

Django is a Python based web framework which allows clean and pragmatic design as well as rapid development. The framework is designed to enable the coders to code without worrying about the security of the code as it guides the coders through securing the vulnerabilities such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Django also has a user authentication system which can be customised to allow a secure way for data to be transferred[?]. Not only does Django provide security, it also allows efficient and easy coding as it follows the model, controller view system which avoids repeated code. With Python being the main language, numerous frameworks are available to make working with Django even easier.[?]

### 0.0.1 Django vs Laravel

Django is Python based as stated above but Laravel[?] on the other hand is a PHP based web framework. Django follows the Model View template similar to Ruby on Rails but Laravel follows Object Oriented Programming as well as the Model View Template. In terms of website security, Django takes it extremely seriously and helps developers avoid the common mistakes which lead to the website having vulnerabilities while Laravel also has a guide to avoid making such mistakes but its contents are not as informative and detailed as Django. You would need to manually test each of the security features unlike Django where the system makes sure that all security protocols are in place with latest updates. Django is naturally very fast as it uses Python which is known for its speed and processing. Django beats Laravels speed in all; Plaintext, JSON and SQLite Fetch[?] as shown in Figure 1.

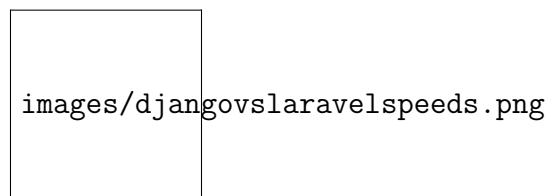


Figure 1: Django vs Laravel Speeds 2016 - CabotSolutions[?]

### 0.0.2 Django vs Ruby on Rails

Both are popular web frameworks, Django more than Ruby on Rails for professional developers. They are open-source which allows all code to be customised in any way. Ruby on Rails allows the use of gems which are designed by other users, these come in handy when programming personal or individual projects however with professional projects licensing is a good idea to allow the code to be usable in the future[?]. Django on the other hand has user coded packages which can be used with the application being created. An example would be *django-pgcrypto*[?] which allows the encryption of database entries. Based on JetBrains research[?][?] we know that Python is a really popular language compared to Ruby and it is highly used for Data Analysis and Web

Development among various other things while developers use a mixture Ruby Versions which makes future updates to the code harder. Table 1 shows the differences between Django and Ruby on Rails

	Django	Ruby on Rails
Pros	<ul style="list-style-type: none"> <li>- Python is versatile</li> <li>- Fast</li> <li>- Caching System</li> <li>- Data Analysis</li> <li>- Great Security and Authentication</li> </ul>	<ul style="list-style-type: none"> <li>- Flexible</li> <li>- Large Community</li> <li>- Gems Available</li> <li>- Easy Migration</li> </ul>
Cons	<ul style="list-style-type: none"> <li>- Hard to debug</li> <li>- Monolithic architecture</li> </ul>	<ul style="list-style-type: none"> <li>- Bloated</li> <li>- No Data Analysis</li> <li>- Very explicit and inelegant to read</li> </ul>

Table 1: Django vs Ruby on Rails Pros and Cons [?]

### 0.0.3 Django vs NodeJS

NodeJS[?] uses C, C++ and JavaScript to build the web application. The use of JavaScript mainly is extremely good when it comes to the design and feel of the web system but not when it comes to security. NodeJS can be used to produce a progressive web application which will make the website feel and work like a smart application even when there is no network as well as work perfectly on larger screens too. However, being that security is an important requirement for our system, Django would win over NodeJS[?]