

下一步看什麼

如果你已經讀完了 [介紹文件](#)，且對繼續使用 Django 感興趣。不過，你讀的是整體文件的精簡版（實際上，如果你逐字閱讀了此文件，你已經閱讀了整體文件的 5%）。

那麼下一步做什麼？

不錯，我們已經透過邊學邊做成為了 Django 的死忠粉了。此時，你應該已經知道如何開始你自己的工程，且會到處搜索其它文件了。想知道更多技巧的話，請回到文件頁。

為了使 Django 的文件達到更易使用，更清晰，更加全面的目的，我們付出了巨大的努力。本文件的剩餘部分將更詳細地介紹此文件的工作方式，以便您可以充分利用它。

（是的，這就是傳說中的說明文件的說明文件。請放心，我們不打算寫一篇關於如何閱讀此文件的文件。）

查找文件

Django 有 [許多](#) 的文件——差不多有 450000 字（英文單詞），所以查找你需要的文件可能需要點技巧。對此，有幾個不錯的去處，[搜索頁面](#) 和 [索引](#)。

或者你可以只是四處看看！

文件是如何組成

Django 的主要文件以“塊”的形式劃分，用於滿足不同的需求：

- The [介紹文件](#) 是專為 Django 初學者或 Web 初學者設計的。它不會做任何的深入，只是以高度概括的方式介紹了如果以 Django “風格”開發應用的方式。
- 另一方面，The [主題指南](#) 則深入介紹 Django 的各個部分。那裡有更多完整的關於 Django 的 [模板系統](#)、[模板引擎](#)、[表單框架](#) 和其它東西的資訊。

這可能是你會花費你大部分時間的地方。如果你詳細閱讀了這些指南，你就能了解幾乎所有關於 Django 的知識。

- Web 開發通常是廣而不深的——問題通常跨越多個領域。我們已經寫了一系欄的文件 [how-to 指引](#) 用於回答常見的“為什麼我會……？”系欄問題。在這裡，你可以找到關於 [透過 Django 產生 PDF](#)、[定義自定義模板標籤](#) 的文件，當然，還有其它的文件。

關於常見問題的回答可參見 [FAQ](#)。

- 這個引導和怎麼做的文件不會覆蓋 Django 中每個可用的類，函數和方法——在你想學的時候，你會發現實在是太多了。作為取代，每個類，函數，方法和模組的細節在 [參考](#) 中介紹。那是你未來查找某個函數的細節或其它你需要的東西的地方。
- 如果你對部署一個公用的工程感興趣，我們有介紹各種部署設置的文件 [幾個指引](#) 和介紹你幾個你需要了解的東西的文件 [部署清單](#)。
- 最後，這裡有一些與大部分開發者無關的“專業”文件。套件含 [發布說明](#) 和 [內部文件](#)，用於向那些想向 Django 提交程式的專業人士。還有一份文件 [不適合放到其它地方的一點內容](#)。

這個文件是如何更新的

就像 Django 的源碼每天都被更新和提升一樣，我們的文件也會持續優化。我們因為以下幾個原因優化文件：

- 更正內容，如更正語法/拼寫錯誤。
- 向已有的需要被擴展的某個章節增加介紹資訊或例子。
- 記錄尚未記錄的 Django 功能。（這些功能的欄表正在縮小，但仍然存在。）
- 在新特性被增加時，或 Django 的 API 或行為有變化時，會增加新文件。

Django 的文件以和它的程式一樣，以程式版本管理系統方式進行管理。它被儲存在 git 倉庫的 [docs](#) 目錄內。每份在線文件都是倉庫內的一份獨立文本文件。

從哪裡獲取這個

你可以以好幾種形式閱讀 Django 的文件。他們按照優先順序排欄：

在網絡上

Django 最新的在線版文件位於 <https://docs.djangoproject.com/en/dev/>。這些網頁由 Django 的源碼控制系統中的純文本文件自動產生。這意味著它們展示了 Django “最新最好” 的修改——它們套件含最新的更正和補充，並討論了最新的 Django 功能，這些功能只可供 Django 開發版的用戶使用。（參見以下關於“不同版本之間的差異”的介紹。）

為提高文件質量，你可以選擇在 [工單系統](#) 中提交變更，修正以及建議，為此我們將十分欣喜。Django 的開發者們會積極的監控工單系統，並使用你的反饋為大家改善文件。

值得一提的是，工單(ticket)應該明確地關聯到文件，而不是詢問籠統的技術支援問題。如果你需要針對你的 Django 設定尋求協助，嘗試聯繫 [django-users](#) 郵件組 或者 [#django IRC channel](#)。

純文本形式

離線閱讀，或僅僅是為了方便，你可以閱讀 Djano 文件的純文本形式。

如果你正在使用的是 Django 的某個正式發布版，注意有一個程式壓縮套件，套件含了 **docs/** 目錄，內含這個版本的完整文件。

如果你正在使用開發版的 Django（又名“trunk”），注意目錄 **docs/** 套件含所有的文件。你可以透過 git 獲取最新的修改。

一種沒啥技術含量的利用純文本文件的方式是使用 Unix 的 **grep** 工具在文件中全域中搜索一個短語。舉個例子，接下來會向你展示 Django 文件中所有提到這個特定短語 “max_length” 的地方：



```
$ grep -r max_length /path/to/django/docs/
```

以本地網頁形式閱讀

經過幾步操作，你可以取得一份網頁文件的拷貝：

- Django 文件使用了一個叫做 [Sphinx](#) 的系統將純文本轉換為網頁。你可以透過 Sphinx 的官方網站或 **pip** 來下載和安裝它：



```
$ python -m pip install Sphinx
```

- 接著，使用其中的 **Makefile** 工具將文件轉換為網頁：

```
$ cd path/to/django/docs
$ make html
```

你需要為此安裝 [GNU Make](#) 工具。

如果你是 Windows 系統，你應該使用其中的批處理文件：

```
cd path\to\django\docs
make.bat html
```

- 這個 HTML 文件將會被放置在 `docs/_build/html`。

版本之間的差異

The text documentation in the master branch of the Git repository contains the "latest and greatest" changes and additions. These changes include documentation of new features targeted for Django's next feature release. For that reason, it's worth pointing out our policy to highlight recent changes and additions to Django.

我們遵循以下原則：

- <https://docs.djangoproject.com/en/dev/> 上的開發文件來自主分支。這些文件對應於官方最新發布的特性，並且套件含所有自當時起，我們在框架中增加或修改的功能。
- 當我們為 Django 的開發版本增加功能時，我們會在相同的 Git commit 事務中更新文件。
- 為區分文件中修改或新增的內容，我們使用短語："New in Django Development version" 來表示其屬於還未發布的開發版，使用 "New in version X.Y" 來表示其屬於已經發布的某個版本。
- 文件的修正和提升可能只會提交到最新的一個發布版本，這取決於提交者，然而，一旦一個 Django 的版本處於 不在維護欄表 中，這個版本對應的文件將不再更新。
- The main documentation Web page includes links to documentation for previous versions. Be sure you are using the version of the docs corresponding to the version of Django you are using!

[◀ 進階指南：如何編寫可重程式](#)

[編寫你的第一個 Django 補丁 ▶](#)
