

### 자신의 강점과 약점은 무엇인가요?

저의 강점은 새로운 기술에 대한 열린 마음과 적극적인 학습 태도입니다. 또한, 문제 해결 능력과 논리적 사고 능력도 강점 중 하나입니다. 반면에 제 약점은 디테일한 부분에 대한 주의 부족입니다. 이를 보완하기 위해 더욱 신중하게 작업하고, 팀원들과 함께 코드 리뷰를 진행하는 등의 방법을 사용하고 있습니다.

-> 자신의 약점에 대한 보완을 얼마나 성실하고 지속적으로 하고 있는지에 대한 내용이 중요합니다.

### Flex와 Grid의 차이점에 대해서 설명해보세요.

Flex는 주로 단일 축 방향의 레이아웃을 구성하는 데 사용되며, 주로 수평 방향으로 레이아웃을 정렬합니다. Grid는 2차원 그리드 시스템을 구성하는 데 사용되며, 수평과 수직 방향으로 모두 레이아웃을 정렬할 수 있습니다. Flex는 아이템들의 크기가 자유로워 유동적으로 변할 수 있습니다. 반면에 Grid는 아이템들의 크기를 미리 정의하여 각 셀의 크기를 일정하게 유지합니다. Flex는 유동적인 레이아웃 변화를 구성하기 적합합니다. 반면에 Grid는 미리 정의된 2차원 그리드 구조를 유지하기 때문에 레이아웃 변화가 적은 경우에 적합합니다.

-> 언제 Flex를 쓰고 언제 Grid를 쓸지에 대해 아는 것이 가장 중요합니다.

### "==" 와 "==="의 차이는 무엇인가요?

"=="는 동등 연산자로, 두 값을 비교할 때 형변환(type coercion)을 수행합니다. 즉, 비교하는 값의 데이터 타입이 다르더라도 자동으로 형변환을 수행한 후에 비교합니다. 이러한 형변환은 때로 예측하지 못한 결과를 초래할 수 있으므로, "=="를 사용할 때는 조심해야 합니다. "==="는 일치 연산자로, 두 값이 데이터 타입과 값이 모두 같은지 비교합니다. 따라서 "==="를 사용하면 형변환 없이 정확한 값을 비교할 수 있습니다. 이러한 일치 연산자를 사용하는 것이 더 안전하고 예측 가능한 결과를 얻을 수 있습니다.

-> 헛갈리지 않게 핵심만 파악해보세요. 동등! 일치!

### 가장 최근에 해결한 기술적인 문제는 무엇이었나요?

최근에는 프로젝트에서 발생한 성능 이슈를 해결하는 문제를 해결했습니다. 문제 해결을 위해 코드를 리팩토링하고, 불필요한 부분들을 최적화하여 크롬 LightHouse를 통해 높은 점수를 받을 수 있었습니다.

-> 본인이 프로젝트에서 직접적으로 개선한 부분들을 말씀하시면 됩니다. 평소에 리팩토링 습관이 있으시다면 대답하시기 수월하실 거라 생각합니다.

### 브라우저 렌더링 원리에 대해서 설명해보세요.

브라우저 렌더링은 HTML, CSS, JavaScript 등의 웹 페이지 자원을 브라우저가 화면에 그리는 과정을 말합니다. 브라우저 렌더링 원리와 순서는 크게 다음과 같은 단계로 구성됩니다.

먼저 DOM을 생성합니다. 브라우저는 HTML 문서를 파싱하여 DOM 트리를 생성합니다. 이때, HTML 태그를 노드로 변환하고, 노드간의 계층 관계를 형성합니다.

두 번째로 CSSOM을 생성합니다. 브라우저는 CSS 파일을 파싱하여 CSSOM 트리를 생성합니다. 이때, CSS 속성을 노드로 변환하고, 노드간의 계층 관계를 형성합니다.

세 번째로 DOM 트리와 CSSOM을 결합하여 렌더 트리를 생성합니다. 이때, 실제 화면에 표시될 요소만을 선택하여 렌더 트리를 형성합니다.

이제, 브라우저는 렌더 트리를 이용하여 각 요소의 크기와 위치를 계산하는 과정인 레이아웃을 거쳐 화면에 요소를 그리는 페인팅 과정을 거치게 됩니다. 이때, 요소의 배경, 테두리, 글자 등을 그리게 됩니다.

-> 전체적인 과정을 머릿 속에 그려놓고, 꼭 위와 같이 설명하지 않아도 흐름을 대략적으로 설명하실 수 있다면 대답하셔도 충분할 것이라 생각합니다.

### async/await에 대해 설명해보세요.

async/await 는 비동기적인 작업을 처리할 수 있는 ES2017 문법입니다. async 함수를 정의하면 함수 내부에서 await 키워드를 이용하여 비동기적으로 처리되는 작업이 완료될 때까지 기다린 후, 결과값을 반환하는 처리를 할 수 있습니다. async/await 는 Promise 를 기반으로 하며, 코드를 보다 간결하고 직관적으로 작성할 수 있도록 해줍니다. async 함수는 항상 Promise 객체를 반환하며, await 키워드를 이용하여 비동기 처리 결과를 기다립니다.

-> 비동기 함수이며, 비동기가 어떤 뜻인지도 설명할 준비가 되어 있으셔야 합니다.

#### 이벤트 버블링에 대해서 설명해주세요.

이벤트 버블링(event bubbling)은 HTML 에서 이벤트가 발생했을 때, 해당 요소에서 이벤트가 처리된 후, 상위 요소로 이벤트가 전파되는 현상을 말합니다. 이벤트 버블링은 이벤트가 전파되는 동안 부모 요소에서도 이벤트를 감지할 수 있다는 장점이 있습니다. 즉, 이벤트를 처리하는 핸들러 함수를 부모 요소에 등록해놓으면, 자식 요소에서 발생한 이벤트도 모두 처리할 수 있습니다. 이를 활용하면, 여러 개의 하위 요소에서 발생하는 이벤트를 하나의 이벤트 핸들러로 바인딩하는 이벤트 위임 처리를 할 수 있습니다. 버블링 및 캡처링을 방지하기 위해서는 이벤트 객체의 stopPropagation() 메서드를 사용하여 이벤트 전파를 중지시키거나, 이벤트 핸들러에서 이벤트가 발생한 요소를 확인하여 처리해야 합니다.

-> 이벤트 버블링과 캡처링의 차이점에 대해서 구분해서 인지하는 것이 중요합니다.

-> 이벤트 위임에 대해서도 인지하고 있어야 합니다.

#### 비동기 함수에 대해서 설명해 보세요.

비동기 함수(Asynchronous function)는 함수의 실행 결과가 즉시 반환되지 않고, 특정 조건이 충족될 때까지 기다리는 함수입니다. 비동기 함수는 일반적으로 콜백 함수나 Promise 객체를 반환합니다. 콜백 함수는 비동기 작업이 완료되었을 때 호출되는 함수이며, Promise 객체는 비동기 작업이 성공적으로 완료되었는지 또는 실패했는지를 나타내는 객체입니다. 비동기 함수는 애플리케이션의 성능을 향상시키고, 사용자 인터페이스의 반응성을 유지하는 데 유용합니다. 그러나 비동기 함수를 사용할 때는 콜백 지옥과 같은 문제를 피하기 위해 적절한 에러 처리와 코드 구성이 필요합니다. 여러개의 비동기 함수가 실행되면, 이벤트 루프는 비동기 함수 호출을 처리하고, 비동기 함수가 완료되었을 때 콜백 함수를 호출합니다. 이벤트 루프는 실행 대기 중인 비동기 함수가 있으면 해당 함수를 호출하고, 실행이 완료될 때까지 다음 비동기 함수를 호출합니다. 따라서, 여러 개의 비동기 함수가 실행될 때 이들 함수는 동시에 실행되며, 이벤트 루프에 의해 비동기 함수의 실행 순서가 제어됩니다.

-> 가장 중요한 질문이라고 생각되네요. 이벤트 루프에 대해서도 꼭 숙지하셔서 좋은 결과 있기를 바랍니다.

#### map 과 forEach 의 차이르 설명해 주세요.

map 은 원본 배열을 변경하지 않고 새로운 배열을 생성합니다.

forEach 는 반환 값이 없으며, 원본 배열을 직접 변경합니다.

#### var, let, const 의 차이를 설명해 주세요.

[var]

- 중복 선언 가능
- 함수레벨 스코프

[let]

- 중복 선언 불가능
- 재할당 가능
- 블록레벨 스코프

[const]

- 중복 선언 불가능
- 재할당 불가능
- 블록레벨 스코프

## 자카스크립트 데이터 타입에 대해 설명해 주세요.

[원시 타입(primitive data type)]

- boolean
- null
- undefined
- number
- string
- symbol(ES6 에서 추가)

[객체/참조 타입(object/reference type)

- object

## URL 과 URI 의 차이

[URI] : 통합 자원 식별자(Uniform Resource Identifier)

URI 는 인터넷상의 리소스(자원) 자체를 식별하는 고유한 문자열 시퀀스입니다.

[URL] : 통합 자원 위치(Uniform Resource Location)

URL 은 네트워크상의 리소스(자원)의 위치를 나타내기 위한 규약입니다.

[예시]

URI=식별자, URL=식별자+위치

- <https://www.example.com/index.html?page=123>

위의 링크에서 index.html 의 위치를 표기한 <https://www.example.com/index.html> 까지는 URL 입니다.

여기서 query string 으로 식별자가 포함된 <https://www.example.com/index.html?page=123> 은 URI 입니다.