

자바스크립트 실행 컨텍스트와 이벤트 루프

자바스크립트의 실행 컨텍스트(Execution Context)와 이벤트 루프(Event Loop)는 자바스크립트의 동작 원리를 깊이 이해하고 핵심 메커니즘을 이해하는 데 중요합니다. 실행 컨텍스트는 함수 호출과 스코프와 밀접하게 관련되어 있고, 이벤트 루프는 `setTimeout`, `setInterval`, `Promise`, `async/await` 등의 비동기 프로그래밍과 밀접한 관련이 있습니다.

실행 컨텍스트 (Execution Context)

실행 컨텍스트는 자바스크립트 코드가 평가되고 실행되는 환경 또는 컨텍스트를 의미합니다. 실행 컨텍스트는 코드의 실행 순서와 스코프, 변수, 객체 등의 정보를 관리합니다. 실행 컨텍스트는 크게 세 가지 주요 구성 요소로 이루어져 있습니다.

종류	설명
전역 실행 컨텍스트 (Global Execution Context)	코드가 처음 실행될 때 생성되며, 전역 변수와 함수를 포함합니다.
함수 실행 컨텍스트 (Function Execution Context)	함수가 호출될 때마다 새로운 실행 컨텍스트가 생성됩니다. 이 컨텍스트는 해당 함수의 변수, 매개변수, 함수 내부의 코드 등을 관리합니다.

실행 컨텍스트의 생명주기는 코드가 실행될 때 생성되고, 코드 실행이 끝날 때 소멸됩니다. 이 과정은 실행 컨텍스트 스택, 또는 콜 스택에 의해 관리됩니다. 새로운 함수가 호출될 때마다 해당 함수의 실행 컨텍스트가 스택의 최상위에 푸시되고, 함수의 실행이 끝나면 스택에서 팝(제거)됩니다.

이벤트 루프 (Event Loop)

이벤트 루프는 자바스크립트의 비동기 처리 모델을 가능하게 하는 메커니즘입니다. 자바스크립트는 싱글 스레드 언어로, 한 번에 하나의 작업만 수행할 수 있습니다. 그러나 이벤트 루프를 통해 비동기적인 작업들을 효율적으로 관리하고 실행할 수 있습니다.

- 이벤트 루프의 주요 과정

구성 요소	설명
콜 스택 (Call Stack)	현재 실행 중인 코드의 컨텍스트를 저장합니다. 콜 스택이 비어있을 때까지 실행된 코드의 처리가 계속됩니다.
콜백 큐 (Callback Queue)	비동기 작업(예: <code>setTimeout</code> , <code>setInterval</code> , 이벤트 핸들러 등)에서 발생한 콜백 함수들이 대기합니다.
이벤트 루프 (Event Loop)	콜 스택이 비어있을 때, 콜백 큐에서 대기 중인 콜백 함수를 콜 스택으로 옮깁니다. 이 과정이 반복되면서, 자바스크립트 엔진은 동시성을 달성합니다.

콜 스택(Call Stack): 현재 실행 중인 코드의 컨텍스트를 저장합니다. 콜 스택이 비어있을 때까지 실행된 코드의 처리가 계속 됩니다. 콜백 큐(Callback Queue): 비동기 작업(예: `setTimeout`, `setInterval`, 이벤트 핸들러 등)에서 발생한 콜백 함수들이 대기합니다. 이벤트 루프: 콜 스택이 비어있을 때, 콜백 큐에서 대기 중인 콜백 함수를 콜 스택으로 옮깁니다. 이 과정이 반복 되면서, 자바스크립트 엔진은 동시성을 달성합니다.

실행 컨텍스트와 이벤트 루프의 기본 작동 원리

동기적 코드는 즉시 실행되고, 비동기적 코드는 이벤트 루프에 의해 나중에 실행됩니다.

```
// 스크립트가 시작될 때 바로 실행됩니다. 콜 스택에 들어가고 바로 실행되어 콜 스택에서 제거됩니다.
console.log('스크립트 시작');
```

// 동기적으로 실행되는 함수입니다. 이 함수는 콜 스택에 들어가고, 즉시 실행된 후 콜 스택에서 제거 됩니다.

```
function syncFunction() { // 동기 함수
  console.log('동기 함수 실행');
}
```

// 이 함수 내부의 `setTimeout`은 비동기적인 동작을 합니다.
// `setTimeout`의 콜백 함수는 지정된 시간(여기서는 3000 밀리초)이 지난 후 콜백 큐에 들어갑니다.

```
function asyncFunction() { // 비동기 함수
  // 2초 후, setTimeout의 콜백 함수가 콜백 큐에서 콜 스택으로 이동하여 실행됩니다. 이 시점에
  // 콜 스택은 비어있으므로, 이벤트 루프는 콜백 함수를 콜 스택으로 옮기고 실행합니다.
  setTimeout(() => {
    console.log('비동기 함수 실행');
  }, 3000);
}
```

`syncFunction();` // 동기 함수 호출
`asyncFunction();` // 비동기 함수 호출

// 이 명령은 `asyncFunction` 호출 후 바로 실행됩니다.
// 이 시점에서 콜 스택은 비어있으므로, 이 메시지는 콜백 함수가 실행되기 전에 출력됩니다.

```
console.log('스크립트 종료');
```

// 콘솔 출력 결과:
// 스크립트 시작
// 동기 함수 실행
// 스크립트 종료
// (약 3초 후) 비동기 함수 실행