

HTML 구조

js, CSS

HTML의 구조

```
<!DOCTYPE html>      ← 문서정의
<html lang="ko">      ← 언어설정
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document Title</title>
  <link rel="stylesheet" href="styles.css"> <!-- CSS 파일 로드 -->
</head>
<body>
  <h1>Hello, World!</h1>
  <p>Sample content here.</p>
  <script src="script.js" defer></script> <!-- JavaScript 파일 로드 -->
</body>
</html>
```

<CSS 사용방법 위치별 차이점>

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

```
<body>
  <link rel="stylesheet" href="styles.css">
</body>
```

```
<head>
  <style>
    body { background-color: lightblue; }
    h1 { color: navy; }
    p { color: darkgray; }
  </style>
</head>
```

위치	렌더링 속도	주의사항
<head>	빠름	FOUC 방지, 대부분의 경우 권장 위치
<body>	느림	페이지가 스타일 없이 먼저 로드될 수 있어 비추천
<style> 또는 인라인 CSS	빠름	한 번만 사용하는 간단한 스타일에 적합하지만, 코드 관리가 어려움

head에 쓴경우 로드 전,
body에 쓴경우 후,
<style> 태그 사용은 동시

<Script 사용법 head 내부>

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>script 속성 예제 (기본, defer, async)</title>

  <!-- 기본 script 태그 -->
  <script src="script-default.js"></script>
  <!-- HTML 파싱이 중단되고 script-default.js가 로드 및 실행됩니다. -->

  <!-- defer 속성 사용 -->
  <script src="script-defer.js" defer></script>
  <!-- HTML 파싱이 완료된 후 script-defer.js가 실행됩니다. 순서 보장. -->

  <!-- async 속성 사용 -->
  <script src="script-async.js" async></script>
  <!-- HTML 파싱과 병행하여 비동기로 로드되며 로드 완료 후 즉시 실행됩니다. 순서 미보장. -->
</head>
<body>
  <h1>Hello, World!</h1>
  <p>이 예제는 기본, defer, async 속성의 차이를 보여줍니다.</p>
</body>
</html>
```

- `script-default.js` : HTML 파싱을 중단하고 이 파일을 로드한 후 바로 실행합니다. 따라서 이 스크립트는 HTML 파싱 도중에 실행됩니다.
- `script-defer.js` : HTML 파싱이 끝난 후에 실행됩니다. 여러 `defer` 스크립트가 있다면 순서대로 실행됩니다.
- `script-async.js` : HTML 파싱과 병행하여 로드되며, 로드가 완료되면 즉시 실행됩니다. 다른 스크립트와의 실행 순서는 보장되지 않습니다.

즉 `<script></script>` 는 html 드로잉중 실행

`<script defer></script>` 는 html 드로잉 후 실행

`<script defer></script>` 는 html 드로잉 병행 실행

<Script 사용법 body 내부>

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>body 내 script 속성 예제 (기본, defer, async)</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>이 예제는 기본, defer, async 속성의 차이를 보여줍니다.</p>

  <!-- 기본 script 태그 -->
  <script src="script-default.js"></script>
  <!-- HTML 파싱이 완료된 후 script-default.js가 로드되고 즉시 실행됩니다. -->

  <!-- defer 속성 사용 -->
  <script src="script-defer.js" defer></script>
  <!-- HTML 파싱이 완료된 후 script-defer.js가 순서대로 실행됩니다. -->

  <!-- async 속성 사용 -->
  <script src="script-async.js" async></script>
  <!-- HTML 파싱과 병행하여 비동기로 로드되며 로드 완료 후 즉시 실행됩니다. 순서 미보장. -->
</body>
</html>
```

1. `script-default.js` 는 HTML 파싱이 완료된 후 바로 로드되고 실행됩니다.
2. `script-defer.js` 는 HTML 파싱이 끝난 후 순서대로 실행되므로, 기본 `<script>` 다음에 실행됩니다.
3. `script-async.js` 는 비동기로 로드되며, 로드가 완료되는 즉시 실행되므로 실행 순서가 보장되지 않습니다.

즉 `<script></script>` 는 html 드로잉 후 실행

`<script defer></script>` 는 html 드로잉 후 기본 이후 실행

`<script defer></script>` 는 html 드로잉 병행 실행

<Script 사용법 정리>

정리

위치	동작 방식	장점 및 단점
<head> 기본	HTML 파싱 중단 후 스크립트 실행	초기 페이지 로딩 느림, DOM 조작이 필요 없는 경우 적합
<head> defer	HTML 파싱 후 스크립트 순서대로 실행	초기 페이지 로딩 빠름, DOM 조작 가능, DOM 조작이 많은 경우 적합
<head> async	HTML 파싱과 병행하여 스크립트 로드, 로드 완료 즉시 실행	로딩 빠르나 실행 순서 보장 안 됨, DOM 조작이 필요 없는 경우 적합
<body> 끝 기본	HTML 파싱 후 스크립트 실행	초기 페이지 로딩 빠름, DOM 조작 가능
<body> defer	HTML 파싱 후 스크립트 순서대로 실행	<head> 에서의 defer와 유사
<body> async	HTML 파싱과 병행하여 스크립트 로드, 로드 완료 즉시 실행	로딩 빠르나 실행 순서 보장 안 됨, DOM 조작이 필요 없는 경우 적합