的膝上型[电脑](https://baike.baidu.com/item/%E7%94%B5%E8%84%91/124859" \t "https://baike.baidu.com/item/Microsoft%20SQL%20Server/_blank)到运行Microsoft Windows 2012 的大型多处理器的服务器等多种平台使用。

Microsoft SQL Server 是一个全面的数据库平台，使用[集成](https://baike.baidu.com/item/%E9%9B%86%E6%88%90/1829613" \t "https://baike.baidu.com/item/Microsoft%20SQL%20Server/_blank)的[商业](https://baike.baidu.com/item/%E5%95%86%E4%B8%9A)智能 (BI)[工具](https://baike.baidu.com/item/%E5%B7%A5%E5%85%B7/81891)提供了企业级的数据管理。Microsoft SQL Server [数据库引擎](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E5%BC%95%E6%93%8E/1239260)为关系型数据和[结构](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84)化数据提供了更安全[可靠](https://baike.baidu.com/item/%E5%8F%AF%E9%9D%A0/3444426)的[存储](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8/1582924)功能，使您可以[构建](https://baike.baidu.com/item/%E6%9E%84%E5%BB%BA/8601024)和管理用于[业务](https://baike.baidu.com/item/%E4%B8%9A%E5%8A%A1/1176273)的高可用和高性能的数据[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445)。

1. **MySQL**

[MySQL](https://baike.baidu.com/item/MySQL/471251)是一种[开放源代码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81/114160" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)的关系型[数据库管理](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86)系统（RDBMS），使用最常用的数据库管理语言--[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80/10450182)（SQL）进行数据库管理。

MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的[需要](https://baike.baidu.com/item/%E9%9C%80%E8%A6%81/9109492" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)对其进行修改。

MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要[事务](https://baike.baidu.com/item/%E4%BA%8B%E5%8A%A1" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)化处理的情况下，MySQL是管理内容最好的选择。

**前端部分**

1. **编程语言**
2. **JavaScript**

选用JavaScript的原因如下：

（1）浏览器唯一规定的编程标准。  
 （2）各大浏览器公司支持。  
 （3）有着庞大的社区。  
 （4）有着庞大而且优秀的知名度高的简单易用的第三方库:jQuery, underscore等等。  
 （5）不用编译。  
 （6）编程效果更加可视化（尤其各种图形，动画效果的编程）。  
 （7）容易用代码直接交流。  
 （8）类和对象和事件创建十分简单。  
 （9）一切代码可见，所以网页有问题，作为程序员可以自行分析。

1. **HTML**

（1）网络标准统一，HTML5本身是由W3C推荐出来的。

（2）多设备、跨平台。

（3）即时更新。

（4）提高可用性和改进用户的友好体验。

（5）有几个新的标签，这将有助于开发人员定义重要的内容。

（6）可以给站点带来更多的多媒体元素(视频和音频)。

1. **备选框架**
2. **Vue**
3. Vue是一套用于构建用户界面的渐进式JavaScript框架；与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用；Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。
4. Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。
5. Vue.js的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件
6. Vue.js自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习，非常容易与其它库或已有项目整合。另一方面，在与相关工具和支持库一起使用时，Vue.js 也能完美地驱动复杂的单页应用。
7. Vue很容易上手，在学会[HTML](https://baike.baidu.com/item/HTML" \t "https://baike.baidu.com/item/Vue.js/_blank),[CSS](https://baike.baidu.com/item/CSS/5457),[JavaScript](https://baike.baidu.com/item/JavaScript)的前提下，自学一小段时间就能学会。
8. Vue非常灵活，简单小巧的核心，渐进式技术栈，足以应付任何规模的应用。
9. **React**
10. React的设计思想极其独特，属于革命性创新，性能出众，代码逻辑却非常简单。
11. React主要用于构建UI。你可以在React里传递多种类型的参数，如声明代码，帮助你渲染出UI、也可以是静态的HTML DOM元素、也可以传递动态变量、甚至是可交互的应用组件。
12. 声明式设计：React采用声明范式，可以轻松描述应用。
13. 高效：React通过对DOM的模拟，最大限度地减少与DOM的交互。
14. 灵活：React可以与已知的库或框架很好地配合。
15. **Angular**
16. Angular是一个[JavaScript](https://baike.baidu.com/item/JavaScript" \t "https://baike.baidu.com/item/AngularJS/_blank)框架。它是一个以 JavaScript 编写的库。它可通过 <script> 标签添加到[HTML](https://baike.baidu.com/item/HTML" \t "https://baike.baidu.com/item/AngularJS/_blank) 页面。
17. Angular通过指令扩展了HTML，且通过表达式绑定数据到 HTML。
18. Angular是以一个JavaScript文件形式发布的，可通过<script>标签添加到网页中。
19. Angular有着诸多特性，最为核心的是：MVC（Model–view–controller）、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。
20. **备选测试框架**
21. **Jasmine**

jasmine是一个用来编写Javascript测试的框架，它不依赖于任何其它的javascript框架。它有拥有灵巧而明确的语法可以让你轻松的编写测试代码。它有四个核心概念：分组，用例，期望，匹配，分别对应Jasmine的四种函数，如下：

1. Describe（string,function）：这个函数表示分组，也就是一组测试用例。
2. It（string，function）：这个函数表示测试用例。
3. Expect（expression）：表示期望expression这个表达式具有某个值或某种行为。
4. To\*\*\*（arg）：这个函数表示匹配。
5. **Jest**

Jest 是 Facebook 开源的一款 JS 单元测试框架，它也是 React 目前使用的单元测试框架。目前除了 Facebook 外，Twitter、Nytimes、Airbnb 也在使用 Jest。Jest 除了基本的断言和 Mock 功能外，还有快照测试、实时监控模式、覆盖度报告等实用功能。同时 Jest 几乎不需要做任何配置便可使用。

1. **Mocha**

Mocha是JavaScript的一种单元测试框架，既可以在浏览器环境下运行，也可以在Node.js环境下运行。  
 使用Mocha，我们就只需要专注于编写单元测试本身，然后，让Mocha去自动运行所有的测试，并给出测试结果。

Mocha主要有两个API，分别是分组和用例，如下：

1. Describe（string,function）：这个函数表示分组，也就是一组测试用例。第一个参数是测试套件的名称，第二个参数是一个实际执行的函数。

（2）It（string，function）：这个函数表示测试用例。第一个参数是测试用例的名称，第二个参数是一个实际执行的函数。

1. **备选包管理**
2. **Npm**

Npm 是 Node.js 默认的、以 JavaScript 编写的软件包管理系统，由三个独立的部分组成，分别为网站，注册表（registry），命令行工具（CLI）。

1. 网站：开发者查找包（package）、设置参数以及管理 npm 使用体验的主要途径。
2. 注册表：是一个巨大的数据库，保存了每个包（package）的信息。
3. 命令行工具：通过命令行或终端运行。开发者通过 CLI 与 npm 打交道。
4. **Component**
5. Component 是一个对客户端 JavaScript 包进行管理的工具，用于更好的构建 Web 应用。
6. 编写模块化组件。
7. 编写包含样式、图片、脚本等内容的组件。
8. 无需注册表发布或者帐号，使用github资料库。
9. **Bower**

Bower是一种包管理器，它可用于搜索、安装和卸载如JavaScript、HTML、CSS之类的网络资源。它是基于node.js的模块化思想来实现的。

**后端部分**

1. **编程语言**
2. **Java**

Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

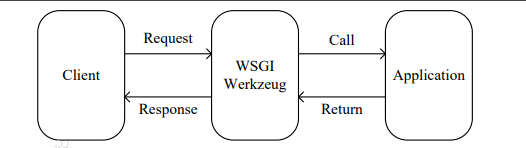
1. **Python**

Python是一种跨平台的计算机程序设计语言。是一种面向对象的动态类型语言，最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。在设计上Python坚持了清晰划一的风格，这使得Python成为一门易读、易维护，并且被大量用户所欢迎的、用途广泛的语言。Python是我们团队新学习的后端开发语言，我们对这语言更加感兴趣，所以我们选择用Python语言能更积极、更高效地完成后端的开发。这也对我们理解Python语言有极大的帮助。

1. **备选框架**
2. **Flask**

（1）Flask是一个轻量级的可定制框架，使用Python语言编写，较其他同类型框架更为灵活、轻便、安全且容易上手。它可以很好地结合MVC模式进行开发，开发人员分工合作，小型团队在短时间内就可以完成功能丰富的中小型网站或Web服务的实现。另外，Flask还有很强的定制性，用户可以根据自己的需求来添加相应的功能，在保持核心功能简单的同时实现功能的丰富与扩展，其强大的插件库可以让用户实现个性化的网站定制，开发出功能强大的网站。

（2）Flask的基本模式为在程序里将一个视图函数分配给一个URL，每当用户访问这个URL时，系统就会执行给该URL分配好的视图函数，获取函数的返回值并将其显示到浏览器上，其工作过程见图。



（3）IT运维的基本点为安全、稳定、高效，运维自动化的目的就是为了提高运维效率，Flask[9]开发快捷的特点正好符合运维的高效性需求。在项目迭代开发的过程中，所需要实现的运维功能以及扩展会逐渐增多，针对这一特点更是需要使用易扩展的Flask框架。另外，由于每个公司对运维的需求不同，所要实现的功能也必须有针对性地来设计，Flask可以很好地完成这个任务。

1. **Django**

（1）Django是高水准的Python编程语言驱动的一个开源模型．视图，控制器风格的Web应用程序框架，它起源于开源社区。Django已经成为web开发者的首选框架，是一个遵循 MVC 设计模式的框架。MVC是Model、View、Controller三个单词的简写，分别代表模型、视图、控制器。Django其实也是一个MTV 的设计模式。MTV是Model、Template、View三个单词的简写，分别代表模型、模版、视图 [3] 。但是在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。它们各自的职责如下：使用这种架构，程序员可以方便、快捷地创建高品质、易维护、数据库驱动的应用程序。这也正是OpenStack的Horizon组件采用这种架构进行设计的主要原因。另外，在Dj ango框架中，还包含许多功能强大的第三方插件，使得Django具有较强的可扩展性。



（2）Django框架具有开发效率块，功能特性完备的特点。像Python语言一样Django也是我们新学习的框架，以便于团队更好的协作与协调，Django与Python完美的结合，所以我们选择使用Django框架。

1. **备选包管理**
2. **DNF**

（1）DNF新一代的RPM软件包管理器。他首先出现在 Fedora 18 这个发行版中。而如今，他取代了YUM，正式成为 Fedora 22 的包管理器。

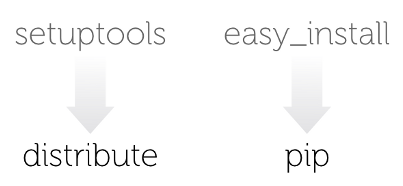
（2）DNF包管理器克服了YUM包管理器的一些瓶颈，提升了包括用户体验，内存占用，依赖分析，运行速度等多方面的内容。

1. **Yum**

Yum（全称为 Yellow dog Updater, Modified）是一个在Fedora和RedHat以及CentOS中的Shell前端软件包管理器。基于RPM包管理，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

1. **Pip**

pip 是一个Python包管理工具，主要是用于安装 PyPI 上的软件包，可以替代 easy\_install 工具。Python最让人的喜欢的就是它有丰富的类库和各种第三方的包，而对于这些包的下载、删除等管理操作，就要用到包管理工具，而Python的包管理工具有easey\_install、setuptools、pip、distribute，



可以看到setuptools将被distribute取代(虽然后来distribute又合并回setuptools中了，欲了解详情参见)。而pip则将要取代easy\_install。所以我们在这里简单说一下pip这个包管理工具，我们通过pip命令可以很方便的对各种包进行管理、下载、卸载、查询等操作。通过pip包管理工具下载第三方包，下面列出常用的pip命令。

1. **前后端参数传递框架**
2. **Ajax**

（1）Ajax 即“**A***synchronous* **J***avascript* ***A****nd* **X***ML*”（异步 JavaScript 和 XML），是指一种创建交互式[网页](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5/99347" \t "https://baike.baidu.com/item/ajax/_blank)应用的网页开发技术。

（2）Ajax = 异步 [JavaScript](https://baike.baidu.com/item/JavaScript) 和 XML 或者是 HTML（[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/6805073)的子集）。

（3）Ajax 是一种用于创建快速动态网页的技术。

（4）Ajax 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。

（5）通过在后台与服务器进行少量数据交换，Ajax 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

（6）传统的网页（不使用 Ajax）如果需要更新内容，必须重载整个网页页面。

1. **Axios**

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。它有以下特点

1. 从浏览器中创建 [XMLHttpRequests](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest)。
2. 从 node.js 创建 [http](http://nodejs.org/api/http.html) 请求。
3. 支持 [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) API。
4. 拦截请求和响应。
5. 转换请求数据和响应数据。
6. 取消请求。
7. 自动转换 JSON 数据。
8. 客户端支持防御 [XSRF](http://en.wikipedia.org/wiki/Cross-site_request_forgery)。
9. **Fetch**

Fetch是一种HTTP数据请求的方式，是XMLHttpRequest的一种替代方案。fetch不是ajax的进一步封装，而是原生js。Fetch函数就是原生js，没有使用XMLHttpRequest对象。它有以下特点：

1. 第一个参数是URL。  
    （2）第二个是可选参数，可以控制不同配置的 init 对象。  
    （3）使用了 [JavaScript Promises](https://links.jianshu.com/go?to=https://davidwalsh.name/promises) 来处理结果/回调。

**五、备选测试框架**

### 1. Robot Framework

作为最重要的Python测试框架之一，Robot Framework主要被用在测试驱动(test-driven)类型的开发与验收中。虽然是由Python开发而来，但是它也可以在基于.Net的IronPython和基于Java的Jython上运行。同时，作为一个Python框架，Robot还能够兼容诸如Windows、MacOS、以及Linux等平台。

**优点**

* 通过使用关键字驱动测试(keyword-driven-test)的方法，它能够帮助测试人员轻松地创建具有可读性的测试用例，进而简化了整个自动化的过程。
* 用户能够轻松地测试数据中的语法。
* 因为是由各种通用工具和测试库所组成，所以它不但拥有庞大的生态系统，而且可以在单独的项目中使用其各个元素。
* 由于拥有许多类型的API，因此该框架具有高度的可扩展性。
* 虽然并非内置功能，但是Robot可以通过Selenium Grid来帮助用户运行各种并行测试。

**缺点**

* 虽然有着上述各种便利，但是Robot Framework在创建自定义的HTML报告方面比较繁琐。您顶多能用它来生成xUnit格式的简短报告。
* 另外，Robot Framework在并行测试方面也并不见长。

### 2. Pytest

适用于多种软件测试的Pytest，是另一个Python类型的自动化测试框架。凭借着其开源和易学的特点，该工具经常被QA(质量分析)团队、开发团队、个人团队、以及各种开源项目所使用。

**优点**

* 过去，开发人员必须将他们的测试包含在各种大型的类中。如今，Pytest可以让用户写出更为紧凑的测试套件。
* 其他的测试工具会要求开发或测试人员使用调试器，或以日志检查的方式检测某个特定值的来源。而用户在用Pytest来编写测试用例时，完全可以将所有的值都存储在测试用例之中，直到再获悉有哪些值失败了、以及哪些值得到了断言。
* 由于涉及到的样板代码(boilerplate code)并不多，因此用户能够容易地编写和理解各种测试。
* 测试夹具(Fixture)函数常被用来向测试函数添加某个参数，并返回不同的值。在Pytest中，您可以通过使用一个夹具来模块化另外一个。同时，用户可以使用多个夹具，在无需重写测试用例的情况下，将测试覆盖到所有参数的组合。
* Pytest开发人员通过发布各种实用插件，来保持框架的可扩展性。例如：pytest-xdist可以在不使用其他测试器的情况下，被用于执行并行测试。同时，单元测试也可以在无需复制任何代码的情况下，实现参数化。
* 通过为开发人员提供各种特殊的例程，它能够使得测试用例的编写更为简单、更不易出错，同时代码也会变得更短、更易被理解。

**缺点**

* 上述提及的特殊例程，也意味着用户必须放弃一定的兼容性。虽然方便了用户编写测试用例，但是这些用例却无法与任何其他的测试框架一起被使用。

### 3.UnitTest/PyUnit

受到了JUnit启发的UnitTest/PyUnit，也是一种标准化的针对单元测试的Python类自动化测试框架。它的基类TestCase提供了各种断言方法、以及所有清理和设置的例程。因此，TestCase子类中的每一种方法都是以“test”作为名词前缀，以标识它们能够被作为测试用例所运行。用户可以使用load方法和TestSuite类来分组、并加载各种测试。当然，您也可以通过联合使用，来构建自定义的测试运行器。正如我们使用Junit去测试Selenium那样，UnitTest也会用到UnitTest-sml-reporting、并能生成各种XML类型的报告。

**优点**

* 开发人员并不需要安装任何其他的模块。
* UnitTest是xUnit的衍生产品，其工作原理与其他xUnit框架十分类似。因此对于那些没有过硬Python背景的人来说，也能很快地上手。
* 用户能够以更为简单的方式运行单个测试用例。您只需在终端上预定好名称，该框架便可灵活地执行各种用例的测试，并产生精炼的输出。
* 它能够在几毫秒内生成各种测试报告。

**缺点**

* 虽然该框架常用snake\_case来命名各种Python代码，但是由于它源自Junit，因此仍保留了一些传统的camelCase命名方法。这往往会让人产生混淆。
* 由于它过多地支持了抽象方法，因此造成了测试代码的目的有时不够清晰。
* 需要大量的样板代码。

**代码质量监测工具**

1. **PMD**

PMD的核心是JavaCC解析器生成器。PMD结合运用JavaCC和EBNF（扩展巴科斯-诺尔范式，ExtendedBackus-Naur Formal）语法，再加上JJTree，把Java源代码解析成抽象语法树（AST，Abstract SyntaxTree）。不过，为了让解析器承认这些普通的文本是合法的Java代码，它们必须符合某种特定的结构要求。这种结构可以用一种称为EBNF的句法元语言表示，通常称为“语法”（Grammar）。JavaCC根据语法要求生成解析器，这个解析器就可以用于解析用Java编程语言编写的程序。

实际运行中的PMD还要经过JJTree的一次转换。JJTree是一个JavaCC的插件，通过AST扩充JavaCC生成的解析器。AST是一个Java符号流之上的语义层。有了JJTree，语法分析的结果不再是“System, ., out, ., .println”之类的符号序列，而是一个由对象构成的树型层次结构。

1. **FindingBugs**

FindBugs是一个java byte code静态分析工具，检测出Java程序中上百种潜在的不同类型的错误。不注重style及format，注重检测真正的bug及潜在的性能问题，尤其注意尽可能抑制误检测(false positives)的发生。除了作为单独的组件安装，还可以集成IDE，以及maven。其检测输出结果可以是XML或文本形式的。支持自定义配置检查规则（做哪些检查，不做哪些检查），校验规则（用户自定义特定的bug模式需要继承它的接口，编写自己的校验类）。

1. **Checkstyle**

CheckStyle检查代码是否符合制定的规范。CheckStyle 检查是基于源码的，所以不需要编译，执行速度快。有以下特点：

（1）它可以有效的帮助我们检视代码以便更好的遵循代码编写标准，特别适用于小组开发时彼此间的样式规范和统一。

（2）Checkstyle提供高可配置性，以便适用于各种代码规范，所以除了使用它提供的几种常见标准之外，你也可以定制自己的标准。

（3）Checkstyle提供支持大多数常见IDE的插件，大部分插件中就含有最新的Checkstyle。

（4）Checkstyle可以检查代码的很多方面，从传统观点看，它主要是用来检查代码层面的，自从第三版以后，它的内部架构作了重大改变，很多其它意图的检测加了进来，现在Checkstyle可以检查像类设计的问题，重复代码，如锁的双重检查的bug模式。

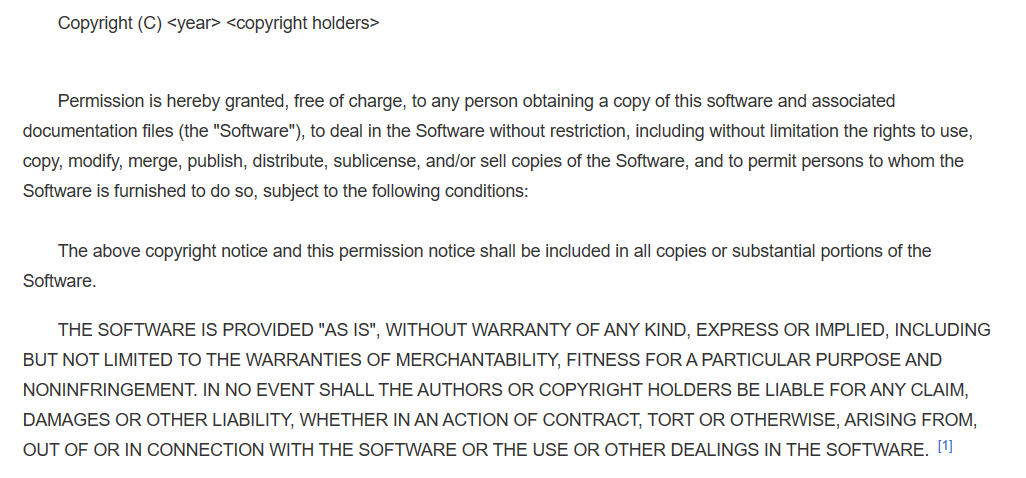
**4. Pylint**

Pylint 是一个 Python 代码分析工具，它分析 Python 代码中的错误，查找不符合代码风格标准和有潜在问题的代码。  
 Pylint 除了平常代码分析工具的作用之外，它提供了更多的功能：如检查一行代码的长度，变量名是否符合命名标准，一个声明过的接口是否被真正实现等等。  
 Pylint 的一个很大的好处是它的高可配置性，高可定制性，并且可以很容易写小插件来添加功能。

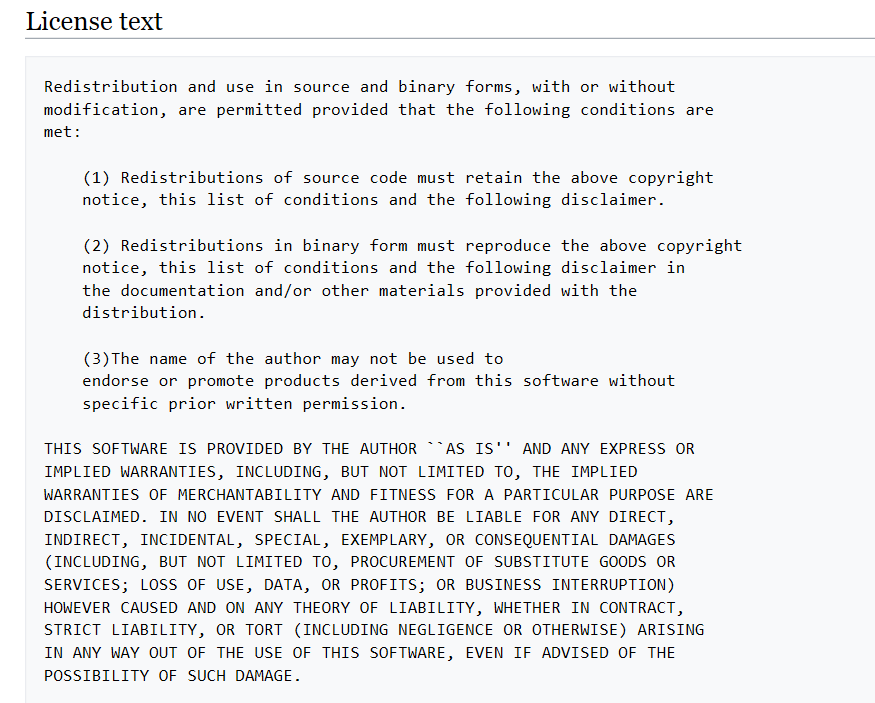
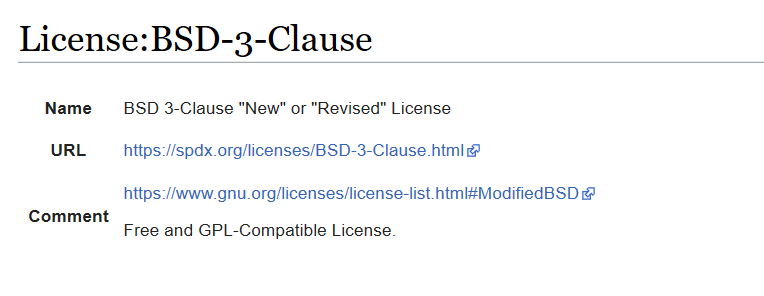
如果运行两次 Pylint，它会同时显示出当前和上次的运行结果，从而可以看出代码质量是否得到了改进。 目前在 eclipse 的 pydev 插件中也集成了 Pylint。

**开源协议**

1. **Vue**



1. **Django**



1. **Pip**



1. **Axios**



**最终选型**

1. **开发环境**

Anaconda3 -version4.6.11

1. **数据库**

MySQL 8.0.16

1. **前端部分**
2. **编程语言**

HTML+JavaScript

1. **框架**

Vue 2.9.6

1. **测试框架**

Jest23

1. **包管理**

Npm 6.13.4

1. **后端部分**
2. **编程语言**

Python 3.7

1. **框架**

Django 3.0.1

1. **包管理**

Pip 19.0.3

1. **前后端参数传递框架**

Axios 0.19.0

**5. 代码测试框架**

Pytest 5.3.2

1. **代码质量监测工具**

Pylint 2.4.4