

第 1 章

開発環境の準備

プログラミングを始めるにあたり、必要となる開発環境を整えていきます。エディタとして「Atom」を、ブラウザとして「Firefox」をご紹介します。

プログラミングを行うための環境整備を行っていきましょう。整える道具は二つ、「エディタ」と「ブラウザ」です。

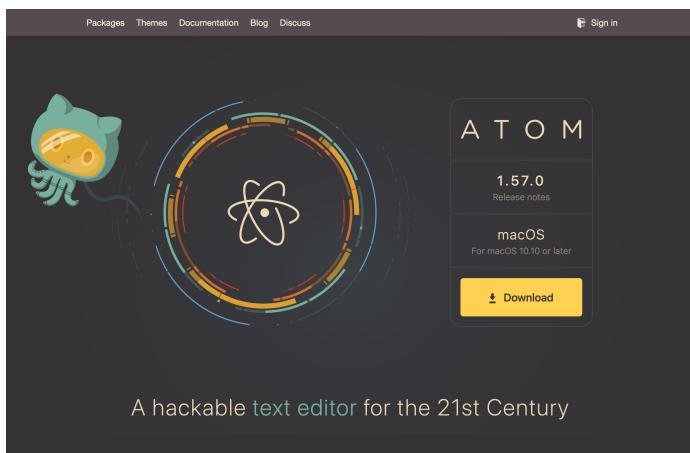
プログラミングを行うために必要となるエディタを準備しましょう。エディタとは、プログラミングが快適に行えるよう様々な支援機能を備えたコーディングのためのソフトウェアです。さまざまなエディタが提供されていますが、ここでは、Atom をご紹介いたします。

「ブラウザ」は、ウェブサイトを閲覧するためのソフトウェアです。じゃんけんゲームは、ウェブアプリとして開発していきますので、コーディング結果を確認するためにブラウザが必要です。さまざまなブラウザが提供されていますが、ここでは、Firefox をご紹介いたします。

1.1

21世紀の高性能エディタ Atom

Atomは、[公式サイト^{*1}](#)より、ダウンロードできます。黄色の「Download」ボタンを押すとダウンロードが始まります。落としたファイルをダブルクリックして、インストールしてください。



▲図1.1: Atom公式サイト

♣ お勧めプラグイン

Atomはそのままでも十分高機能に使えるGitHub謹製のテキストエディタです。そして有志の方により、多くの「プラグイン」が提供されています。プラグインとは、差し込む、差込口などの意味を持つ英単語で、ITの分野では、ソフトウェアに機能を追加する小さなプログラムのことを指します。^{*2}

プラグインのことを、Atomでは、「パッケージ」と呼んでいます。たくさんのパッケージがありますが、いくつかお勧めをご紹介いたします。

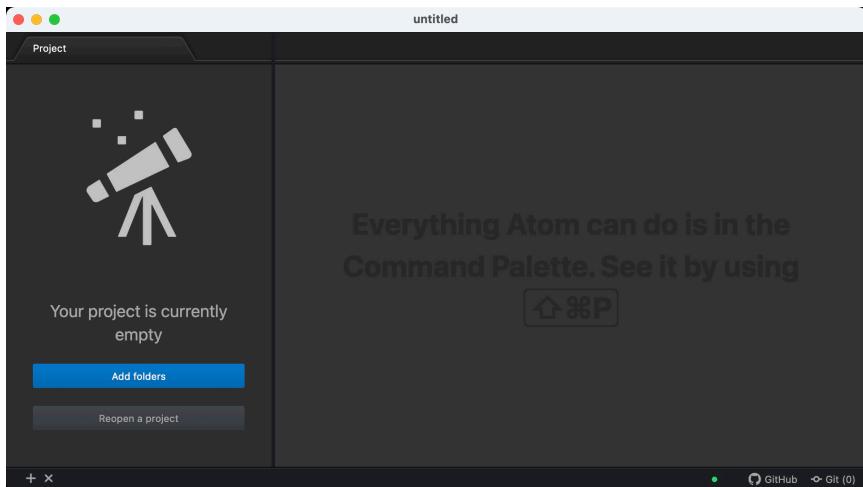
^{*1} <https://atom.io>

^{*2} 出典：IT用語辞典

1. メニューバーや設定画面などを日本語化 [japanese-menu^{*3}](#)
2. カラーピッカー [color-picker^{*4}](#)
3. CSS の色指定 (#ffffff) を背景色に表示 [pigments^{*5}](#)
4. ファイルの種類に応じたアイコンを表示 [file-icons^{*6}](#)
5. カーソルがある列を強調表示 [highlight-column^{*7}](#)
6. 選択箇所を強調表示 [selection-highlight^{*8}](#)
7. AI によるコード補完機能 [tabnine^{*9}](#)

それでは、インストールしていきましょう。

1. Atom を起動します。



▲図 1.2: 起動直後

^{*3} <https://atom.io/packages/japanese-menu>

^{*4} <https://atom.io/packages/color-picker>

^{*5} <https://atom.io/packages/pigments>

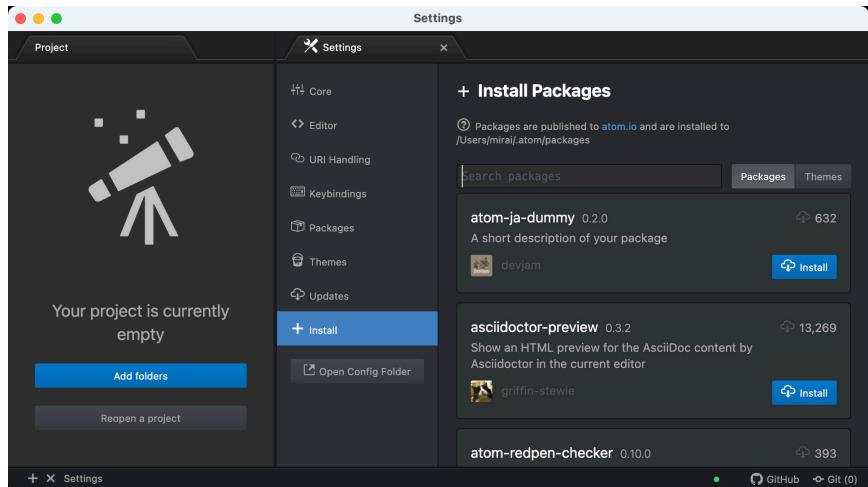
^{*6} <https://atom.io/packages/file-icons>

^{*7} <https://atom.io/packages/highlight-column>

^{*8} <https://atom.io/packages/selection-highlight>

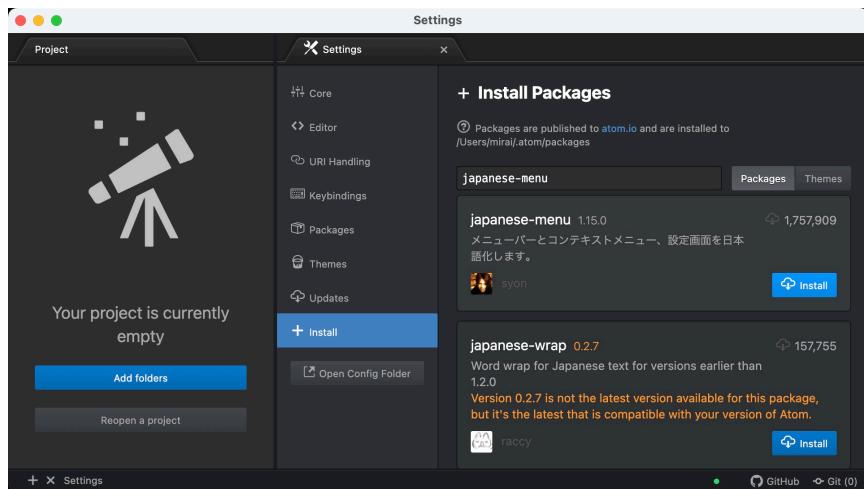
^{*9} <https://atom.io/packages/tabnine>

- File メニューから Preferences をクリックし、環境設定画面を開きます。様々な項目を設定することができます。パッケージをインストールするために、中ほどのメニュー内の「Install」ボタンを押します。



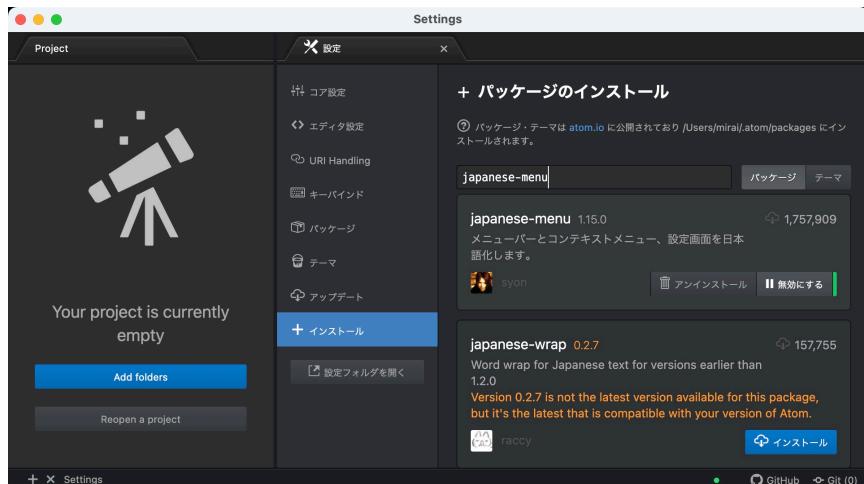
▲図 1.3: 設定画面

- 右上の Install Packages の下に入力枠がありますので、インストールしたいパッケージ名を入力します。ここでは、`japanese-menu`と入力します。`japanese-menu`というパッケージが表示されますので、「Install」ボタンを押します。



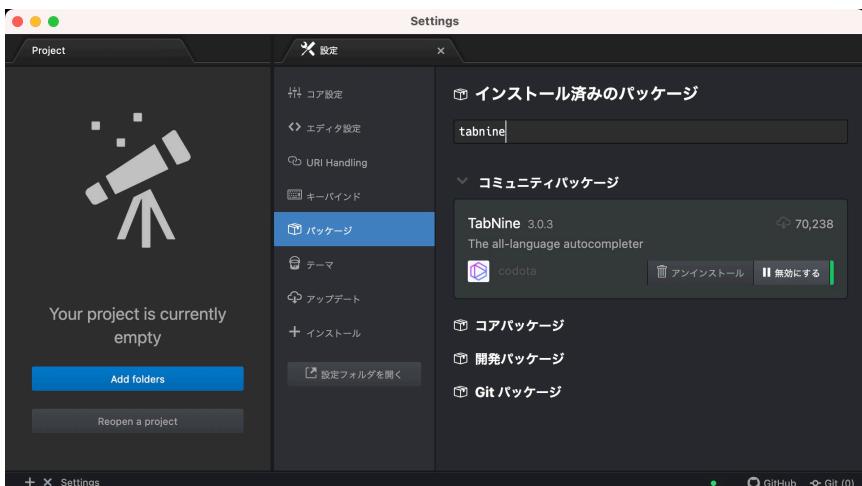
▲図 1.4: パッケージ名を入力

4. インストールが完了しました。今まで英語だった説明文が、日本語になっています。



▲図 1.5: インストール完了

- 一度インストールしたパッケージを削除したいときには、中ほどのメニュー内の「パッケージ」ボタンを押します。右上の インストール済みのパッケージ の下に入力枠がありますので、アンインストールや無効にしたいパッケージ名を入力します。例えば `tabnine` と入力すると、「アンインストール」ボタンや「無効にする」ボタンが表示されます。



▲ 図 1.6: パッケージの削除や無効化の方法

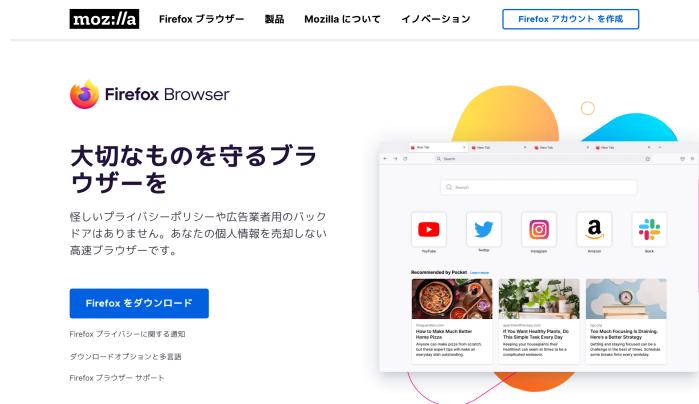
この他にも、いくつかお勧めのパッケージを見繕いましたので、Rails での Web アプリ開発に愛用している Atom プラグイン^{*10} にご紹介しています。いろいろなパッケージをインストールして自分の使いやすいエディタに育ててやって下さい。

1.2 セキュリティ重視のブラウザ Firefox

Firefox は、Mosaic, Netscape の血を受け継ぐ、ブラウザです。CSS グリッ

*10 https://zenn.dev/atelier_mirai/articles/c3ed79af5ba395

ドの確認がし易く、丁寧なコメントの開発者ツールが特徴です。Firefox は、[公式サイト](#)^{*11} より、ダウンロードできます。青色の「Firefox をダウンロード」ボタンを押すとダウンロードが始まります。落としたファイルをダブルクリックして、インストールしてください。



▲図 1.7: Firefox 公式サイト

1.3 基本的な作り方

Atom と Firefox がインストールできましたので、プログラムの基本的な作り方をご案内します。

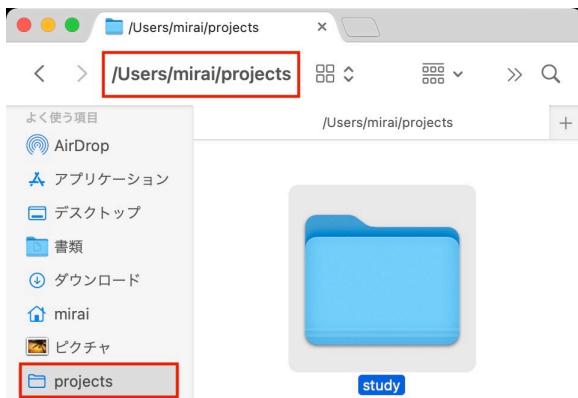
0. 左側に Atom を配置、右側に Firefox を配置します。
1. Atom で、プログラムをいろいろ書いていきます。
2. Firefox で、作ったプログラムの表示や動作を確認します。
3. 出来上がりが良ければ、プログラムを公開して完了です。
もう少し改良したければ、1. に戻ります。

*11 <https://www.mozilla.org/ja/firefox/new/>

♣ 手順のご案内

作業ディレクトリの作成方法や、Atom や Firefox の簡単な使い方のご紹介です。（Mac 利用者向けです。Windows の方は適宜読み替えてください。）
より、詳細な解説は [テキストエディタ Atom 入門*12](#) がお勧めです。

0. Users/利用者名ディレクトリの直下に、`projects` ディレクトリを作成します。`projects` ディレクトリには、ご自身で作成するいろいろなプロジェクト（案件）を格納すると良いと思います。良く使うことになるので、Finder のサイドバーに登録しておくと便利です。`projects` ディレクトリが作成できたら、その下に今回の学習で用いる `study` ディレクトリを作成します。



▲ 図 1.8: study ディレクトリの作成

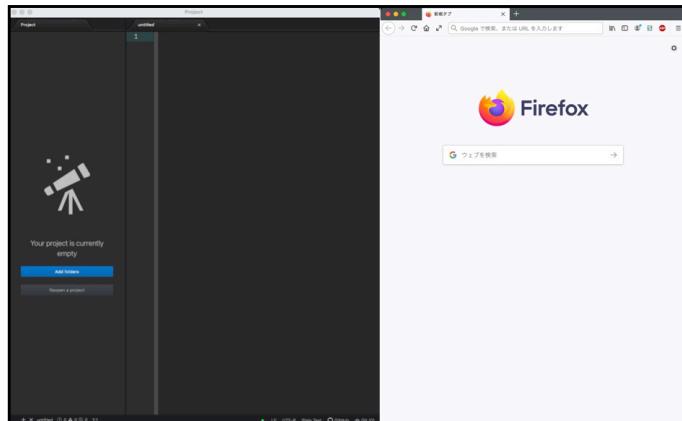
1. Atom のアイコンと Firefox のアイコンです。
良く使うのでドックに登録しておくと便利です。

*12 <https://books.oiax.jp/items/atom>



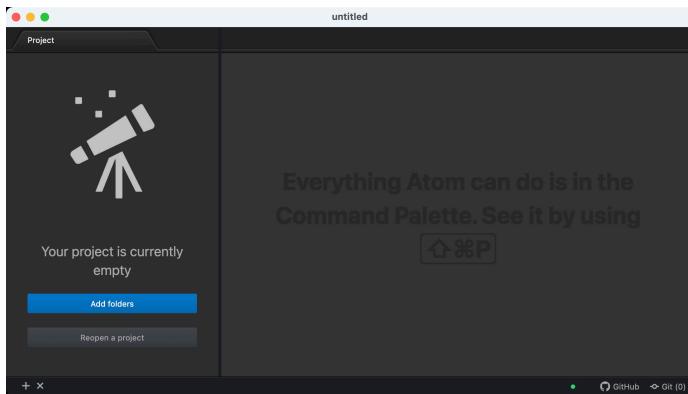
▲図 1.9: Atom(左)と Firefox(右)のアイコン

2. Atom と Firefox を左右に配置します。



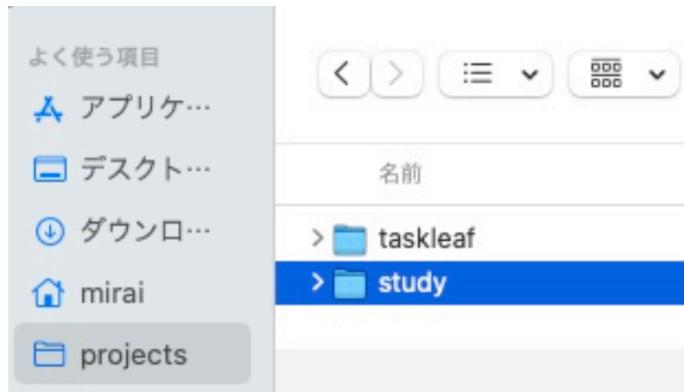
▲図 1.10: Atom(左)と Firefox(右)を配置

3. Atom には、「ツリービュー」に「プロジェクトフォルダ」を追加する機能があります。「プロジェクトフォルダ」を追加すると、Atom の画面左側に配置されている「ツリービュー」にそのプロジェクトフォルダ内のファイルやディレクトリが一覧表示されます。ファイルの追加や編集、削除を簡単に行うことができるようになるので、とても便利です。プロジェクトフォルダを追加するために、左側の青色の「Add folders」ボタンを押します。



▲図 1.11: 左側の青色の「Add folders」ボタンを押す

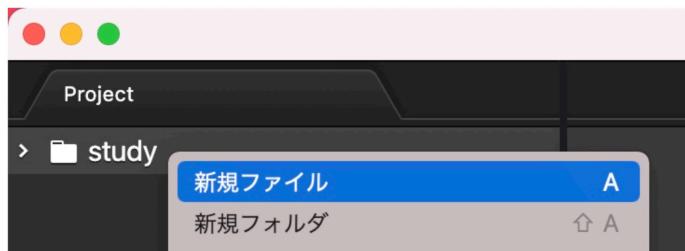
4. Finder が開くので、サイドバーから projects を選びます。projects ディレクトリ内には、taskleaf と study の二つのディレクトリがありますが、ここでは study ディレクトリを開きます。



▲図 1.12: projects 内の study を選ぶ

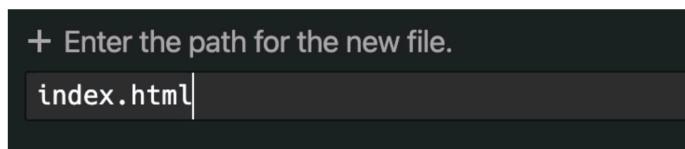
5. 左側の「ツリービュー」に study ディレクトリが表示されました。この study ディレクトリ内にいろいろなファイルを納めていき、最終的に一本の作品を創り上げていきます。今は何もないディレクトリな

ので、新しくファイルを作成しましょう。ツリービューの `study` ディレクトリを右クリックします。コンテンツメニューが表示されるので、「新規ファイル」をクリックします。



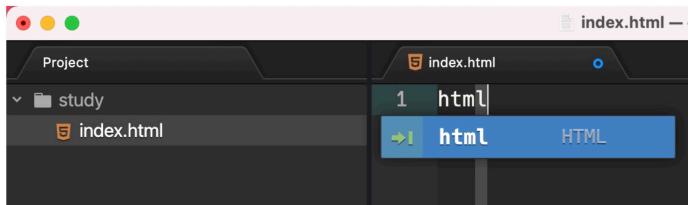
▲図 1.13: `study` を右クリックし 新規ファイルをクリックする

6. 「Enter the path for the new file.」(新しいファイル名を入力して下さい)と、表示されるので、`index.html` とファイル名を入力します。



▲図 1.14: 新しいファイル名として `index.html` と入力する

7. 左側の「ツリービュー」に、新規作成した `index.html` が、表示されました。早速これをクリックします。すると右側の編集領域(「ペイン」と呼ばれています)に、今作成した `index.html` が表示されます。当然、今作成したばかりなので中身は何もなく空っぽです。早速 HTML を書いていきましょう。Atomには、コードの入力を手助けしてくれる「入力支援機能」が備わっています。`html` と入力して、エンターキーを押します。



▲図1.15: 「html」と入力して「入力支援機能」を活用する

8. 次のように雛形が入力されます。

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    ...
  </body>
</html>
```

The image shows the completed HTML template from the previous screenshot. The entire document structure is now visible, including the head and body sections with their respective tags and content. The code is color-coded according to the Atom theme, with tags in pink and attributes in green.

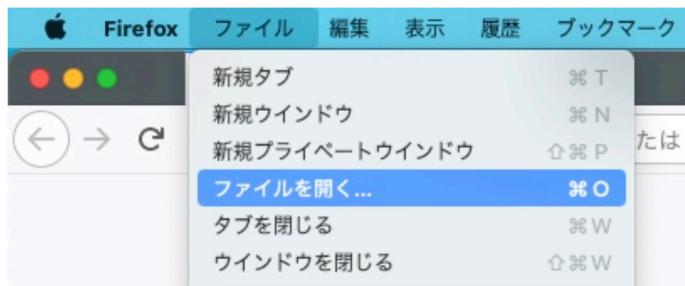
▲図1.16: 入力支援機能により入力された雛形

9. 「色」がついていることにも着目してください。「シンタックスハイライト」と呼ばれる、HTMLの文法に分かりやすいよう、着色する機能です。それでは今雛形で入力した結果を利用して、次のようにしましょう。`h1`とタイプしエンターキーを押すと、`<h1></h1>`と入力されます。Atomの入力支援機能を活用して編集しましょう。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML学習</title>
  </head>
  <body>
    <h1>初めてのHTML</h1>
    <p>
      HTMLを学習して、
      素敵なサイトを作れるよう
      成ります。
    </p>
  </body>
</html>
```

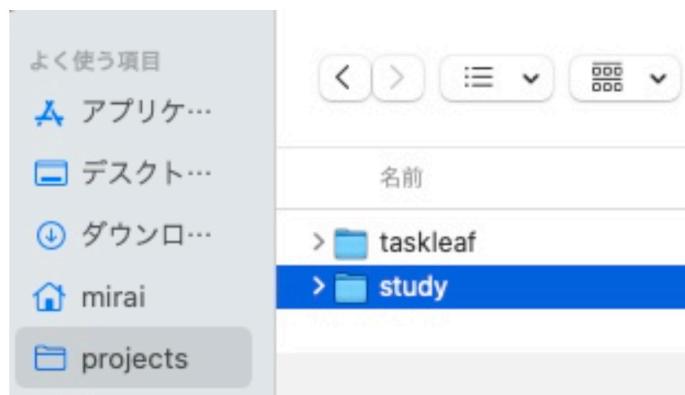
▲ 図 1.17: 適宜編集して完成した HTML

10. HTML を入力して、`index.html` ファイルが完成しました。Atom の「ファイル」メニューから「保存」をクリックして、保存しましょう (`Command + S`) を押しても保存することができます)。保存ができたら、どのように表示されるのか、Firefox を使って確認してみましょう。Firefox の「ファイル」メニューから、「ファイルを開く」をクリックします (`Command + O`) を押しても開くことができます)。



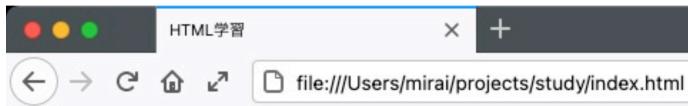
▲図1.18: ファイルメニューから「ファイルを開く」をクリック

11. Finder が開くので、サイドバーから projects を選びます。
projects ディレクトリ内には、taskleaf と study の二つのディレクトリがありますが、ここでは study ディレクトリを開きます。



▲図1.19: projects内の「study」を選ぶ

12. Atom で編集した index.html を Firefox で閲覧できます。

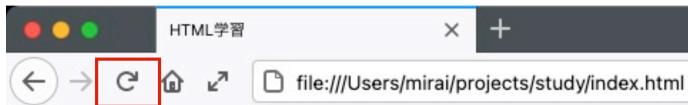


初めてのHTML

HTMLを学習して、素敵なサイトを作れるように成ります。

▲図 1.20: index.html を Firefox で閲覧する

13. 紹介に出来上がっていたら完成です。もしもう少し改善したい点があるなら、Atom に戻って、index.html を編集します。編集が終わったら、Command + S を押して保存しましょう。保存が終わったら、編集結果を Firefox で見てみましょう。「再読み込みボタン」を押すか、または Command + R を押して、編集結果を再確認することができます。



▲図 1.21: 再読み込みボタン