**1. 자바스크립트는 실행시 컴파일과 런타임 두 단계를 한꺼번에 거친다. 컴파일단계에서 수행되는 작업과 런타임 단계에서 수행되는 작업을 각각 설명하시오.**

**컴파일 단계**

**- 엔진 :** 컴파일레이션 시작부터 끝까지 모든 과정과 프로그램 실행을 책임짐 (런타임 환경을 실질적으로 엔진이 담당하고 있다고 볼 수 있겠습니다.)

**- 컴파일러 :** 파싱, 코드 생성 등(컴파일레이션)

**- 스코프 :** 선언된 변수 목록을 작성하고 유지

출처: <https://yuddomack.tistory.com/entry/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8-%EC%BB%B4%ED%8C%8C%EC%9D%BC%EB%A0%88%EC%9D%B4%EC%85%98%EC%BD%94%EB%93%9C%EC%9D%98-%EC%8B%A4%ED%96%89%EA%B3%BC%EC%A0%95>

**런타임**

프로그래밍 언어가 구동되는 환경

**Call-stack** : 콜스택은 기본적으로 우리가 프로그램의 어디에 있는지를 기록하는 자료 구조입니다. 우리가 함수 안으로 들어가는 순간 해당 함수를 이 스택의 제일 위에 놓게됩니다. 이 함수에서 돌아오면 스택의 가장 윗 부분의 것이 제거됩니다. 스택이 할 수 있는 일은 이것 뿐입니다.

**Web API** : Web API는 자바스크립트 엔진에서 정의되지 않았던 setTimeout이나 HTTP 요청 메서드, DOM 이벤트 등의 메서드를 지원합니다.

**Callback Queue** : 콜백 큐는 Web API 결괏값을 쌓아 두는 큐입니다. 예를 들어 자바스크립트에서 setTimeout(cb, 5000)를 호출하게 되면, Web API는 타이머를 동작시켜 5초 후에 콜백 큐에, cb을 쌓습니다.

**이벤트 루프**: 이벤트 루프는 콜 스택과 콜백 큐를 관찰하는 역할을 합니다. 콜 스택이 비어 있으면 콜백 큐의 첫 번째 콜백을 스택에 쌓습니다.

출처: <https://beomy.github.io/tech/javascript/javascript-runtime/>

**2. var, let, const 변수의 큰 차이점중에 하나가 호이스팅이다.**

**var는 호이스팅 되지만 let,const는 호이스팅 되지 않는다.**

**var와 let, const를 컴파일 과정에서 비교 설명하시오.**

console.log(foo) // undefined

var foo

console.log(bar) // Error: Uncaught ReferenceError: bar is not defined

let bar

console.log(bar) // Error: Uncaught ReferenceError: bar is not defined

const bar

**3. $("#id) 로 엘리먼트 접근시 수행되는 제이쿼리 내부구조를**

**설명하시오.**

**4. css 스타일의 상속과 적용우선순위에 대해 설명하시오.**

**css 스타일의 상속** - 상속이란 상위(부모, 조상) 요소에 적용된 프로퍼티를 하위(자식, 자손) 요소가 물려 받는 것을 의미

**출처:** <https://poiemaweb.com/css3-inheritance-cascading>

**5. css로 수평 중앙정렬, 수직중앙정렬 하는 법을**

**각각 아는대로 설명하시오.**

* Text-align 사용
* Position 사용
* Table, table-cell 사용
* Transform(translate 사용)
* Flex 사용

**6. React Component에서 store의 변경, Rerendering with changed props까지 전과정이 어떻게 되는가?**

**Action -> dispatch -> reducer -> store -> prop 전달**

**7. Redux Middleware의 역할이 무엇인가? 아는대로 쓰시오.**



미들웨어는, 액션이 디스패치(dispatch) 되어서 리듀서에서 이를 처리하기전에 사전에 지정된 작업들을 설정합니다. 미들웨어를 액션과 리듀서 사이의 중간자라고 볼 수 있다.

**출처**: <https://velopert.com/3401>