

03

JavaScript의 실행 환경 및 디버깅





학습내용

- 01 JavaScript 실행 환경
- 02 웹 브라우저
- 03 Node.js



학습목표

- JavaScript의 실행 환경을 이해하고, 웹 브라우저와 Node.js의 차이점을 설명할 수 있다.
- 웹 브라우저에서 JavaScript를 실행하고, 개발자 도구의 콘솔과 디버깅 기능을 활용할 수 있다.
- Node.js의 기본 개념과 실행 방법을 익히고, JavaScript가 서버 측에서도 사용될 수 있음을 설명할 수 있다.



01



JavaScript 실행 환경



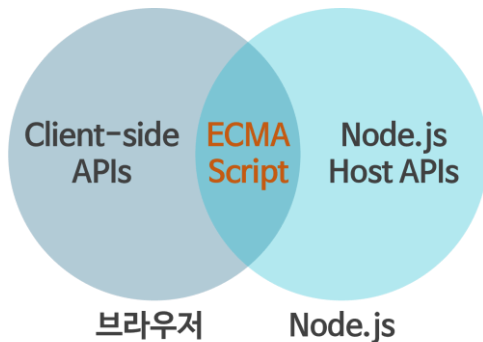


모든 브라우저와 Node.js는 JavaScript를 해석하고 실행할 수 있는 JavaScript 엔진을 내장하고 있음



이 둘은 코드가 동일하게 동작하지만, 용도가 다름

브라우저와 Node.js는 JavaScript의 코어인 ECMAScript를 실행할 수 있지만, ECMAScript 이외에 추가로 제공하는 기능은 호환하지 않음



브라우저

- 주 목적
 - HTML, CSS, JavaScript를 시행해 웹 페이지를 브라우저 화면에 렌더링하는 것
- ECMAScript와 DOM API 등의 클라이언트 사이드 Web API를 기본적으로 제공

Node.js

- 주 목적
 - 브라우저 외부에서 JavaScript 실행 환경을 제공하는 것
- ECMAScript와 파일 시스템 등의 고유 API를 기본적으로 제공

웹 브라우저

- 구글 크롬 브라우저의 V8 JavaScript 엔진
 - Node.js에서도 사용 중



02

웹 브라우저



{ 모든 브라우저에는 ECMAScript 실행엔진을 보유하고 있음 }



구글 크롬 브라우저 사용 예정

- 크롬은 ECMAScript 사양을 준수하는 것은 물론이고, 시장 점유율 65.47%로 가장 큼



구글 크롬 브라우저 다운로드

- https://www.google.com/intl/ko_kr/chrome/

“ 브라우저에 기본 내장되어 있음 ”

운영체제별 단축키

Window

F12 또는

Ctrl + Shift + I

macOS

Command ⌘ +
Option (⌥) + I

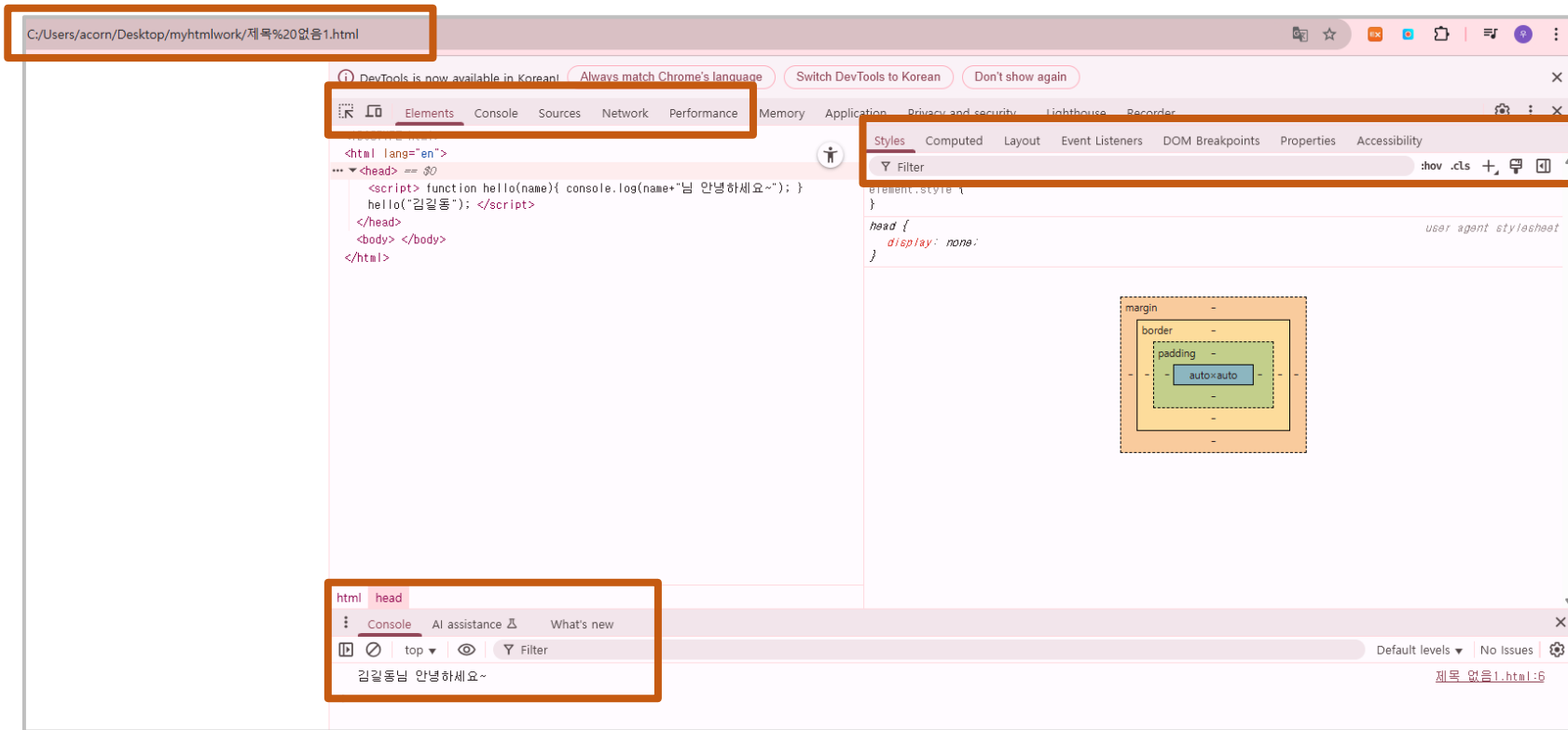
- 제공하는 기능

메모장이나 VSCode와 같은 도구를 활용해
HTML 문서를 작성 후, 웹 브라우저에서 코드를 실행하는 경우

Editplus로 문서를 작업한 화면

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <script>
5       function hello(name){
6         console.log(name+"님 안녕하세요~");
7       }
8
9       hello("김길동");
10    </script>
11  </head>
12  <body>
13
14  </body>
15 </html>
```

● 제공하는 기능



● 제공하는 기능

1/2

패널	설명
Element	<ul style="list-style-type: none">로딩된 웹 페이지의 DOM과 CSS를 편집해서 렌더링된 뷰를 확인해 볼 수 있음편집된 내용이 저장되지는 않음웹 페이지가 의도된 대로 렌더링 되지 않았을 때 힌트를 얻을 수 있음
Console	<ul style="list-style-type: none">로딩된 웹 페이지의 에러를 확인할 수 있음JavaScript 소스 코드에 작성한 console.log() 메서드의 실행 결과를 확인할 수 있음
Sources	<ul style="list-style-type: none">로딩된 웹 페이지의 JavaScript 코드를 디버깅할 수 있음

● 제공하는 기능

2/2

패널	설명
Network	<ul style="list-style-type: none">로딩된 웹 페이지에 관련된 네트워크 요청(Request) 정보와 성능을 확인할 수 있음
Application	<ul style="list-style-type: none">웹 스토리지, 세션, 쿠키를 확인하고 관리할 수 있음

3) 콘솔(Console)



개발자 도구의 Console 패널 (이하 콘솔)

에러 확인용

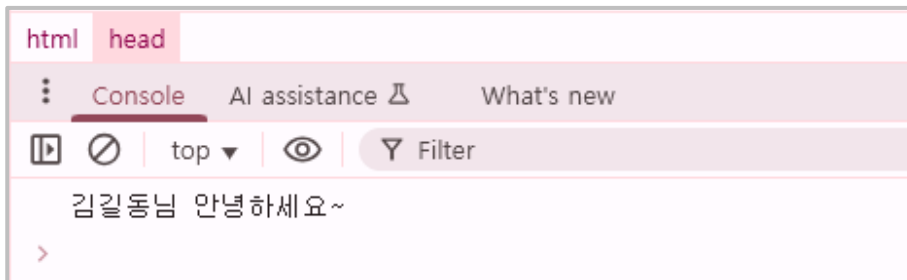


console.log(...) 사용

- 소괄호 안의 코드를 평가해서 그 결과를 콘솔에 출력하는 메서드



REPL(Read Eval Print Loop) 환경으로 사용

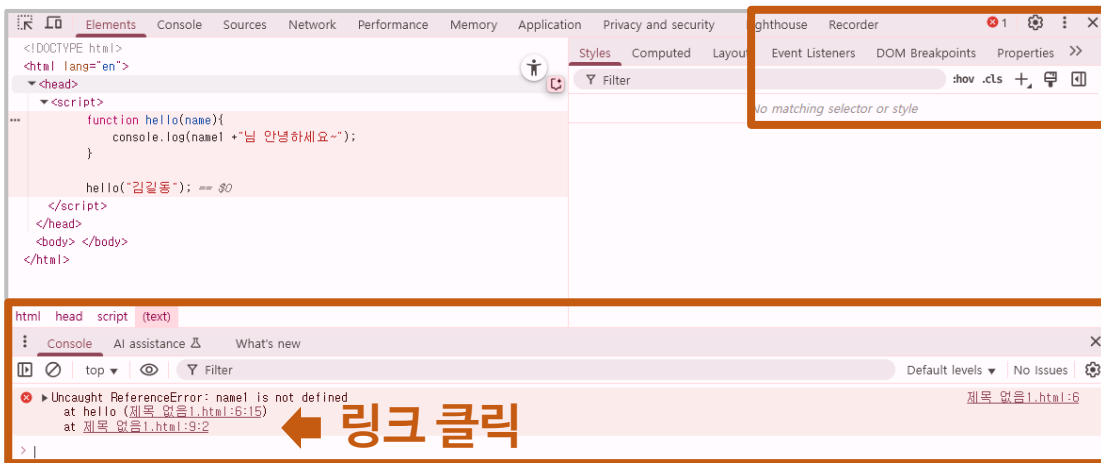


4) 디버깅(Debugging) 활용

- 웹 브라우저에서 코드 실행 후 에러가 발생하는 경우

01 에러 정보의 오른쪽에 에러 발생 위치를 나타내는 링크 클릭

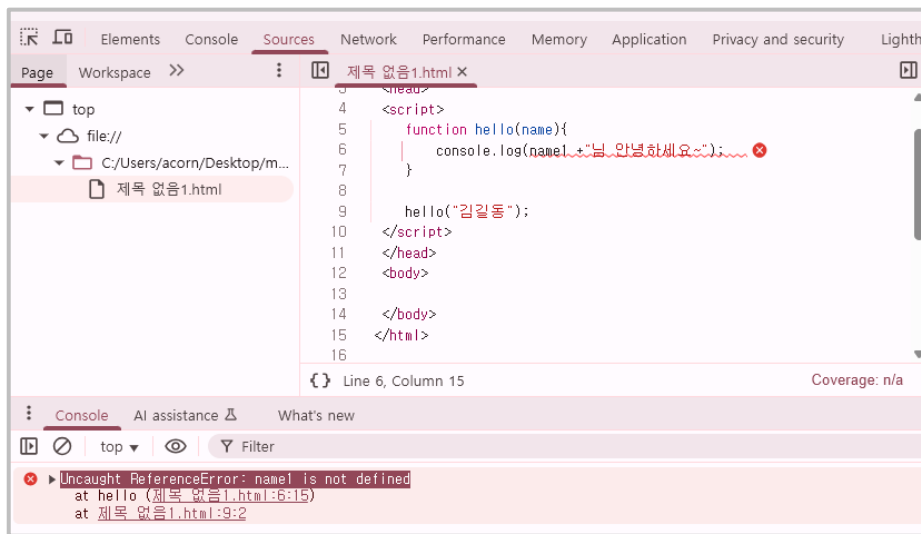
02 JavaScript 코드를 디버깅할 수 있는 Sources 패널로 이동



● 웹 브라우저에서 코드 실행 후 에러가 발생하는 경우

03

에러 발생한 위치에 빨간 밑줄이 표시되고, 그 위에 마우스를 올려보면
“제목 없음1.html:6 Uncaught ReferenceError: name1 is not defined”
에러 정보 표시

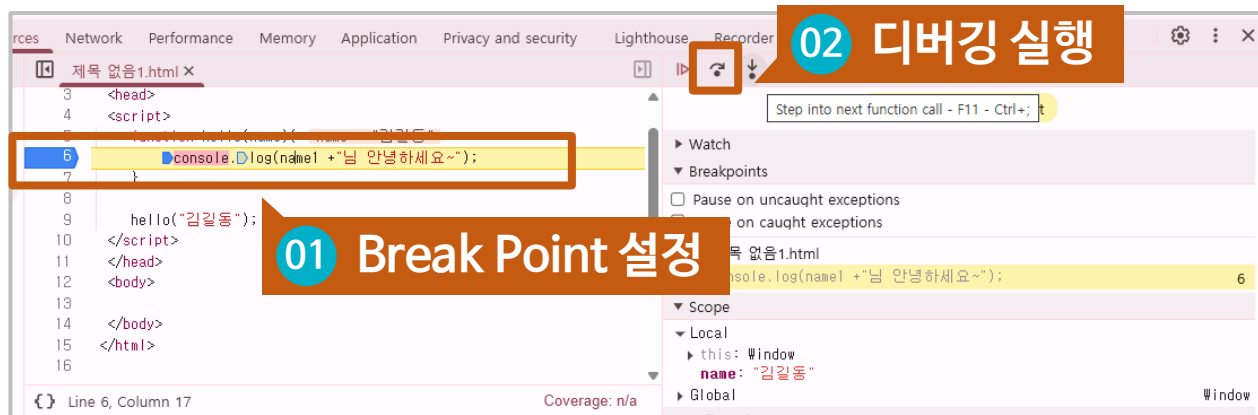


● 웹 브라우저에서 코드 실행 후 에러가 발생하는 경우

04

우선 name1의 변수 값이 null인지 확인해보고, null이라면 그 이유를 알아내서
에러가 발생한 원인을 제거해보기

- 에러가 발생한 코드 왼쪽의 라인 번호를 클릭해 브레이크 포인트(중단점)을
걸고 (01) 브라우저를 다시 실행 하면 다음과 같이 디버깅 모드로 들어감
- 이후 한 명령어씩 수행하려면 다음 버튼을 눌러 디버깅을 수행함 (02)



4) 디버깅(Debugging) 활용



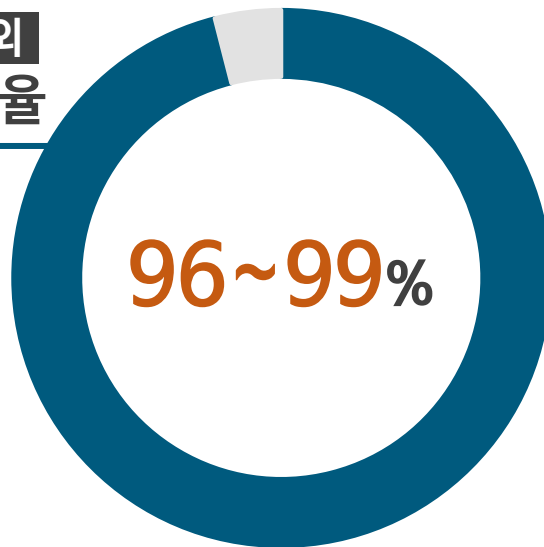
● 크롬 도구에서 제공하는 디버깅을 위한 단축키 (Window 기준)

기능	단축키
개발자 도구 열기/닫기	F12 또는 Ctrl + Shift + I
요소 선택 모드 (DOM 탐색기)	Ctrl + Shift + C
콘솔 탭 열기	Esc (또는 개발자 도구 내 Console 클릭)
검색 (파일, 함수 등)	Ctrl + P
심볼/함수 검색	Ctrl + Shift + P
현재 페이지 내 텍스트 검색	Ctrl + F
브레이크포인트 설정/해제	코드 줄 번호 클릭
한 줄씩 코드 실행 (Step Over)	F10
함수 내부로 진입 (Step Into)	F11
함수 밖으로 빠져나오기 (Step Out)	Shift + F11
다음 브레이크포인트까지 실행 (Resume)	F8
콘솔에서 명령 실행	Esc → Console 열기

5) ES6 브라우저 지원 현황



인터넷 익스플로러 제외
모든 브라우저의 ES6 지원 비율



참고

인터넷 익스플로러나 구형 브라우저는 대부분 ES6를 지원하지 않음

Q

브라우저에서 아직 지원하지 않는 최신 기능을 사용하거나,
인터넷 익스플로러나 구형 브라우저를 고려해야 하는 상황이라면?

바벨(Babel)과 같은 트랜스파일러를 이용해 ES6 이상의 사양으로 구현한 소스코드를
ES5 이하의 사양으로 다운그레이드 할 필요가 있음

ES6 classes - OTHER

Usage: Global 97.18% + 0.06% = 97.24%

Baseline Widely available across major browsers

ES6 classes are syntactical sugar to provide a much simpler and clearer syntax to create objects and deal with inheritance.

Current aligned Usage relative Date relative Filtered All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-41				10-28												
42-48	12	3.1-8	2-44	29-35			3.2-8.4	4								
49-135	13-134	9-18.3	45-137	36-116	6-10		9-18.3	5-26		12-12.1		2.1-4.4.4				2.5
136	135	18.4	138	117	11	135	18.4	27	all	80	15.5	135	137	14.9	13.52	3.1
137-139		18.5-TP	139-141				18.5									

초록색 버전에서 ES6 지원



03

Node.js





2009년,
라이언 달이
발표

Node.js

구글 V8 JavaScript 엔진으로 빌드된 JavaScript 런타임 환경

브라우저의 JavaScript 엔진에서만 동작하던 JavaScript를
브라우저 이외의 환경에서도 동작할 수 있도록 JavaScript 엔진을
브라우저에서 독립시킨 JavaScript 실행 환경

1) 설치



Node.js® 다운로드

Node.js® v22.15.0 (LTS) 를 Windows 환경에서 fnm 방식으로 npm

를(을) 사용해 설치하세요.

```
1 # fnm 다운로드 및 설치:  
2 winget install Schniz.fnm  
3  
4 # Node.js 다운로드 및 설치:  
5 fnm install 22  
6  
7 # Node.js 버전 확인:  
8 node -v # "v22.15.0"가 출력되어야 합니다.  
9  
10 npm 버전 확인:  
11 npm -v # 10.9.2가 출력되어야 합니다.
```

PowerShell

클립보드에 복사

"fnm"은 플랫폼 간 Node.js 버전 관리자입니다. 문제가 발생하면 [fnm's 웹사이트](#) 를 방문하세요.

또는 x64 아키텍처가 실행 중인 Windows

환경에서 미리 빌드된 Node.js®를 다운로드하세요.

Windows 설치 프로그램 (.msi)

Standalone Binary (.zip)

1) 설치

설치 후, 터미널 창에 node라고 입력한 후, Enter를 누르면
설치한 버전이 표시되며, 노드가 실행되고 커서가 > 모양으로 바뀐다



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치 하세요! https://aka.ms/PSWindows

PS C:\Users\acorn> node
Welcome to Node.js v22.14.0.
Type ".help" for more information.
> |
```

← 설치한 버전이 표기됨

일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

커서가 > 모양으로 바뀌면, JavaScript를 사용할 수 있는 상태이므로, 다음과 같이 간단한 JavaScript 코드를 작성해 보세요.

```
PS C:\Users\acorn> node
Welcome to Node.js v22.14.0.
Type ".help" for more information.
> console.log(1+2)
3
undefined
> |
```

설치한 버전 표기

```
PS C:\Users\acorn> node
Welcome to Node.js v22.14.0.
Type ".help" for more information.
> console.log(1+2)
3
undefined
> +function hello(name){
... console.log(name+"님, 안녕하세요?");
... }
undefined
> hello
[Function: hello]
> hello("홍길동")
홍길동님, 안녕하세요?
undefined
> |
```

함수를 정의하고 호출하는 코드 예시



학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

JavaScript는 어떤 목적으로 개발되었는가?

- 1 서버 측 데이터베이스 관리
- 2 브라우저 내 동적 웹 페이지 구현
- 3 운영 체제 개발
- 4 모바일 애플리케이션 개발



학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

JavaScript는 어떤 목적으로 개발되었는가?

- 1 서버 측 데이터베이스 관리
- ☒ 2 브라우저 내 동적 웹 페이지 구현
- 3 운영 체제 개발
- 4 모바일 애플리케이션 개발

정답

2

해설

JavaScript는 웹 페이지를 더 인터랙티브하고 생동감 있게 만들기 위해 탄생되었습니다.



학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

JavaScript의 표준 명세는 무엇이라고 불리는가?

- 1 JavaScript Standard
- 2 ECMAScript
- 3 JS Core
- 4 WebScript



학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

JavaScript의 표준 명세는 무엇이라고 불리는가?

1 JavaScript Standard



2 ECMAScript

3 JS Core

4 WebScript

정답

2

해설

ECMAScript는 JavaScript의 공식 표준 이름이며, 다양한 버전이 나오면서 진화되었습니다.



학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

JavaScript의 표준화가 필요했던 주된 이유는 무엇인가?

- 1 다양한 브라우저 간의 호환성 문제
- 2 서버 성능 저하
- 3 모바일 기기 지원 부족
- 4 보안 취약점 증가

학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

JavaScript의 표준화가 필요했던 주된 이유는 무엇인가?



다양한 브라우저 간의 호환성 문제

2

서버 성능 저하

3

모바일 기기 지원 부족

4

보안 취약점 증가

정답

1

해설

예전에는 브라우저마다 JavaScript가 다르게 작동해서 웹 사이트 개발이 어려웠기 때문에 표준화가 필요해졌습니다.



학습 평가

Q1

Q2

Q3

Q4

Q5

Q4

JavaScript를 비동기 통신에 활용하게 해 준 기술은 무엇인가?

1 JSON

2 SOAP

3 AJAX

4 FTP





학습 평가

Q1 Q2 Q3 **Q4** Q5

Q4

JavaScript를 비동기 통신에 활용하게 해 준 기술은 무엇인가?

1 JSON

2 SOAP

☒ 3 AJAX

4 FTP

정답

3

해설

AJAX 덕분에 웹 페이지를 새로고침하지 않고 서버와 통신할 수 있게 되었습니다.



학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

JavaScript에서 객체 간 상속은
어떻게 이루어지는가?

- 1 클래스를 확장해서
- 2 인터페이스를 구현해서
- 3 다른 객체를 참조하는 방식으로 (프로토타입 체인)
- 4 상속 불가



학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

JavaScript에서 객체 간 상속은 어떻게 이루어지는가?

1 클래스를 확장해서

2 인터페이스를 구현해서



다른 객체를 참조하는 방식으로 (프로토타입 체인)

4 상속 불가

정답

3

해설

객체는 다른 객체를 프로토타입으로 연결해 기능을 공유하며, 이를 프로토타입 체인이라고 합니다.

JavaScript의 특징

- 웹 브라우저에서 동작하는 유일한 프로그래밍 언어
- 개발자가 별도의 컴파일 작업을 수행하지 않는 인터프리터 언어
 - 대부분의 모던 JavaScript 엔진은 인터프리터와 컴파일러의 장점을 결합해 비교적 처리 속도가 느린 인터프리터의 단점 해결
- 명령형, 함수형, 프로토타입 기반 객체지향 프로그래밍을 지원하는 멀티 패러다임 프로그래밍 언어

학습정리

2/2

JavaScript의 특징

- 클래스 기반이 아닌 프로토타입 기반의 객체지향 언어

참고 [프로토타입이란?](#)

객체 간의 상속이 아닌 다른 객체를 참조하는 방식

02주. 데이터 준비와 연산 명령어로 기본 프로그램 만들기

01

변수 선언과 메모리 구조

