为了熟悉Three.js，我们需要花一些时间来了解它的目录结构、文档和示例。这里面包含的东西很多，了解一下对以后学习的用处很大。

~build/

Three.js的完整项目代码输出目录，包括未压缩版本和压缩版本。Three.js使用Google Closure进行编译发布，编译好的输出文件包含完整的Three.js库代码，它由不同代码目录下的源代码编译而成。如果你不熟悉Closure并且希望了解更多关于它的信息，请访问<https://developers.google.com/closure/compiler/>。注意，你并不需要再次执行Three.js的编译发布过程。如果你并不想处理编译发布相关的事情，那么只需直接使用build目录中的编译好的three.js或three.min.js。

~docs/

这个目录包含以HTML格式编写的完整API文档。虽然这份文档并不详细，但至少它为熟悉整个库提供了一份很好的设计概述。

~editor/

Three.js团队开发了一个用于创建3D场景的编辑系统。

~examples/

这个目录包含几百个示例。这些示例覆盖了Three.js的各种特性和效果，包括Canvas、css和WebGL渲染器的使用范例。其中一些是简单的“技术演示”，用于展现一些典型的特性。另一些则是综合运用了各类特性的精心大作。请花费一些时间来浏览每一个示例以及阅读它们的代码，这是你熟悉并领略Three.js强大功能的最佳途径。

~src/

整个库的代码文件目录。这是一个复杂的文件树结构，大致上可以分为两个部分：核心和扩展。核心部分包括主要的特性集合，这是Three.js可运行的基础。倘若这部分缺失，你将无法使用Three.js来渲染场景。扩展部分包含大量有用的特性，包括：内置几何形状，如立方体、球体和圆柱；动画相关的通用函数；以及图片加载类。你完全可以基于Three.js核心自行构建这些东西，但你显然并不愿意这么做。因此，虽然扩展部分，但它们默认包含在编译输出的完整库文件中。

~utils/

这个目录包含各种工具，包括用于编译压缩和非压缩版本库文件的Google Closure脚本，用于将各种3D文件格式转换为Three.js JSON和二进制文件格式的文件转换器，还提供了用于从建模软件导出Three.js格式文件的插件，如Blender和Maya