



PandAを便利にする Chrome拡張を作ってみよう！

2024/05/30

KMC 46th  Walnuts



つくるもの

「PandA Downloader」

- PandAの授業資料を
まとめてダウンロードできる
拡張機能です！





流れ

- 初期状態でまずインストールしてみる
- 簡単なバグを直す
- ダウンロード機能を追加する
- ローディングの表示を作る

Codeboxにログイン

- 今回もみなさんに開発してもらう環境を用意しています。
- 配布したURLにアクセスし、パスワードを入力してください。

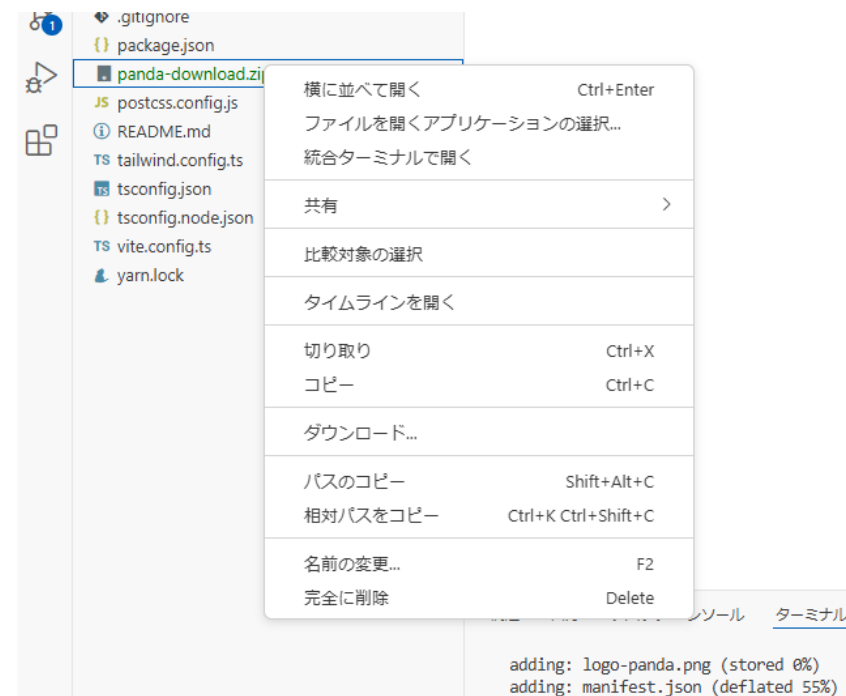
Welcome to code-server

Please log in below. Password was set from \$PASSWORD.

ビルド

- Chrome拡張をビルドしてみます

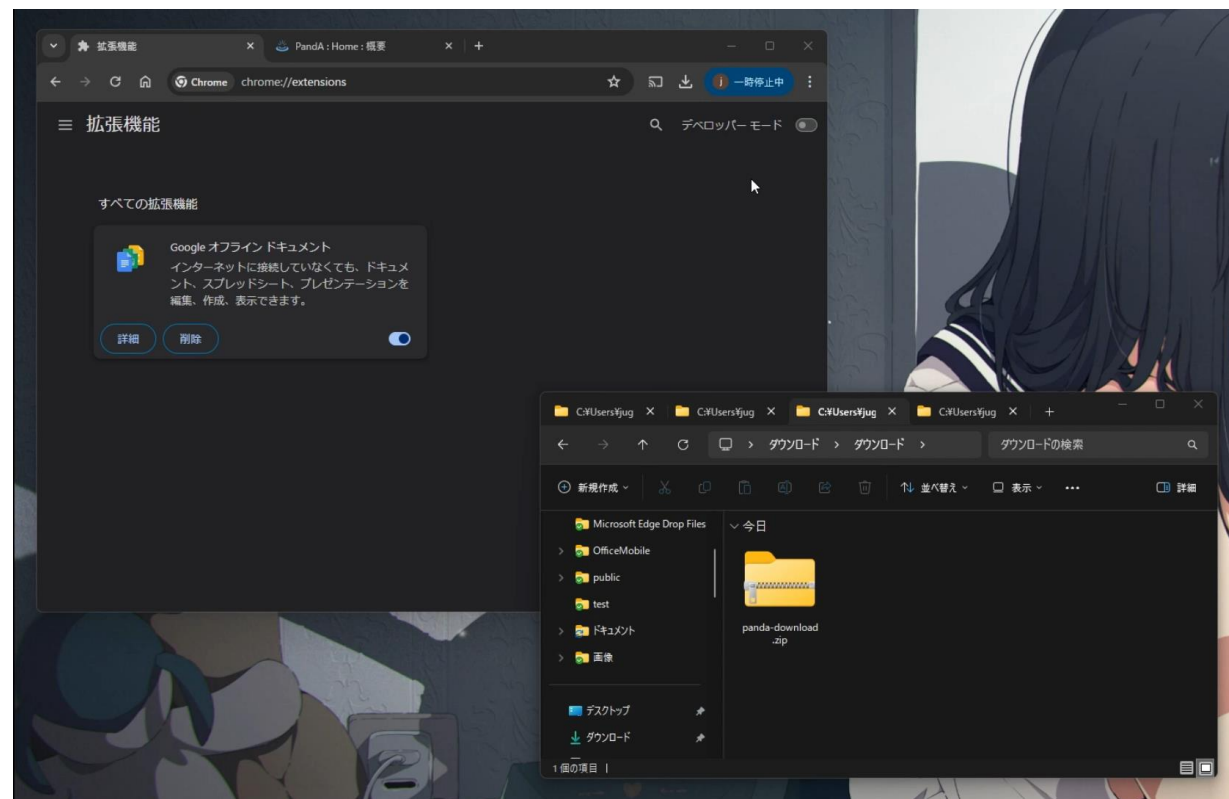
1. `ctrl + shift + @` or     でターミナルを開く
2. `yarn build` で拡張機能をビルドする
3. 出来上がった `panda-download.zip` を右クリックして、「ダウンロード」



インストール

1. Chromeで、chrome://extensions/ を開く
2. 右上の「デベロッパーモード」をONにする
3. さっきダウンロードしたZIPを、
ドラッグ&ドロップ
4. 完了 🎉

コードを書き換えるたびに毎回これをします！

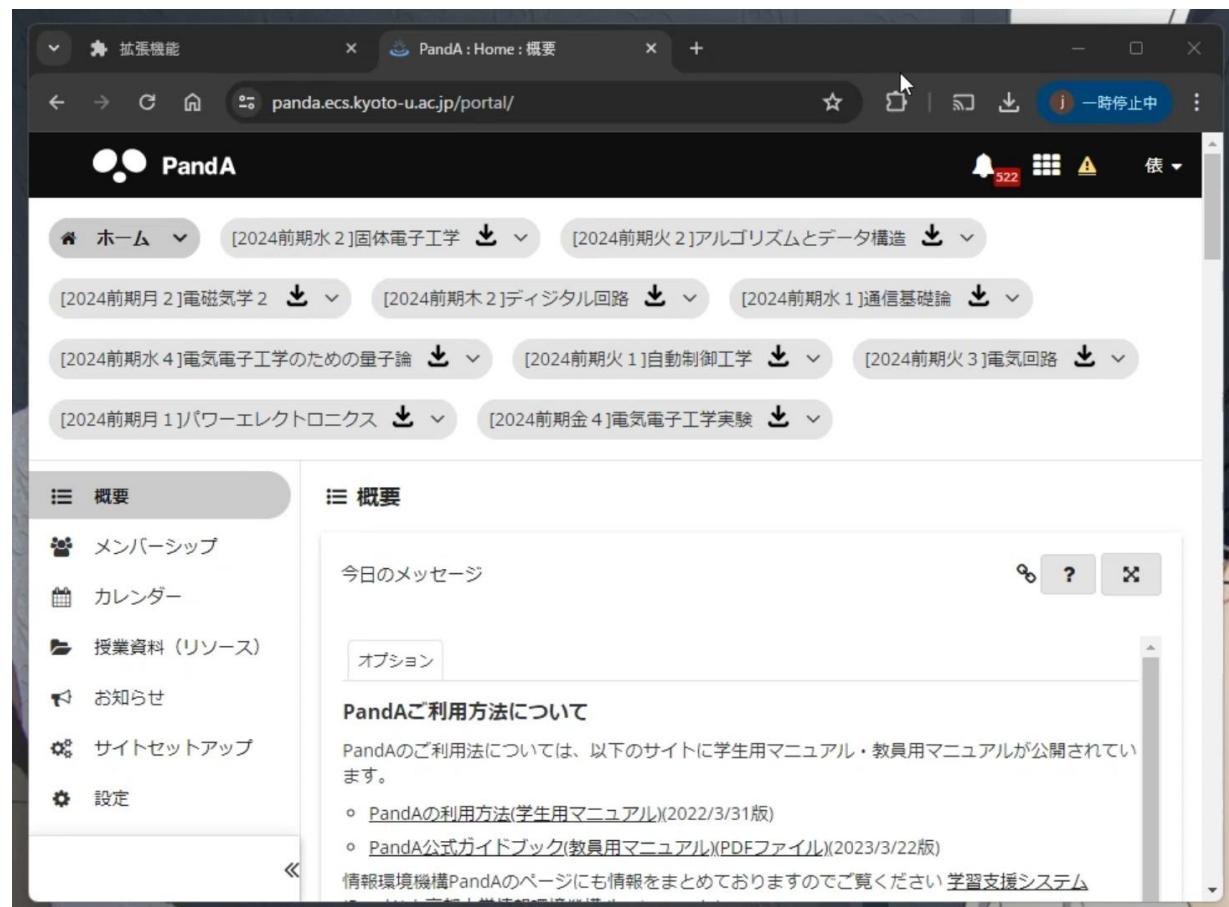


動作確認

✓ <https://panda.ecs.kyoto-u.ac.jp/portal/>にアクセスした時、

ダウンロードボタンが増えている

✓ 拡張機能のアイコンをクリックすると、
ポップアップが表示される





ステップ1: ポップアップを編集

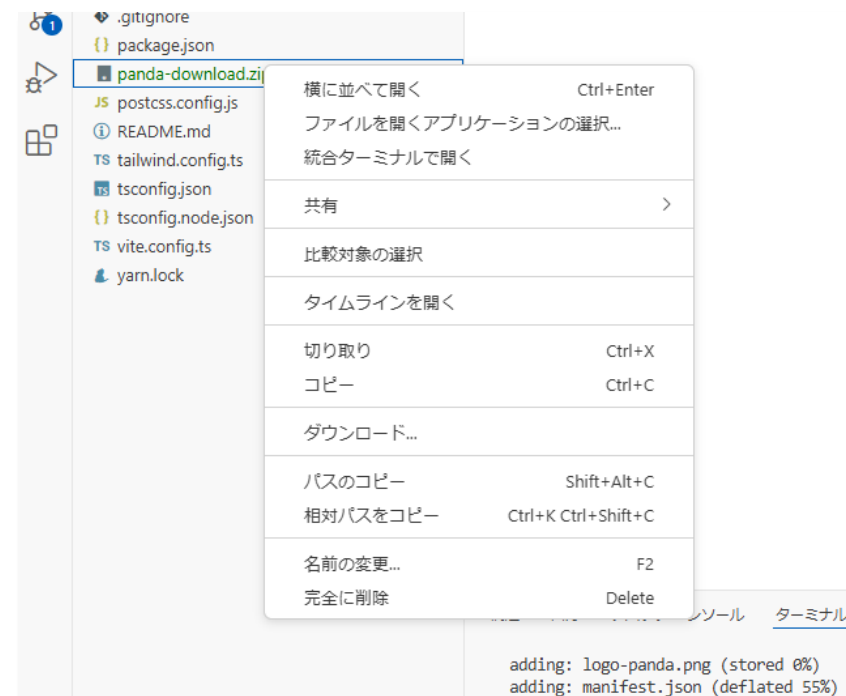
編集するファイル: `src/popup/App.tsx`

1. PandAのリンクをなおそう
2. -1されるバグを直そう

ビルド

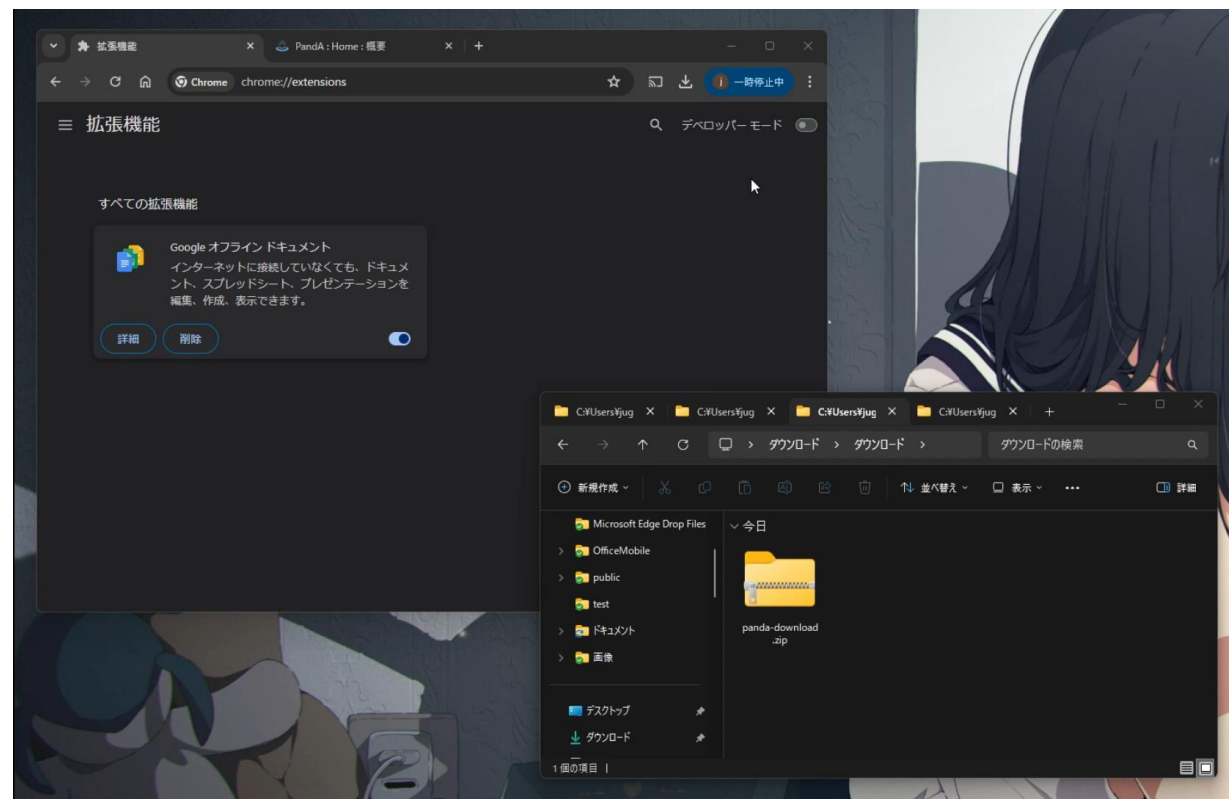
- Chrome拡張をビルドしてみます

1. `ctrl + shift + @` or     でターミナルを開く
2. `yarn build` で拡張機能をビルドする
3. 出来上がった `panda-download.zip` を右クリックして、「ダウンロード」



インストール

1. Chromeで、chrome://extensions/ を開く
2. 右上の「デベロッパーモード」をONにする
3. さっきダウンロードしたZIPを、ドラッグ&ドロップ
4. 完了 🎉



ステップ1: 解説

- App()関数のreturn部分で、ポップアップの画面を定義しています
 - index.htmlを見てみると、includeされている

```
function App() {  
  const [count, setCount] = useState(0);  
  return (  
    <div className=" flex flex-col gap-4">  
      ...  
    </div>  
  );  
}  
  
export default App;
```

ステップ1: 解説: リンクの編集

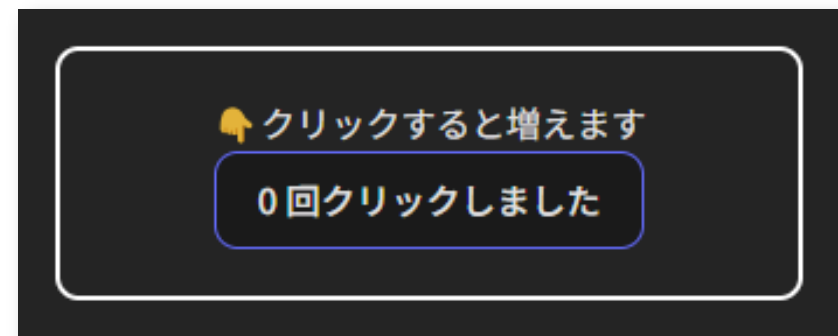
- `<p>`は段落を表します
- `<a>`はリンクを表します
 - `<a>`で囲まれた部分がリンクになる
 - hrefがリンク先のURL
 - href=「*hypertext reference*」

PandAはこちら

```
<p className="text-lg">
  PandAは
  <a href="https://www.ueno-panda.jp/" target="_blank" rel="noopener">
    こちら
  </a>
</p>
```

ステップ1: 解説: カウンター

- <button>はその名の通りボタンです。押されるたびにonClickの中身を実行します
- setCountの部分は、(count)で、現在の値を受け取って、=> の後で新しい値を定義します
 - たとえばsetCount((count)=>1)なら何回押しても1



```
const [count, setCount] = useState(0);  
...  
<button type="button" onClick={() => setCount((count) => count +  
1)}>  
  {count} 回クリックしました  
</button>
```

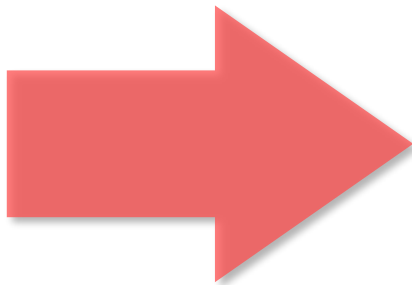
ステップ1: 解説:useState

- しかし、なぜ`count=count+1`ではダメなのでしょう？

JS

```
let a = "変数の中身";  
return (<p>  
  {a}  
</p>);
```

Render



DOM (HTML)

```
<html lang="ja">  
  <body>  
    <p>  
      変数の中身  
    </p>  
  </body>  
</html>
```

ステップ1: 解説:useState

- しかし、なぜ`count=count+1`ではダメなのでしょう？

JS

```
let a = "変数の中身";  
return (<p>  
  {a}  
</p>);  
~~~~~  
a = "書き換えた";
```

.....

DOM (HTML)

```
<html lang="ja">  
  <body>  
    <p>  
      変数の中身  
    </p>  
  </body>  
</html>
```

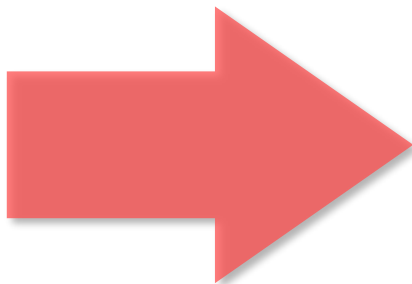
ステップ1: 解説:useState

- useStateを使うと...

JS

```
const [a, setA] =  
  useState("変数の中身");  
  
return (<p>  
  {a}  
</p>);
```

Render



DOM (HTML)

```
<html lang="ja">  
  <body>  
    <p>  
      変数の中身  
    </p>  
  </body>  
</html>
```

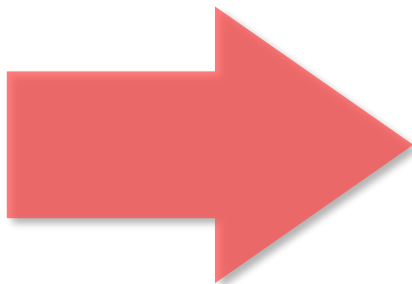

ステップ1: 解説:useState

- useStateを使うと...

JS

```
const [a, setA] =  
  useState("変数の中身");  
  
return (<p>  
  {a}  
</p>);  
~~~~~  
setA("書き換えた")
```

再Render



DOM (HTML)

```
<html lang="ja">  
  <body>  
    <p>  
      書き換えた  
    </p>  
  </body>  
</html>
```

Good!

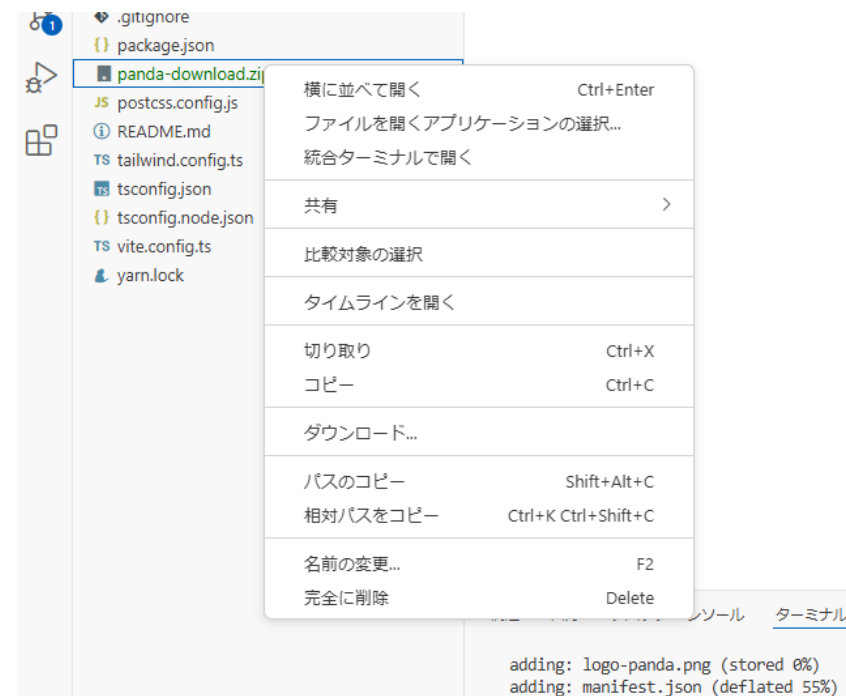


👉 setAの中で、書き換わったことを検知し、再レンダリングをトリガーしている

ビルド

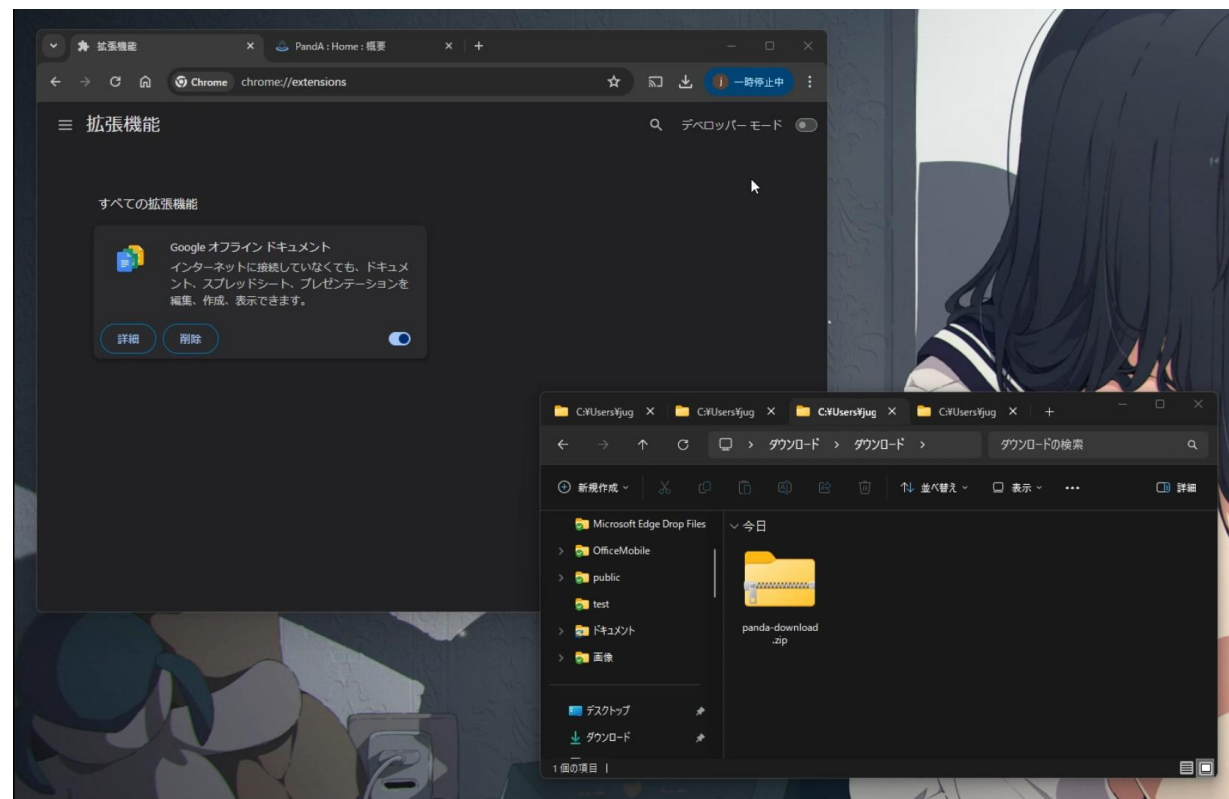
- Chrome拡張をビルドしてみます

1. `ctrl + shift + @` or     でターミナルを開く
2. `yarn build` で拡張機能をビルドする
3. 出来上がった `panda-download.zip` を右クリックして、「ダウンロード」



インストール

1. Chromeで、chrome://extensions/ を開く
2. 右上の「デベロッパーモード」をONにする
3. さっきダウンロードしたZIPを、ドラッグ&ドロップ
4. 完了 🎉



ステップ2: Downloadを呼び出す

編集するファイル: `src/content/Download.tsx`

1. DownloadボタンのonClickの中で、download関数を呼び出そう
 - download関数は、siteIDとsiteNameを受け取って、そのサイトの授業資料をダウンロードする関数です
 - 呼び出すときは、`await download(siteID, siteName);`
2. download関数を呼びだしている行の前後に、「ダウンロード中かどうか」を表す、isDownloadingを操作する処理を書こう
 - さっき説明した、setIsDownloading関数を使います
3. isDownloadingだったら、<LoadingIcon/>、そうじゃなかったら<DownloadIcon/>を表示するようにしよう



ステップ2: 解説

- onClickの中身

```
// ✅ setIsDownloading関数を使って、ダウンロード中かどうかを管理しよう！
setIsDownloading(true);

// ✅ download 関数に siteID と siteName を渡して、ダウンロードできるようにしよう！
await download(siteID, siteName);

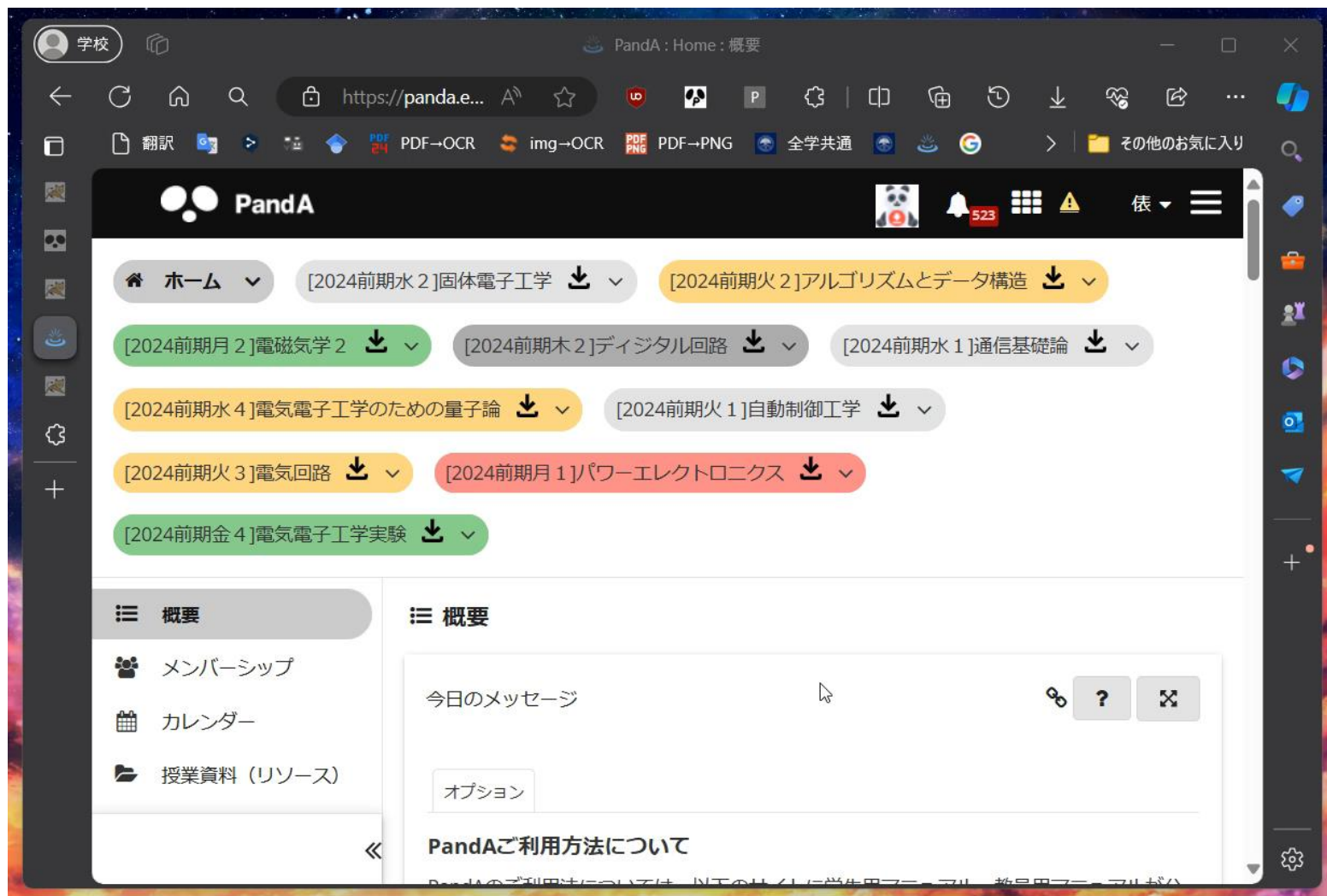
setIsDownloading(false);
}}
```

ステップ2: 解説

- ボタン要素の中身

```
{ isDownloading ? (  
  // ✅ <LoadingIcon /> を使ってダウンロード中なことを示そう！  
  <LoadingIcon />  
) : (  
  <DownloadIcon />  
)}
```

ステップ2: 解説



ステップ3: アニメーションを付けよう

編集するファイル: `src/icons/loading.css`

1. `<LoadingIcon />`に回転アニメーションを付けてみましょう
2. 時間があれば、オリジナルのアニメーションもつけてみましょう
 - `src/icons/download.css` をいじってもOK

GENERATORS

CSS

ANIMATION

Keyframe Animation

BACKGROUND

Background Color

Background Gradient

Background Image

BOX

Border

Border Image

Border Radius

Box Resize

Box Shadow

Opacity

Outline

Overflow

COLOR

Text Color

FILTER

Blur

Brightness

Contrast

Drop Shadow

Grayscale

Keyframe Animation CSS Generator



OPTIONS

PREVIEW

Attention

Background

Basic

👉 Basicを使う

Animation type

Rotate Center

👉 アニメーションの種類

Name

myAnim

Duration

2

👉 何秒かけてアニメーションするか

Timing function

ease

Delay

0

seconds

Iteration count

何回実行するか（空欄で無限）

Direction

normal

Fill mode

forwards

null

CODE

Copy this to the element you want to animate.

`animation: myAnim 2s ease 0s infinite normal forwards;`

Copy

Copy this after the above selector.

`@keyframes myAnim {
 0% {
 transform: rotate(0);
 }

 100% {
 transform: rotate(360deg);
 }
}`

Copy

ステップ3: アニメーションを付けよう

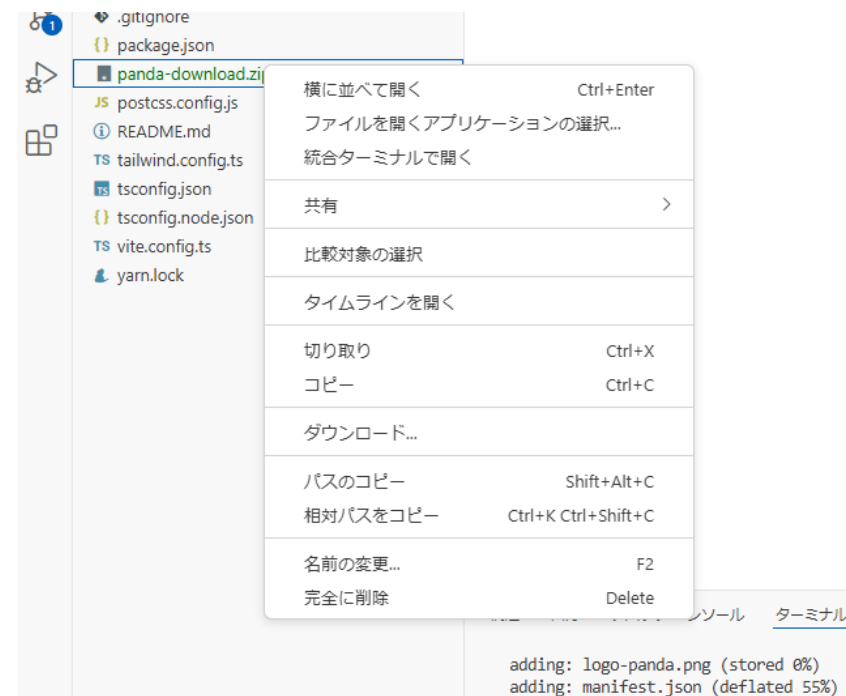
- 右側の、CODEをコピーして `src/icons/loading.css` に貼り付ける

```
.download-icon {  
  animation: myAnim 2s ease 0s infinite normal forwards;  
}  
  
@keyframes myAnim {  
  0% {  
    transform: rotate(0);  
  }  
  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

ビルド

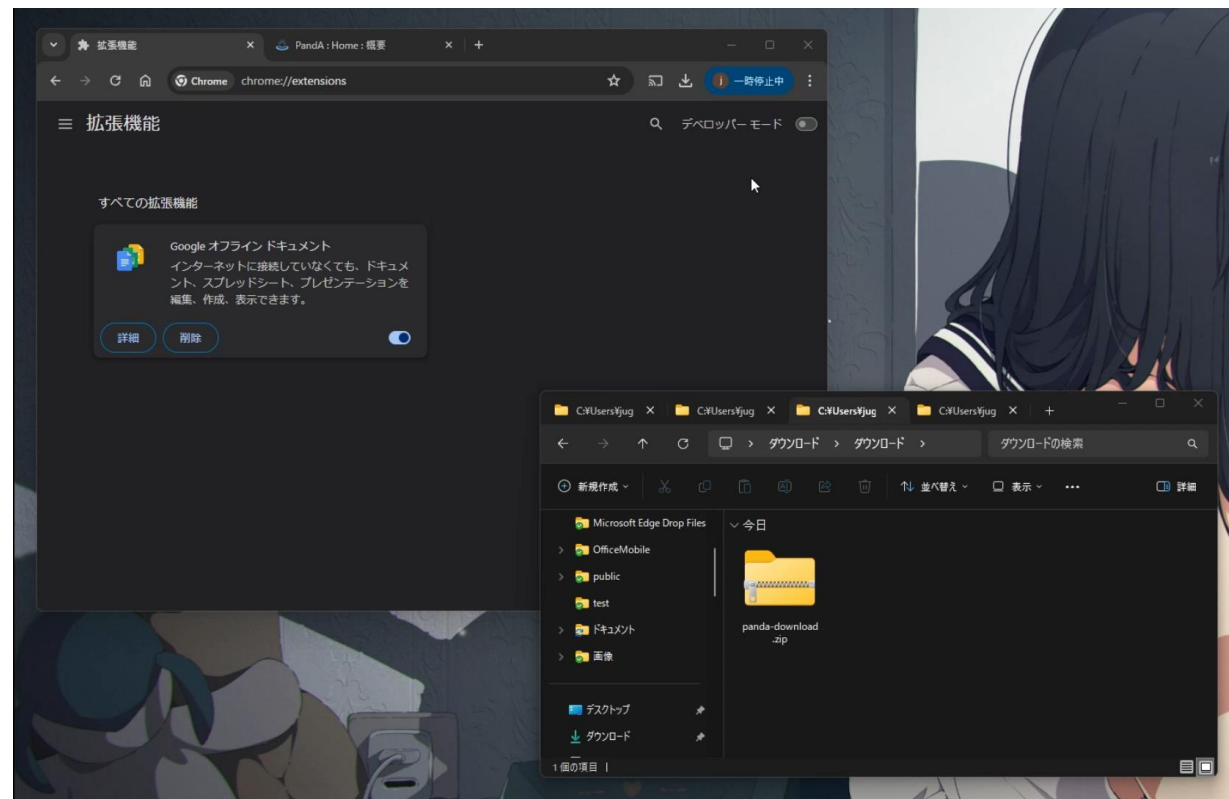
- Chrome拡張をビルドしてみます

1. `ctrl + shift + @` or     でターミナルを開く
2. `yarn build` で拡張機能をビルドする
3. 出来上がった `panda-download.zip` を右クリックして、「ダウンロード」



インストール

1. Chromeで、chrome://extensions/ を開く
2. 右上の「デベロッパーモード」をONにする
3. さっきダウンロードしたZIPを、ドラッグ&ドロップ
4. 完了 🎉



完成 🎉

- これからも使えるので是非活用&魔改造してみてください！
- ReactはChrome拡張を作る以外にもたくさん使えます！
 - （そもそもChrome拡張を作るためのものではない）
 - <https://walnuts.dev>
 - <https://oekaki.walnuts.dev/public>
 - とかはReactです

完成 🎉



完成 🎉

Walnuts.dev



Walnuts

X @walnuts1018

🐙 walnuts1018

📷 @walnuts_1018

✉️ r.juglans.1018@gmail.com

🏢 46th Chairman /
47th Representative

🏠 Kyoto Univ. / B3

❤️ Kubernetes / Proxmox VE
/ Golang / Next.js

📖 😊

🔧 walnuts1018/infra

Kubernetesのインフラ構成を管理するリポジトリです。

🔧 Walnuts.devを支える技術 (物理環境/ネットワーク編)

🔧 walnuts1018/k8s-badge

README用に、k8sクラスタのstatus badgeを生成するActionです。

🔧 walnuts1018/OpenChokin

【開発中】貯金を公開するサービスです。 / 鴨川ハッカソンで優秀賞をいただきました。

🔧 お絵描き伝言ゲーム

NF2023で開催したお絵描き伝言ゲームの結果発表ページです。

🔧 HedgeDocをKubernetesに構築する

🔧 walnuts1018/waketime-to-slack

Waketimeから今書いている言語をリアルタイムで取得し、Slackのステータスに反映します。

🔧 walnuts | しずかなインターネット

🔧 ESP32と赤外線LEDを用いてエアコンを遠隔操作する

🔧 walnuts1018/machine-status-api

ラズパイと自作回路を組み合わせ、遠隔でサーバーの電源を操作するAPIです。

🔧 DIGI-CON HACKATHON 2023

DIGI-CON HACKATHON 2023でジャンプ+賞をいただきました。

