

개발환경 구축

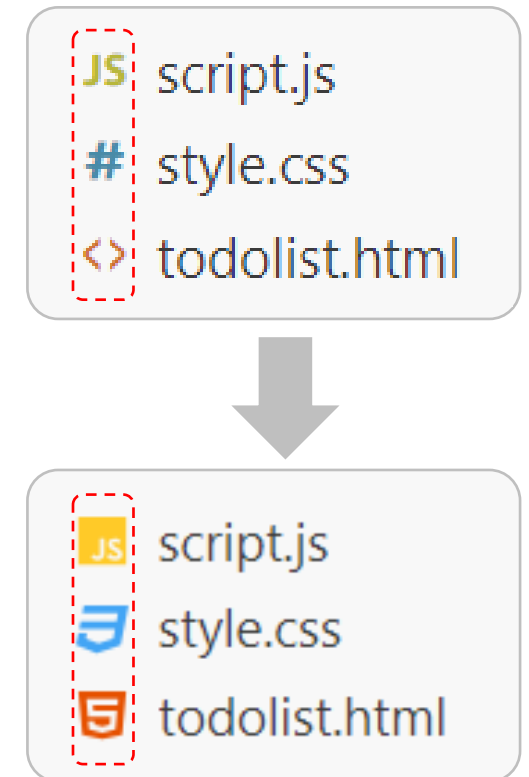
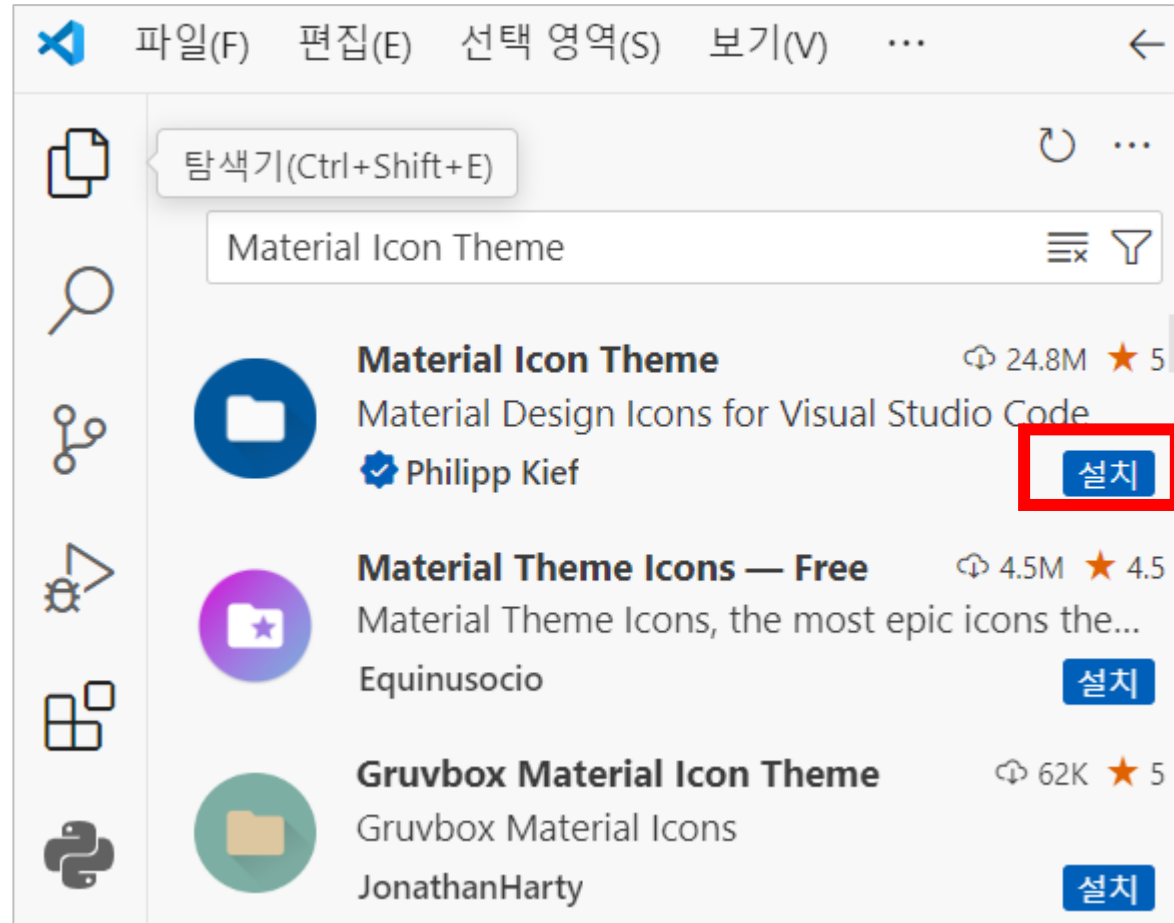
- ☑ VisualStudioCode 설치
- ☑ 확장 프로그램 설치
- ☑ 작업영역 만들기
- ☑ 웹문서 작성 및 테스트

개발환경



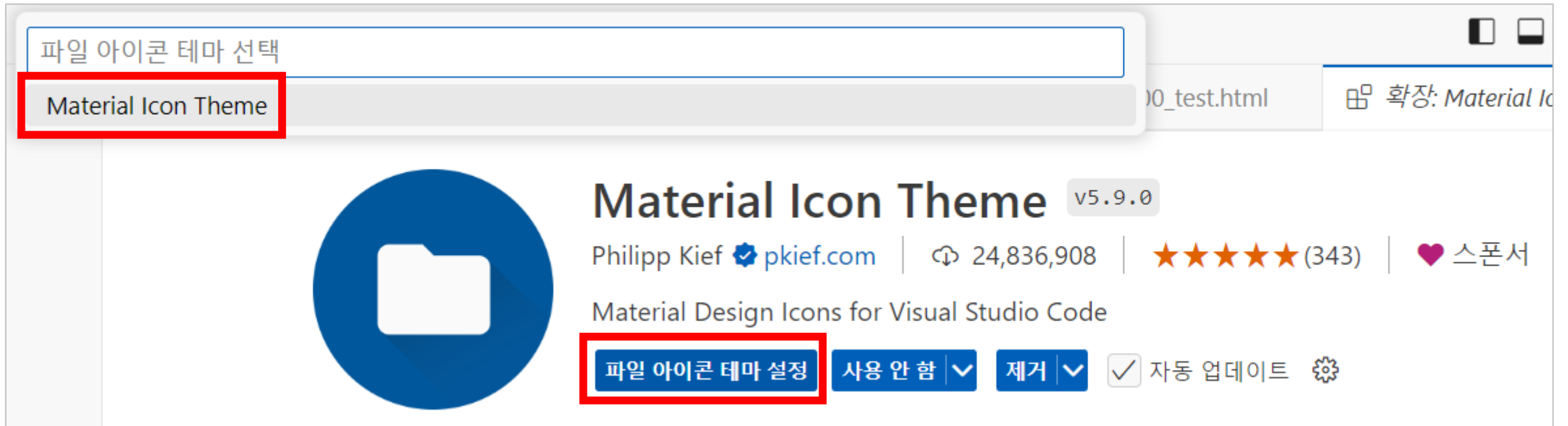
확장프로그램 설치

- **Material Icon Theme** : 파일아이콘을 직관적으로 표시



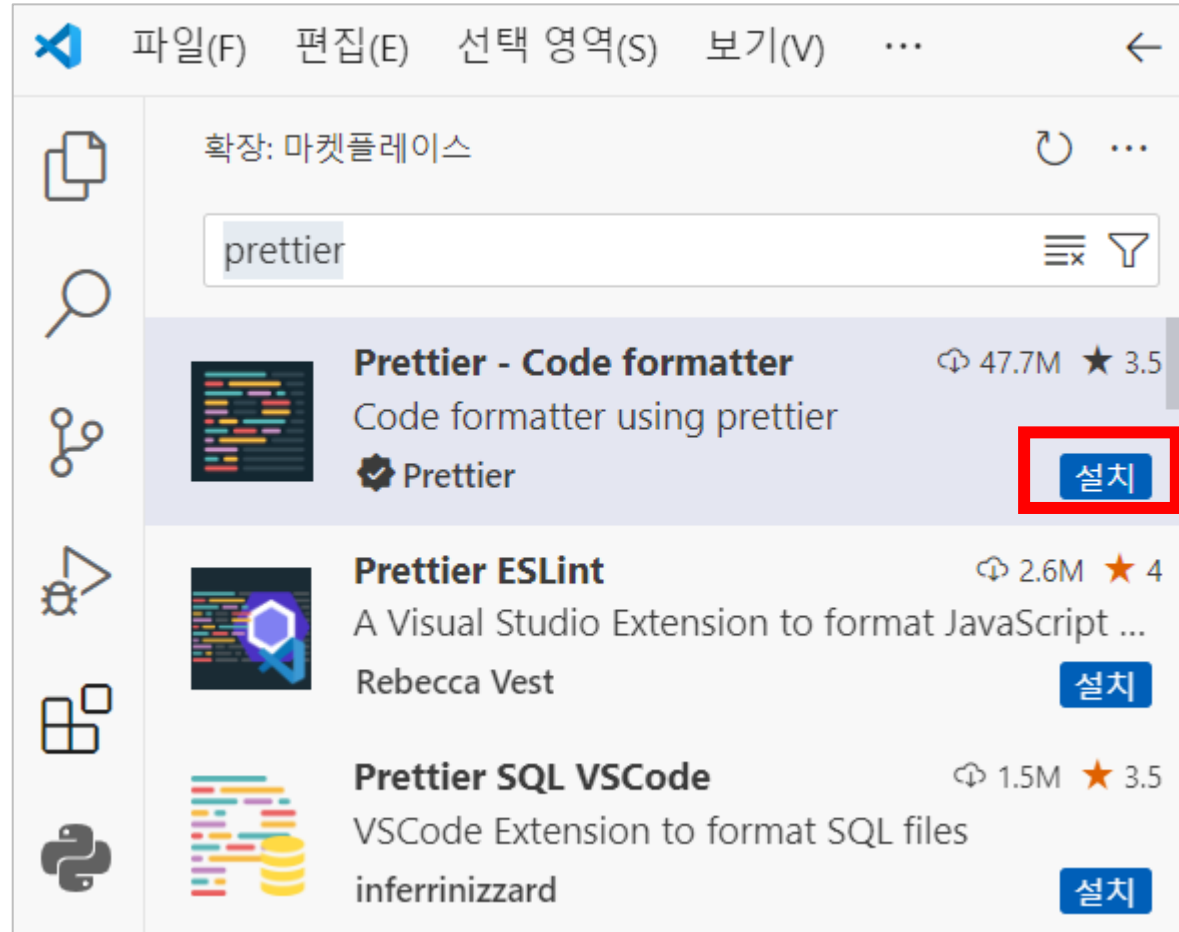
확장프로그램 설치

- **Material Icon Theme** : 파일아이콘을 직관적으로 표시



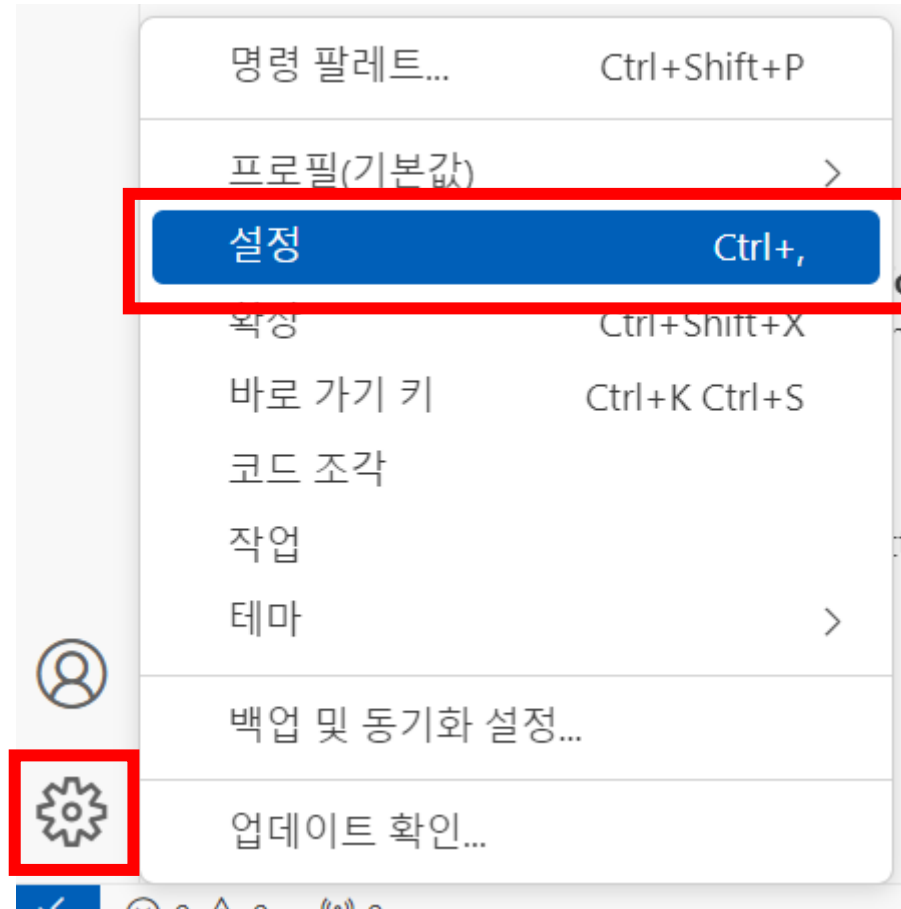
확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅



확장프로그램 설치

- Prettier - Code formatter : 코드를 일관된 스타일로 자동 포매팅



확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포매팅

formatter

3개 설정 찾기

사용자 작업 영역 폴더 ▼

백업 및 동기화 설정

텍스트 편집기 (1)

✓ 기능 (1)

Notebook (1)

✓ 확장 (1)

Prettier (1)

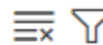
Editor: Default Formatter
다른 모든 포맷터 설정보다 우선하는 기본 포맷터를 정의합니다. 포맷터를 제공하는 확장 프로그램의 식별자 여야합니다.
Prettier - Code formatter ▼

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅

format on save

5개 설정 찾음



사용자 작업 영역 폴더 ▼

백업 및 동기화 설정

✓ 텍스트 편집기 (4)

서식 (2)

파일 (2)

✓ 기능 (1)

Notebook (1)

Editor: Format On Save Mode

저장 시 포맷이 전체 파일을 포맷하는지 또는 수정 내용만 포맷하는지를 제어합니다. [Editor: Format On Save](#)를 사용하도록 설정한 경우에만 적용됩니다.

file



file 저장 시 자동 포메팅

Editor: Format On Save

- ☒ 파일 저장 시 서식을 지정합니다. 포맷터를 사용할 수 있어야 하며, 파일이 지연 후에 자동으로 저장되지 않아야 하고, 편집기가 종료되지 않아야 합니다.

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅

tab width

6개 설정 찾기



사용자 작업 영역 폴더 ▼

백업 및 동기화 설정

- ▼ 워크벤치 (2)
 - 편집기 관리 (2)
- ▼ 기능 (2)
 - 터미널 (2)
- ▼ 확장 (2)
 - Emmet (1)
 - Prettier (1)

자식 요소가 많을 때 코드의 가독성 고려하여 tab width는 2 추천

Prettier: Tab Width

Number of spaces it should use per tab.

2

Terminal > Integrated: Commands To Skip Shell

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅

html

45개 설정 찾음

사용자 작업 영역 폴더 ▼

백업 및 동기화 설정

일반적으로 사용되... (1)

✓ 텍스트 편집기 (4)

파일 (1)

✓ 워크벤치 (1)

HTML > Format: Enable

☒ 기본 HTML 포맷터를 사용하거나 사용하지 않도록 설정합니다.

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포매팅

CSS

82개 설정 찾음

사용자 작업 영역 폴더 ▼

백업 및 동기화 설정

✓ 텍스트 편집기 (4)

- 글꼴 (2)

✓ 기능 (1)

- Notebook (1)

CSS > Format: Enable

☒ 기본 CSS 포맷터를 활성화/비활성화합니다.

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅

javascript 77개 설정 찾음

사용자 작업 영역 폴더 ▼ 백업 및 동기화 설정

✓ 텍스트 편집기 (1)
 제안 사항 (1)
✓ 기능 (1)

JavaScript > Format: Enable

☒ 기본 JavaScript 포맷터를 사용하거나 사용하지 않습니다.

확장프로그램 설치

- **Prettier - Code formatter** : 코드를 일관된 스타일로 자동 포맷팅

editor.linkedEditing

1개 설정 찾을 수 없음

사용자 작업 영역 frontend 폴더

백업 및 동기화 설정

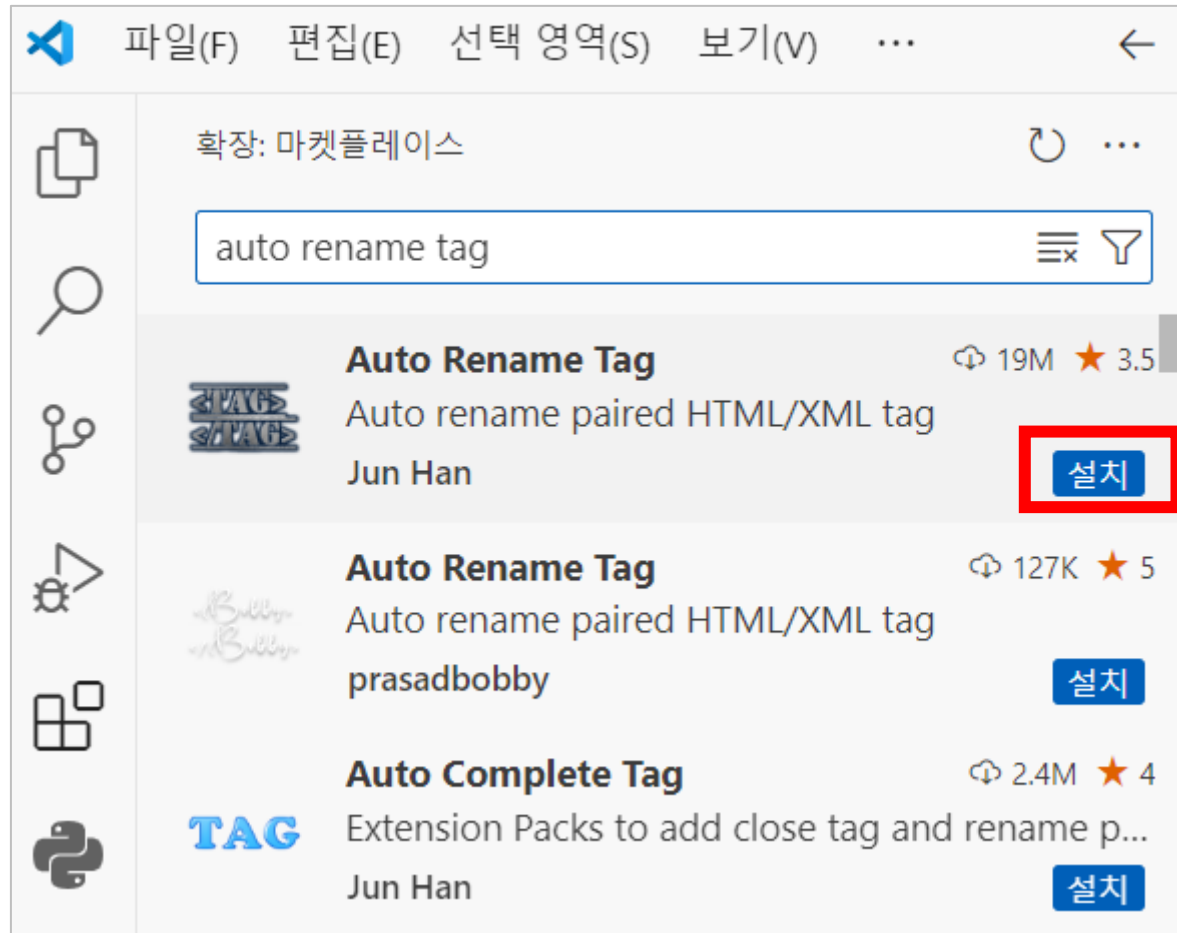
텍스트 편집기 (1)

⚙️

Editor: Linked Editing
☒ 편집기에서 연결된 편집이 사용하도록 설정되었는지를 제어합니다. 언어에 따라 관련 기호(예: HTML 태그)가 편집 중에 업데이트됩니다.

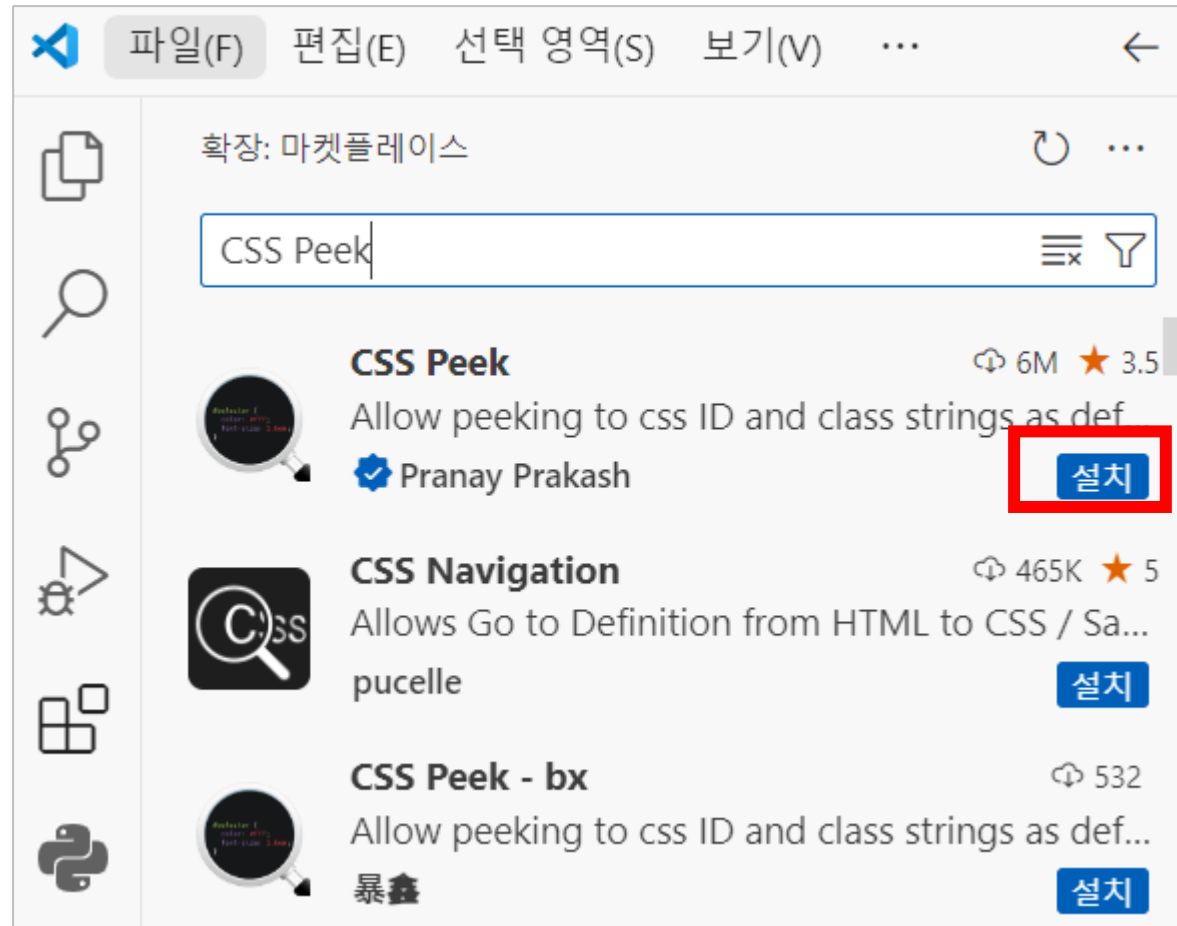
확장프로그램 설치

- **Auto Rename Tab** : HTML 태그의 이름을 변경할 때 여는 태그와 닫는 태그를 자동으로 함께 변경



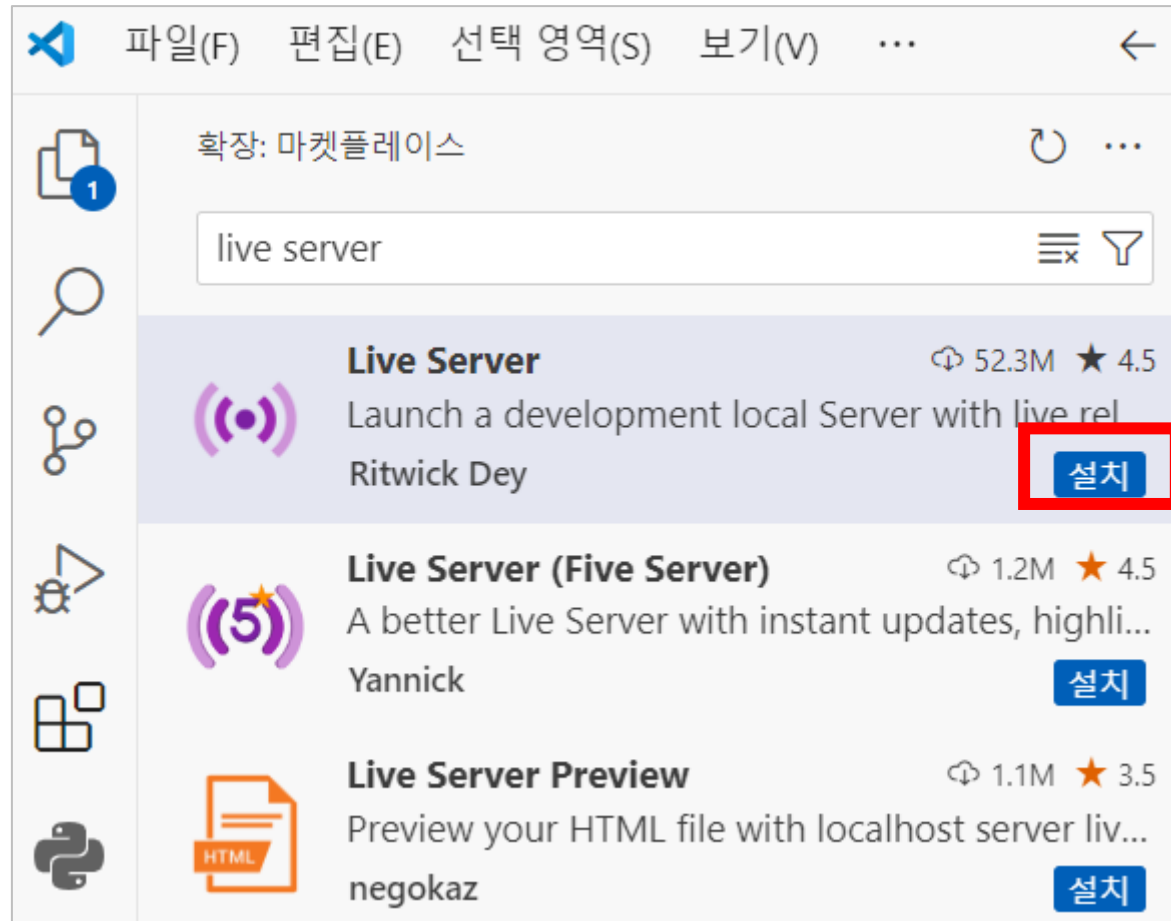
확장프로그램 설치

- **CSS peek** : HTML 파일에서 CSS 정의로 바로 이동



확장프로그램 설치

- **live server** : 로컬 서버를 바로 실행하고, 파일 변경 시 자동 새로고침



기본 브라우저 설정

- Chrome을 기본 브라우저로 설정한다.

window

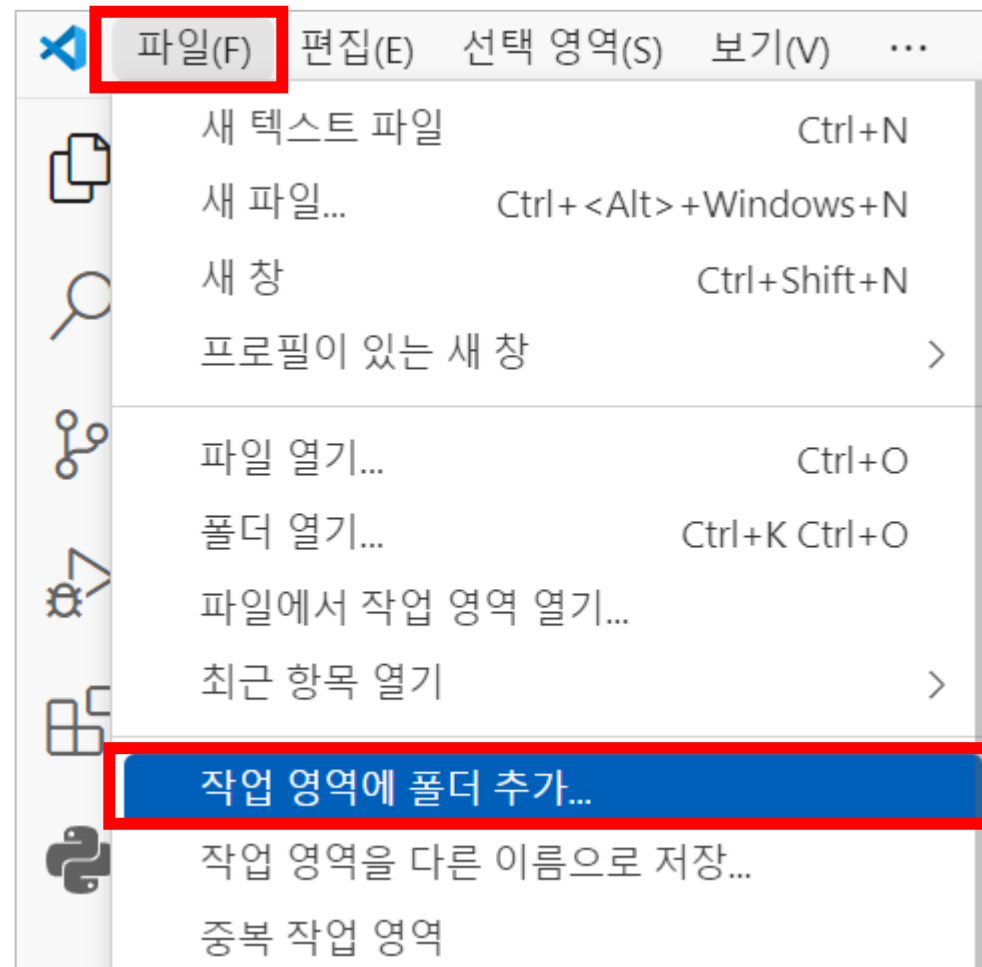
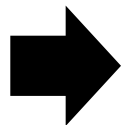
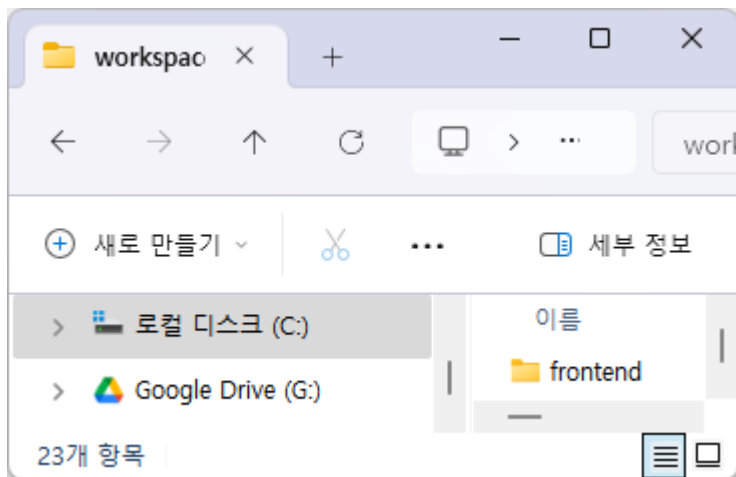
<https://support.google.com/chrome/answer/95417?hl=ko&co=GENIE.Platform%3DDesktop#zipppy=%2Cwindows>

mac

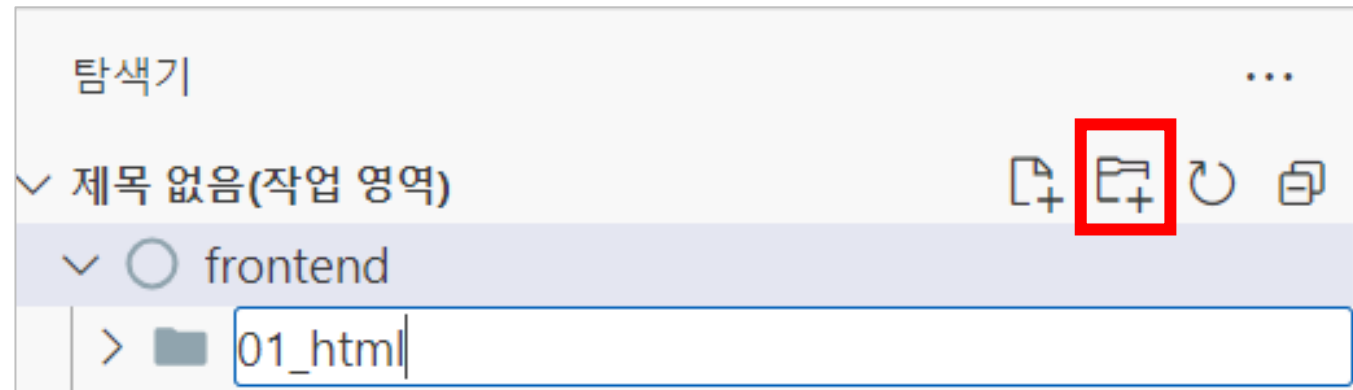
<https://support.apple.com/ko-kr/102362>

작업영역에 폴더 추가

윈도우 탐색기에서
작업폴더를 먼저 생성합니다.

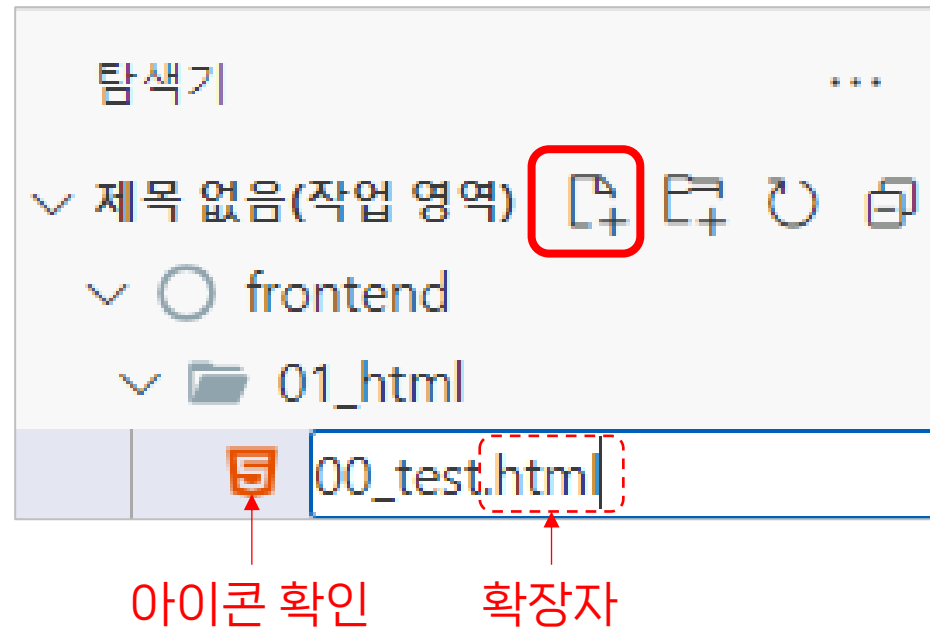


새 폴더 만들기



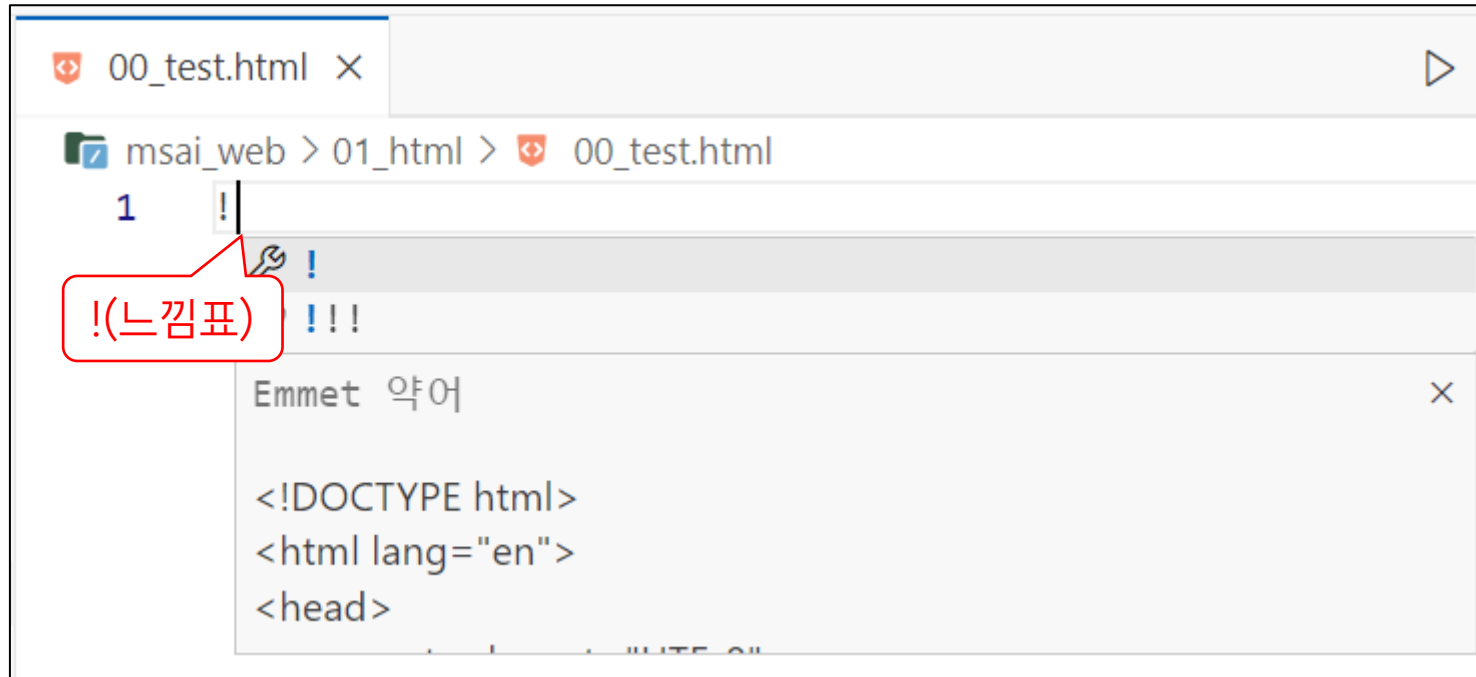
웹문서 만들기

- 새 html 파일 만들기



웹문서 만들기

- html 기본 코드 작성하기



웹문서 만들기

- html 기본 코드 작성하기

html5 기본 구조가 자동으로 완성된다.

```
00_test.html •
frontend > 01_html > 00_test.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9    |
10 </body>
11 </html>
```

웹문서 만들기

- html 기본 코드 작성하기

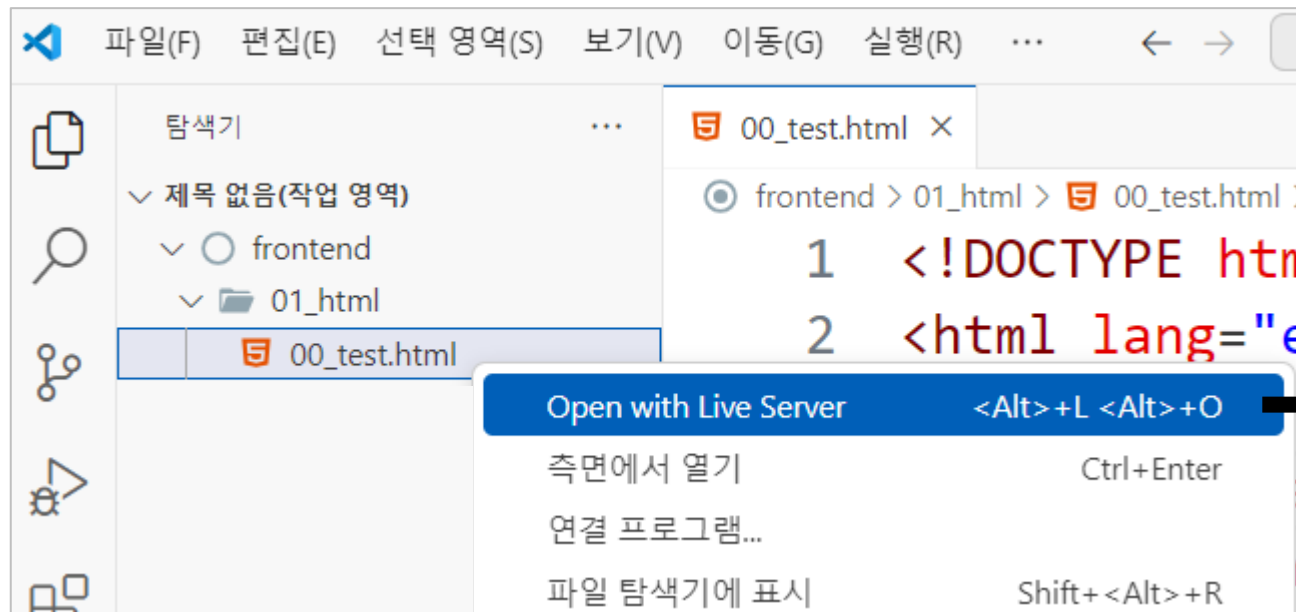
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>문서제목</title>
  </head>
  <body>
    <h1>웹 개발 기초</h1>
    <p>html</p>
    <p>css</p>
    <p>javascript</p>
    <button>시작하기</button>
  </body>
</html>
```

<body></body> 사이에 내용을 작성한다.

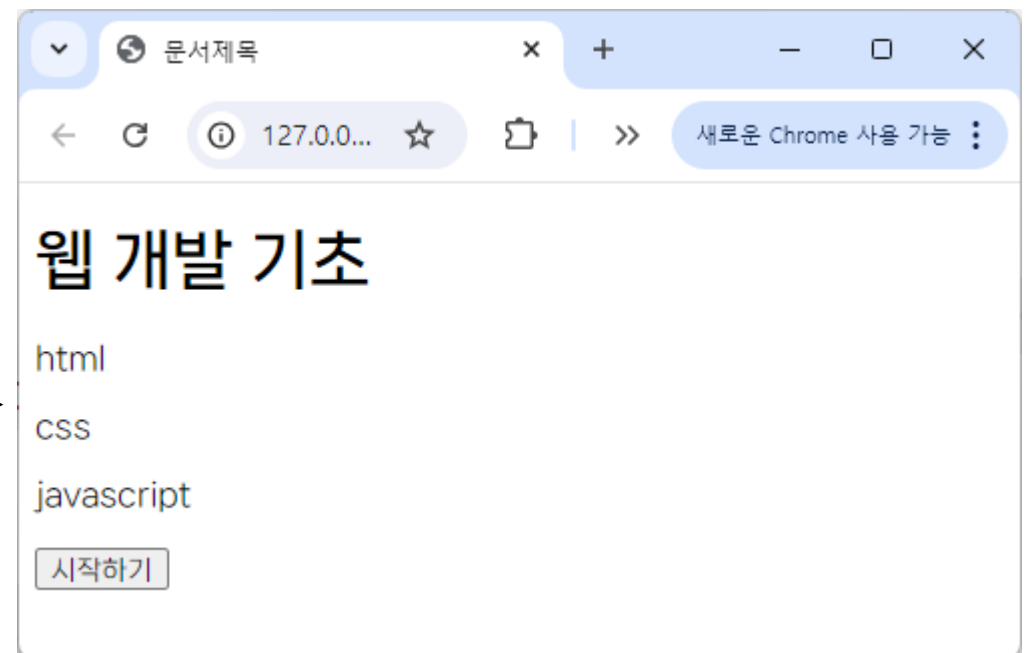
웹문서 만들기

- 라이브서버로 웹페이지 열기

html 파일 위에서 마우스 우클릭 → Open with Live Server 선택

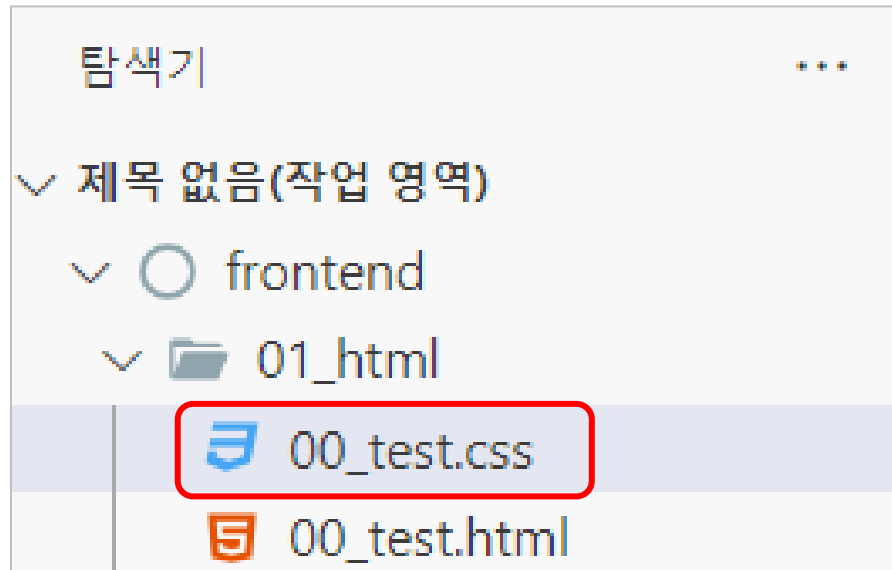


브라우저로 웹문서가 열린다.



웹문서 스타일 지정하기

- CSS 파일 만들기



```
h1 {  
    color: red;  
}  
p {  
    color: blue;  
}  
button {  
    background-color: yellow;  
}
```

00_test.css

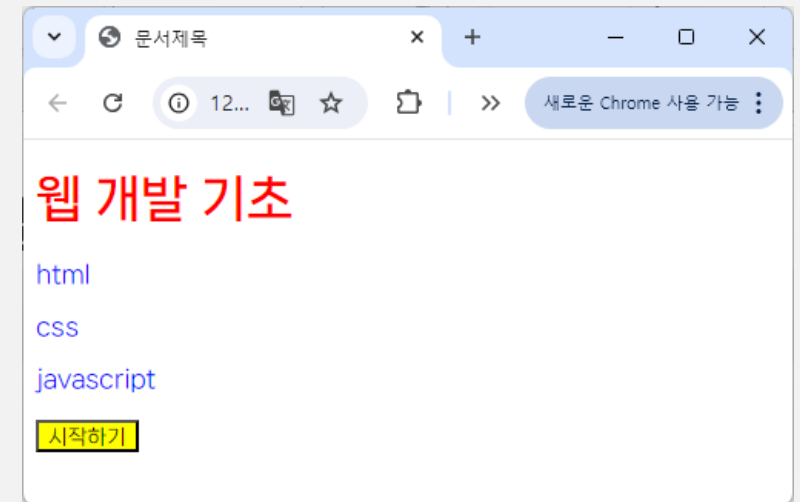
웹문서 스타일 지정하기

- html에 css파일 연결하기(외부 스타일시트)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>문서제목</title>
    <link rel="stylesheet" href="00_test.css" />
  </head>
  <body>
    <h1>웹 개발 기초</h1>
    <p>html</p>
    <p>css</p>
    <p>javascript</p>
    <button>시작하기</button>
  </body>
</html>
```

00_test.html

head에서
css 파일 연결



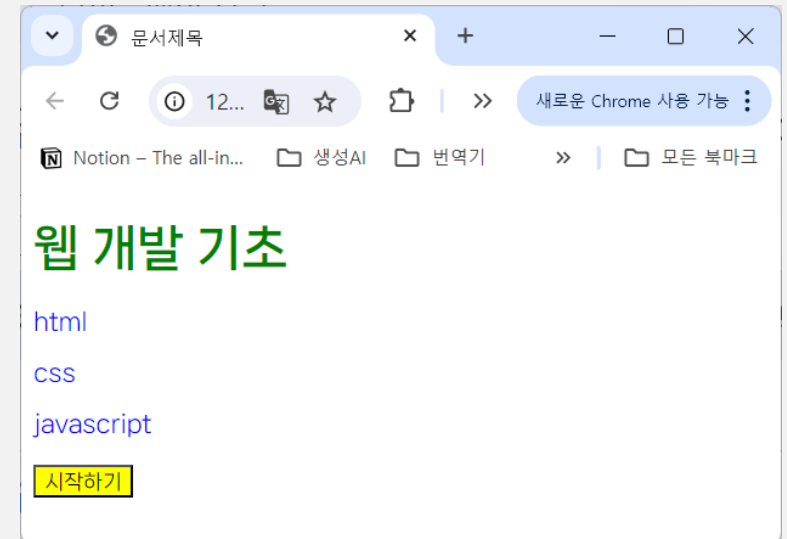
웹문서 스타일 지정하기

- html에서 css 지정하기(내부 스타일시트)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>문서제목</title>
    <link rel="stylesheet" href="00_test.css" />
    <style>
      h1 {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1>웹 개발 기초</h1>
    <p>html</p>
    <p>css</p>
    <p>javascript</p>
    <button>시작하기</button>
    <script src="00_test.js"></script>
  </body>
</html>
```

html 내부에서
css 스타일을 직접 작성

00_test.html



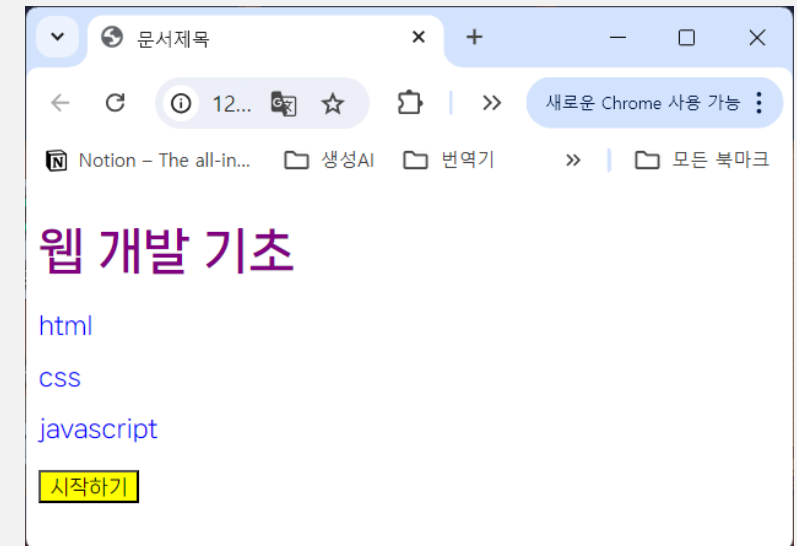
웹문서 스타일 지정하기

- 태그에서 css 지정하기(인라인스타일)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>문서제목</title>
    <link rel="stylesheet" href="00_test.css" />
    <style>
      h1 {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1 style="color: purple">웹 개발 기초</h1>
    <p>html</p>
    <p>css</p>
    <p>javascript</p>
    <button>시작하기</button>
    <script src="00_test.js"></script>
  </body>
</html>
```

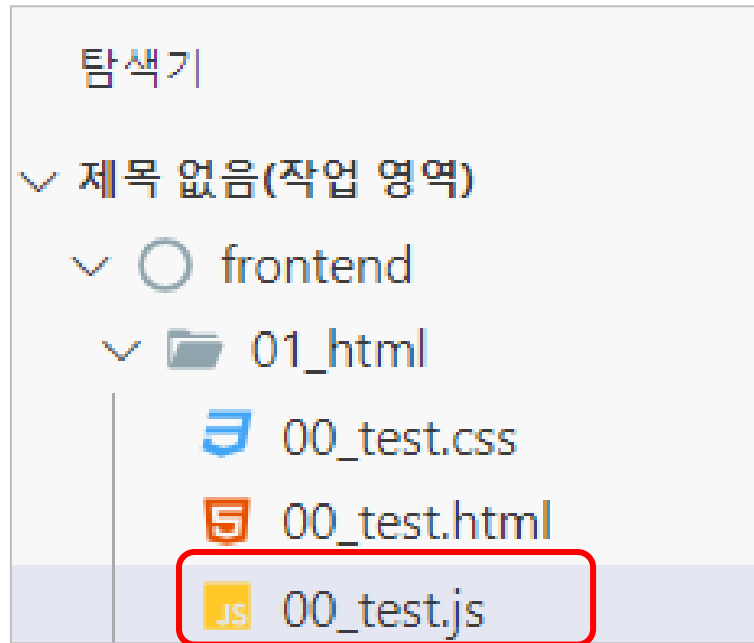
태그에
css 스타일을 직접 작성

00_test.html



웹문서에 동작 추가하기

- js 파일 만들기



00_test.js

```
const btnStart = document.querySelector("button");
btnStart.onclick = function () {
    alert("start");
};
```

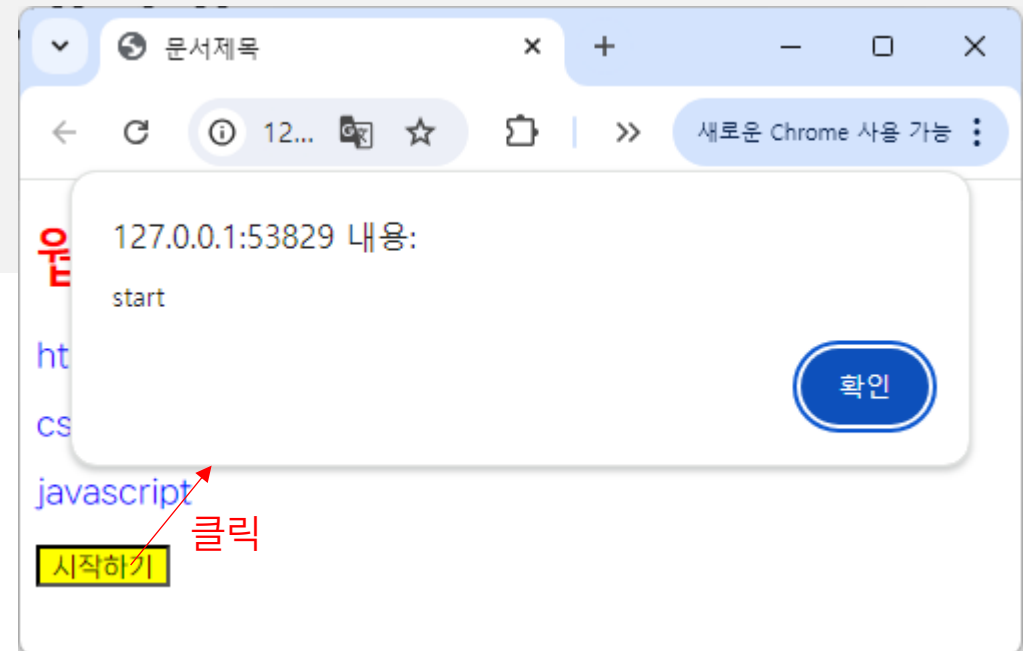
웹문서에 동작 추가하기

- html에 js파일 연결하기(외부 JavaScript 파일)

---생략---

```
<body>
  <h1>웹 개발 기초</h1>
  <p>html</p>
  <p>css</p>
  <p>javascript</p>
  <button>시작하기</button>
  <script src="00_test.js"></script>
</body>
</html>
```

js 파일은 어디서든 연결할 수 있지만
내용이 다 작성된 후 마지막에 body가 닫히기 전인
</body> 위에서 연결하는 것을 권장한다.



웹문서에 동작 추가하기

- html 내부에 script 작성(내부 JavaScript)

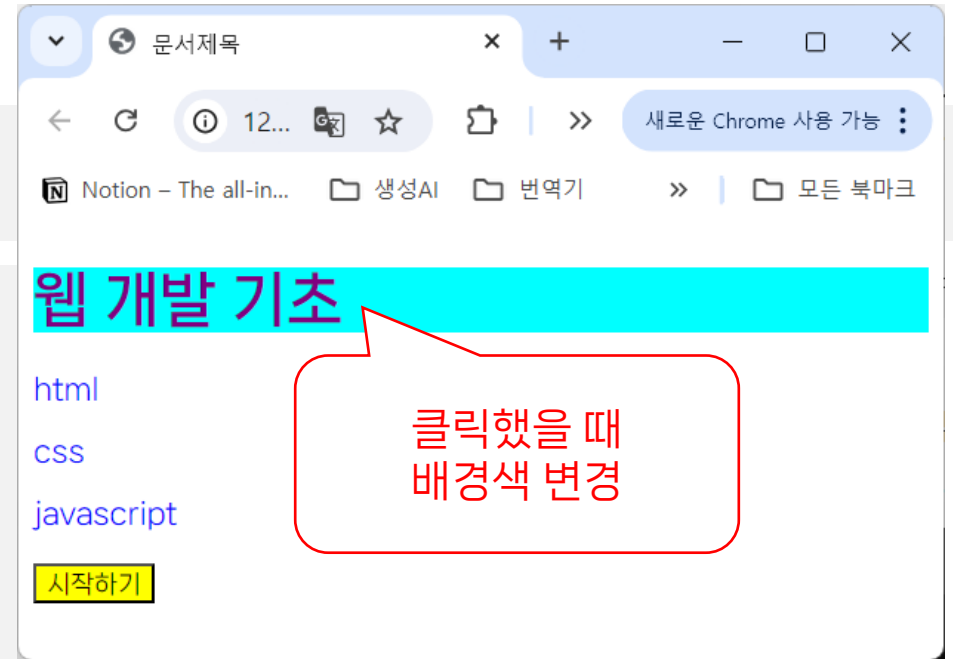
--- 생략 ---

```
<body>
  <h1 style="color: purple">웹 개발 기초</h1>
  <p>html</p>
  <p>css</p>
  <p>javascript</p>
  <button>시작하기</button>
  <script src="00_test.js"></script>
```

```
<script>
  let element_h1 = document.querySelector("h1");
  element_h1.onclick = function () {
    element_h1.onclick = element_h1.style.backgroundColor = "aqua";
  };
</script>
```

```
</body>
</html>
```

<scrip></script> 사이에 작성한다.



웹문서에 동작 추가하기

- 태그에 script 직접 작성(인라인 방식)

```
--- 생략 ---
<body>
  <h1 style="color: purple">웹 개발 기초</h1>
  <p onclick="alert('html')">html</p>
  <p>css</p>
  <p>javascript</p>
  <button>시작하기</button>
  <script src="00_test.js"></script>
  <script>
    let element_h1 = document.querySelector("h1");
    element_h1.onclick = function () {
      element_h1.onclick = element_h1.style.backgroundColor = "aqua";
    };
  </script>
</body>
</html>
```

