

HTML,CSS,JavaScript의 기본 개념과 특징은 무엇인가?

HTML의 개념과 특징

- 개념

HyperText Markup Language의 약자로 웹 페이지의 구조를 정의하는 마크업 언어. 웹 페이지의 내용을 표현하는데 사용되며, 태그를 이용해 문서의 구조를 표현. 헤더, 텍스트, 이미지, 리스트, 링크 등을 표현할 수 있다.

- 특징

1. 구조화된 문서

- HTML은 웹 페이지의 구조를 구성하는 마크업 언어로, 여러 태그를 사용하여 문서의 구조를 표현한다. 각 태그는 서로 다른 의미와 기능을 가지고 있으며, 웹 페이지의 헤더, 본문, 링크, 이미지 등을 표현할 수 있다.

2. 플랫폼 독립적

- HTML은 여러 웹 브라우저와 운영체제에서 호환되는 플랫폼 독립적인 언어이다. 이는 웹 개발자가 다양한 환경에서 동일한 결과를 보장할 수 있게 해준다.

3. 간결한 문법

- HTML은 직관적이고 간결한 문법을 가지고 있다. 이는 웹 개발자가 빠르게 웹 페이지를 작성하고 이해할 수 있는 장점이 있다.

4. 웹 접근성

- HTML은 웹 접근성에 중요한 역할을 한다. 적절한 HTML 태그를 사용하여 문서의 구조를 명확하게 표현함으로써, 스크린 리더와 같은 보조 기술을 사용하는 사용자에게 웹 페이지의 콘텐츠를 올바르게 전달할 수 있다.

5. 확장성

- HTML은 확장성이 뛰어나, 개발자들은 기존의 HTML 태그를 사용하여 웹 페이지를 작성할 수 있을 뿐만 아니라, 웹 컴포넌트 기술을 사용하여 사용자 정의 태그를 생성하여 재사용 가능한 웹 요소를 만들 수도 있다.

웹 컴포넌트(Web Components)란?

웹 개발에서 사용되는 기술 모음. 이 기술들은 사용자 정의 요소(Custom Elements), HTML 템플릿(Templates), Shadow DOM, 그리고 HTML 임포트(Imports)로 구성되어 있다. 웹 컴포넌트의 주요 목적은 재사용 가능하고 캡슐화된 커스텀 웹 페이지 요소를 만드는 것.

1. 사용자 정의 요소(Custom Elements): 개발자가 자신만의 HTML 태그를 정의하고 해당 태그의 기능과 동작을 구현할 수 있게 한다.
2. HTML 템플릿(Templates): 화면에 직접 렌더링되지 않지만, 동적으로 콘텐츠를 추가할 수 있는 템플릿을 정의할 수 있게 한다.
3. Shadow DOM: 웹 컴포넌트의 내부 구조와 스타일을 캡슐화하여, 외부의 스타일과 충돌하지 않게 한다. 이를 통해 컴포넌트가 독립적으로 작동하게 할 수 있다.

웹 컴포넌트를 사용하면, 웹 페이지에 재사용 가능한 구성 요소를 쉽게 추가하고 유지 관리할 수 있으며, 다양한 프레임워크와 라이브러리와의 호환성도 높아진다.

6. 시맨틱 태그

- HTML5부터는 시맨틱 태그가 도입되어 웹 페이지의 구조를 더욱 명확하게 표현할 수 있게 되었다. 이러한 시맨틱 태그들은 웹 페이지의 의미를 명확하게 전달하며, 검색 엔진 최적화(SEO)에도 긍정적인 영향을 미친다.

CSS의 개념과 특징

• 개념

Cascading Style Sheets의 약자로, 웹 페이지의 디자인 요소를 제어하는 스타일 언어. CSS는 웹 페이지의 레이아웃, 폰트, 색상 등을 조절할 수 있다. 이를 통해 중복 코드를 줄이고 웹 페이지의 유지 관리를 용이하게 한다. 또한, HTML과 CSS는 서로 분리되어 있어 콘텐츠와 디자인이 서로 영향을 주지 않는다.

• 특징

1. 스타일 지정

- CSS는 웹 페이지의 스타일을 지정하는 데 사용되며, 폰트, 색상, 배경, 레이아웃 등의 디자인 요소를 제어할 수 있다. 이를 통해 웹 페이지의 디자인을 사용자에게 적합한 형태로 제공할 수 있다.

2. 콘텐츠와 디자인의 분리

- CSS는 HTML과 분리되어 작성되어, 콘텐츠와 디자인이 서로 영향을 주지 않는다. 이를 통해 웹 페이지의 유지 관리가 용이해지며, 중복 코드를 줄일 수 있다.

3. 선택자를 이용한 스타일 적용

- CSS는 선택자를 사용하여 HTML 요소에 스타일을 적용한다. 이를 통해 특정 요소 또는 요소 그룹에 스타일을 일관되게 적용할 수 있다.

4. 상속과 우선순위

- CSS는 상속과 우선순위를 통해 스타일을 적용하는 방식을 결정한다. 상속은 부모 요소의 스타일이 자식 요소에 적용되는 원리이며, 우선순위는 여러 스타일 규칙이 충돌할 때 어떤 규칙이 적용될지 결정하는 방식이다.

5. 반응형 디자인

- CSS는 반응형 디자인을 지원하여, 다양한 화면 크기와 장치에서 웹 페이지가 올바르게 표시되도록 할 수 있다. 이를 위해 미디어 쿼리, 그리드 레이아웃, 플렉스 박스 등의 기능을 사용할 수 있다.

6. CSS 전처리기

- CSS 전처리기를 사용하면, 변수, 함수, 중첩, 믹스인 등의 고급 기능을 사용하여 CSS 코드를 작성할 수 있다. 이는 웹 페이지의 스타일을 더 효율적이고 일관성 있게 관리할 수 있게 해준다.

JavaScript의 개념과 특징

• 개념

JavaScript는 웹 페이지에 동적인 기능을 추가하는 스크립트 언어이다. 웹 페이지의 동작을 제어하고, 클라이언트 측에서 실행되어 사용자와 상호작용을 가능하게 한다. 예를 들어, 이벤트 처리, 애니메이션, 폼 유효성 검사 등의 작업을 수행할 수 있다. 또한, 최근에는 Node.js와 같은 서버 측 플랫폼 덕분에 백엔드 개발에서도 활용되고 있다.

• 특징

1. 인터프리터 언어

- JavaScript는 인터프리터 언어로, 컴파일 과정 없이 코드를 바로 실행할 수 있다. 이는 개발 과정에서 시간을 절약하고, 즉각적인 피드백을 받을 수 있는 장점이 있다.

2. 동적 타이핑

- JavaScript는 동적 타이핑 언어로, 변수의 타입을 선언하지 않아도 된다. 변수의 타입은 프로그램이 실행되는 도중에 자동으로 결정된다. 이는 코드의 간결함을 유지하고 빠르게 개발할 수 있는 이점이 있지만, 예기치 않은 오류를 유발할 수 있는 단점도 있다.

3. 객체 지향 프로그래밍

- JavaScript는 객체 지향 프로그래밍 패러다임을 지원하며, 프로토타입 기반의 상속을 사용한다. 클래스 기반 언어와 달리, 객체를 바로 생성하거나 기존 객체로부터 새로운 객체를 생성할 수 있다.

4. 이벤트 드리븐

- JavaScript는 이벤트 드리븐 프로그래밍 모델을 따르며, 이벤트 리스너를 등록해 특정 이벤트가 발생했을 때 동작하도록 코드를 작성할 수 있다. 이를 통해 사용자와의 상호작용을 원활하게 처리할 수 있다.

이벤트 드리븐(event-driven)이란?

이벤트 드리븐 프로그래밍은 프로그램의 실행 흐름이 외부 이벤트에 의해 결정되는 프로그래밍 패러다임. 이벤트는 사용자 입력, 시스템 메시지, 네트워크 요청 등 다양한 형태로 발생할 수 있다. 이벤트 드리븐 프로