테일윈드 다크모드 적용 학습

테일윈드 다크모드

다크모드

- 밝은 색의 텍스트, 아이콘, GUI요소들을 어두운 배경 위에 사용하는 모드
- 일부 사용자들은 다크모드에 더 시각적 매력을 느끼기도 하고 전력소비 감소도 경험
- 어두운 환경에서 밝은 화면은 시각적 피로를 주기 마련인데 이런 모드를 통해 사용자에게 편리함을 제공하기 위한 모드이다.

테일윈드의 다크모드

CSS 미디어 기능

- dark 미디어를 이용하여 dark모드에 적용할 CSS를 작성할 수 있다.

```
<div class="bg-white dark:bg-slate-800 rounded-lg px-6 py-8 r
  <div>
    <span class="inline-flex items-center justify-center p-2
      <svg class="h-6 w-6 text-white" xmlns="http://www.w3.or
    </span>
  </div>
  <h3 class="text-slate-900 dark:text-white mt-5 text-base fo
```

```
  <p class="text-slate-500 dark:text-slate-400 mt-2 text-sm">
    The Zero Gravity Pen can be used to write in any orientat
  </p>
</div>
```

수동으로 다크모드 전환하기

- Tailwind CSS는 기본적으로 CSS의 **prefers-color-scheme**을 사용
  - 단,다크모드의 경우 이를 지원해주기 위해 dark 클래스를 사용
  - dark 클래스는 다크모드와 라이트 모드를 수동으로 설정할 수 있게 해줌
- tailwind.config.js에서 darkMode : 'class'추가
  - class에 dark를 add,remove 하는 것을 통해 다크모드 컨트롤이 가능
  - 다크 모드를 추가하고 나면 hover나 반응형 레이아웃을 사용하는 것처럼 다크모드 스타일은 기존 스타일에 `dark:`을 사용하여 스타일을 추가 가능

```
// tailwind.config.js
module.exports = {
  darkMode: 'class',
}
```

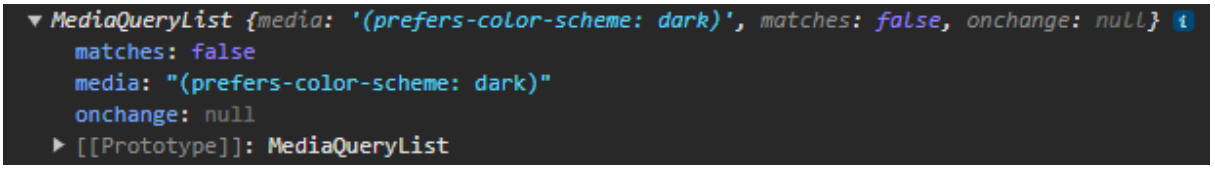
다크모드 적용해보기

- 다크모드 & 라이트 모드
  - html class에 dark를 add, remove를 하여 컨트롤할 수 있게 해줘야.
  - 이러한 작업을 위해 보통 localStorage를 많이 이용

```
if( localStorage.theme === 'dark'
   || (!('theme' in localStorage)
   && window.matchMedia('(prefers-color-scheme: dark)').match
  document.documentElement.classList.add('dark');
} else {
```

```
  document.documentElement.classList.remove('dark');
}
```

- window.matchMedia(('prefers-color-scheme: dark'))
  - 사용자의 기본 시스템 설정에서 라이트/다크모드 설정을 확인할 수 있는 web API
  - matchMedia().matches는 boolean으로 받을 수 있어 토글의 조건으로 사용



- redux-toolkit을 이용한 적용 예시 (전역 상태관리)

```
import { createSlice } from '@reduxjs/toolkit';

const initialState = {
  isDark: false,
}

const themeSlice = createSlice({
  name: theme,
  initialState,
  reducers: {
    lightMode: (state) => {
      state.isDark = false;
      document.documentElement.classList.remove('dark');
    },
    darkMode: (state) => {
      state.isDark = true;
      document.documentElement.classList.add('dark');
    }
  }
})

export const { lightMode, darkMode } = themeSlice.actions;
export default themeSlice.reducer;
```

첫 접속시 테마 적용하기 (커스텀 훅 이용방법)

```
import { useEffect } from 'react';
import { useDispatch } from 'react-redux';

import { lightMode, darkMode } from 'store/modules/theme';

// 커스텀 훅
export const useThemeEffect = () => {
  const dispatch = useDispatch();

  const save = ( value ) => {
    localStorage.setItem( 'theme', value );
  }

  useEffect( () => {
    const theme = localStorage.getItem( 'theme' );

    if ( !theme || theme === 'light') {
      dispatch( lightMode() );
    } else {
      dispatch( darkMode() );
    }

    save( theme || 'light' );

    const mediaCheck = window.matchMedia( '(prefers-color-sch

    if ( theme !== 'dark' && mediaCheck ) {
      dispatch( darkMode() );
      save( 'dark' );
    } else if (theme === 'dark' && !mediaCheck) {
      dispatch( lightMode() );
      save( 'light' );
    }
```

```
  }, [] );
}
```

토글 버튼 (라이트모드, 다크모드 전환)

```
const DarkToggle = () => {
  const dispatch = useDispatch();
  const { isDark } = useSelector(state => state.theme);

  const save = (value) => {
    localStorage.setItem('theme', value);
  }

  const toggleHandler = () => {
    if( isDark ){
      dispatch( lightMode());
      save('light');
    } else {
      dispatch( darkMode());
      save('dark');
    }
  }
  return (
    <aside role='button' onClick={toggleHandler}>
      <div className={`${isDark} && styles.isDark`}>
      </div>
    </aside>
  )
}
```

다크모드 팔레트

- 라이트모드/다크모드의 팔레트를 정해야.
- 팔레트는 Tailwind가 구성해 놓은 팔레트를 그대로 써도 되지만 light, dark를 나누어 확장 할수도 있다.

- 색상을 선택할 때 구글 Material Design에서는 글자와 배경 간에 적어도 15.8:1 비율의 대비를 적용하도록 권장.
    - 대비율은 Color Contrast Tools 에서 확인 가능

```
//tailwind.config.js
module.exports = {
  theme: {
    extend: {
      colors: {
        primary: {
          DEFAULT: '#CCE4F0',
          1: '#99C9E2',
          2: '#66ADD3',
          3: '#3392C5',
          4: '#0077B6',
        },
        light: {
          text: {
            DEFAULT: '#212529',
            1: '#CED4DA',
            2: '#868E96',
            3: '#495057',
          },
        },
        dark: {
          text: {
            DEFAULT: '#ECECEC',
            1: '#D9D9D9',
            2: '#ACACAC',
            3: '#595959',
          },
        },
      }
    }
  }
}
```