



실증 아카데미 리액트 특강


JavaScript와 웹 브라우저의 이해



일시 2021년 01월 10일

주최 동아대학교





CONTENTS—

01

JavaScript란?

스크립트 언어
스크립트 언어와 컴파일 언어
JavaScript

02

JavaScript의 특징

프로토타입 기반, 동적 타이핑,
일급 객체, 클로저 등

03

JavaScript 사용처

04

웹 브라우저란?

05

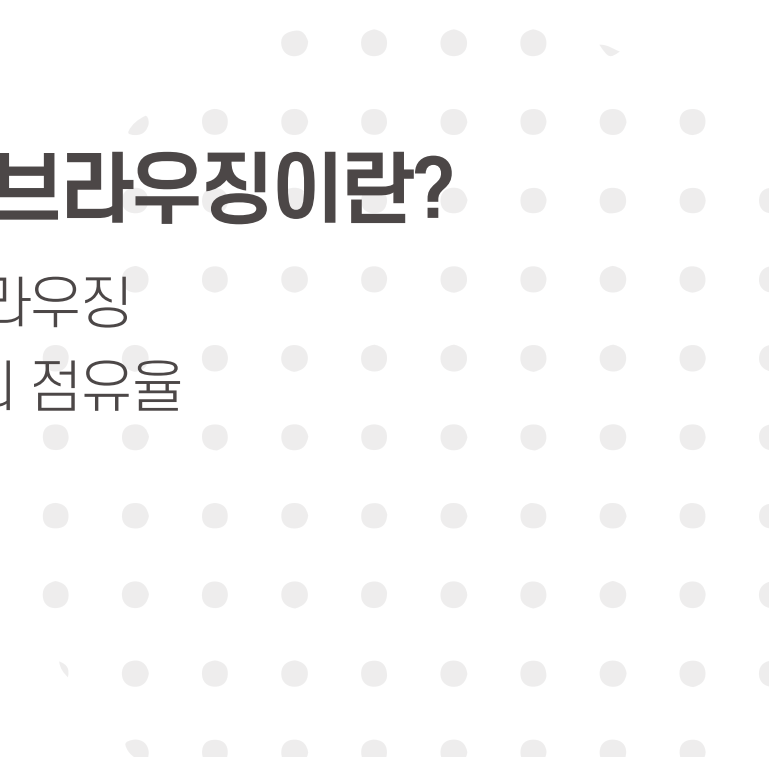
웹 브라우저의 동작 방식

브라우저가 하는 일
웹 브라우저

06

크로스 브라우징이란?

크로스 브라우징
브라우저의 점유율





01

JavaScript란?

스크립트 언어

스크립트 언어와 컴파일 언어

JavaScript

스크립트 언어

기존에 존재하는 소프트웨어(애플리케이션)를 제어하기 위한 용도로 쓰이는 언어

- 스크립트 언어는 대체로 인터프리터 방식을 사용함 (인터프리터 언어에 속함)
- 컴파일 과정을 거치지 않고 실시간으로 텍스트를 분석하며 실행
- 코드에 문법 오류가 있으면 실행 시점에 발견
- 단일 응용 프로그램으로 사용되기보다는 응용 프로그램 내에서 특정 기능을 수행하는 역할
 - ※ 요즘은 고도로 스크립트 언어가 고도로 발전되어 언어의 경계를 정의하는 것이 무의미해져 감
- JavaScript, JSP, Perl, PHP, Python, Lua, Ruby 등이 있음



스크립트 언어와 컴파일 언어

컴파일 언어

- 소스 코드 전체를 기계어(Low-level 언어)로 번역한 후에 실행(단 한 번)
- 실행 파일을 생성
- 컴파일 단계와 실행 단계가 분리되어 있음
- 수정할 때마다 컴파일해야 하므로 코드 수정이 불편함
- 구동되는 소스 코드를 읽을 수 없음
- 각 운영체제에 따라 컴파일 목표를 다르게 하므로 운영체제마다 컴파일하여 실행

스크립트 언어

- 코드를 한 줄씩 번역하며 바이트 코드로 변환한 후에 실행(번역 하자마자, 반복 수행)
- 실행 파일을 생성하지 않음
- 인터프리트 단계와 실행 단계가 분리되어 있지 않음
- 컴파일 과정이 없어 코드 수정이 간단함
- 구동되는 소스 코드를 쉽게 읽을 수 있음
- 스크립트 엔진이 있으면 모든 운영체제에서 실행할 수 있음(JavaScript는 웹 브라우저가 엔진)

JavaScript

프로토타입 기반의 멀티 패러다임 프로그래밍 언어

- 명령형, 함수형, 프로토타입을 지원함
- 스크립트 언어에 해당
- HTML, CSS와 함께 웹을 구성하는 3요소(동작을 담당)

MDN 문서: <https://developer.mozilla.org/ko/>

- **초창기에는 매우 불완전한 언어**

각각 브라우저마다 각기 다른 Script를 제공하여 같은 소스 코드가 다른 브라우저에서는 동작하지 않는 크로스 브라우징 이슈가 빈번하게 발생

- **ECMAScript**

JavaScript의 표준안 제시 사양으로, 각 브라우저 제조사는 최대한 ECMAScript 사양을 준수하여 JavaScript 엔진을 구현(IE는 대체로 지원하지 않음)

- **SPA 서비스의 발전**

기존에는 페이지 이동으로 웹 서비스를 구현했으나, 너무 느린 속도로 인해 JavaScript를 사용하여 한 페이지 내에서 여러 서비스를 지원해주는 Single Page Application이 대중화, Front-End가 발전

02

JavaScript의 특징

프로토타입 기반, 동적 타이핑,
일급 객체, 클로저 등

JavaScript의 특징

01

비동기 I/O를 지원하며, 단일 스레드 이벤트 루프 기반으로 동작함으로써 요청 처리 성능이 좋음

03

동적 타이핑 언어로 모든 타입이 객체(Object)로 이루어져 있음(함수 또한 객체)

02

지속하여 발전된 언어기 때문에 과거의 문법이 많이 남아 있고, 여러가지 문법이 마구 섞여 있어 소스 코드를 혼란스럽게 함

04

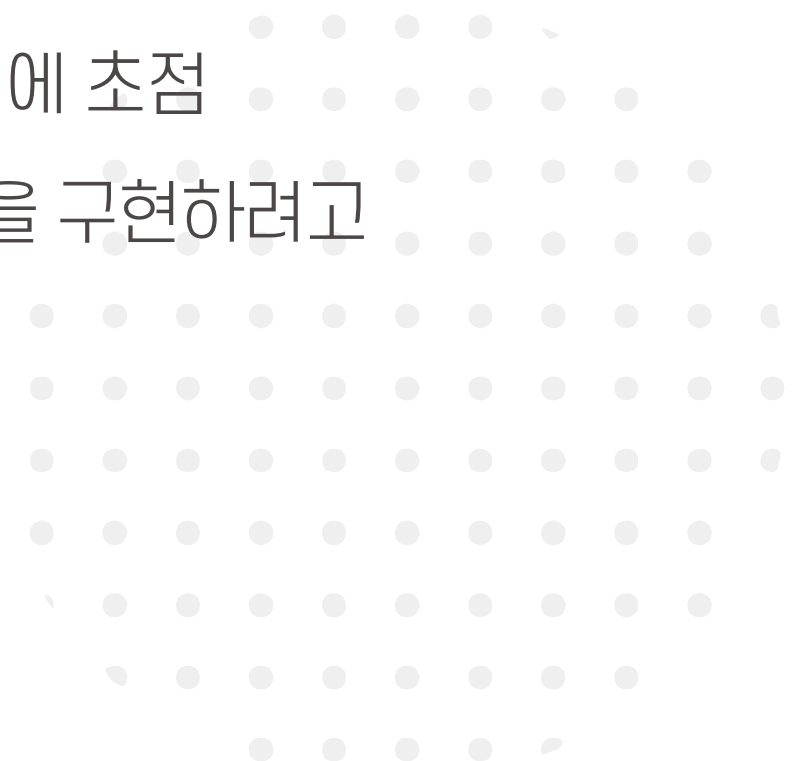
고차 함수, 호이스팅, 클로저 등의 함수형 프로그래밍 개념이 잘 구현되어 있으며, 확장성이 매우 높은 언어



프로토타입 기반 언어

클래스와는 다른 방식으로 객체지향을 구분하는 방법

- 이론상 부모, 자식의 구분이 없으나 JavaScript에서는 class 라는 문법적 설탕으로 구현됨(실제로는 아님)
- 클래스 - 인스턴스의 관계가 없어 기존의 객체를 복사하여 새로운 객체를 생성
- JavaScript의 모든 타입은 객체이며, 이 객체는 모두 Object라는 프로토타입으로 연결됨
- 프로토타입 체인이라는 것을 통해 상속을 구현함
- 클래스 기반 언어는 메소드의 모임으로 동작, 프로토타입 기반은 여러 가지 표본의 동작 방식에 초점
- 현재는 프로그램의 정확성, 안전성, 예측성, 효율을 떨어뜨리며, JavaScript 또한 class 기반을 구현하려고 하는 등 비평을 하는 추세



동적 타이핑 언어

C, Java와 같이 타입을 명시하지 않고 변수를 선언

- 선언이 아닌 할당에 의해 타입이 결정되며, 재할당에 의해 타입이 언제든지 변할 수 있음
- 이는 스크립트 언어의 특징
- 유연성은 높으나 신뢰성은 떨어지는 코드로, 가독성이 나쁨
- 상수(const)를 사용하여 최대한 값의 변경을 억제해야 함
- 전역 변수는 최대한 사용하지 않아야 하고, 동적 타이핑인 만큼 변수의 이름이나 범위(스코프)를 더더욱 신경써서 작성해야 함



일급 객체

JavaScript의 함수는 객체이며, 함수를 변수 명에 할당할 수 있음

- 함수에 함수를 매개변수로 넘길 수 있음
- 실행하는 시점에 함수 생성이 가능함
- 변수나 자료구조(객체, 배열)에 저장할 수 있음
- 함수의 반환(리턴) 값으로 함수를 반환할 수 있음



클로저

클로저는 함수 내부의 변수가 함수가 끝난 후에도 남아 있어 사용 가능한 매커니즘

- JavaScript는 스코프(변수의 생명 주기, 활동 기간)이 복잡한 언어
- 함수 호출 시점이 아닌 함수 생성 위치 중요함
- 외부 함수보다 중첩 함수가 더 오래 유지되는 경우, 중첩 함수는 이미 생명주기가 종료된

외부 함수의 변수를 참조할 수 있는 현상

- 상태를 안전하게 변경하고 유지하기 위해서 사용(캡슐화, 정보 은닉)
- 난해한 만큼 실수를 유발하기 쉬움

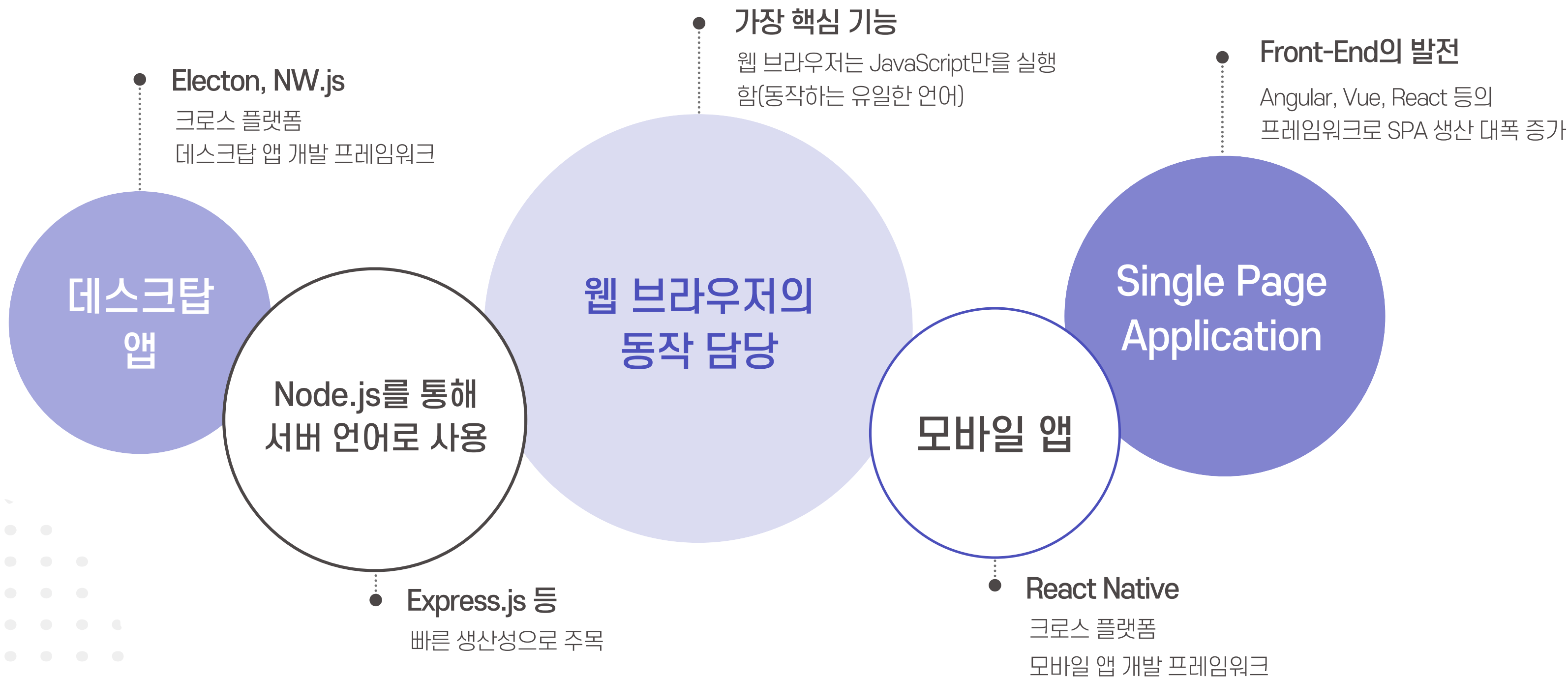




03

JavaScript 사용처

JavaScript 사용처



04

웹 브라우저란?

웹 브라우저



- HTML 문서 등 월드 와이드 웹을 기반으로 한 인터넷 콘텐츠 검색 및 열람 응용 프로그램
- HTML5 등의 발전으로 점차 기능이 확대되어 비대해지고 상황
- 대표적인 HTTP 사용자 에이전트
- 브라우저마다 엔진(JavaScript, 렌더)이 다름
- Edge, IE, Chrome, Firefox, Safari, Opera, 네이버 웨일, 삼성 인터넷 등

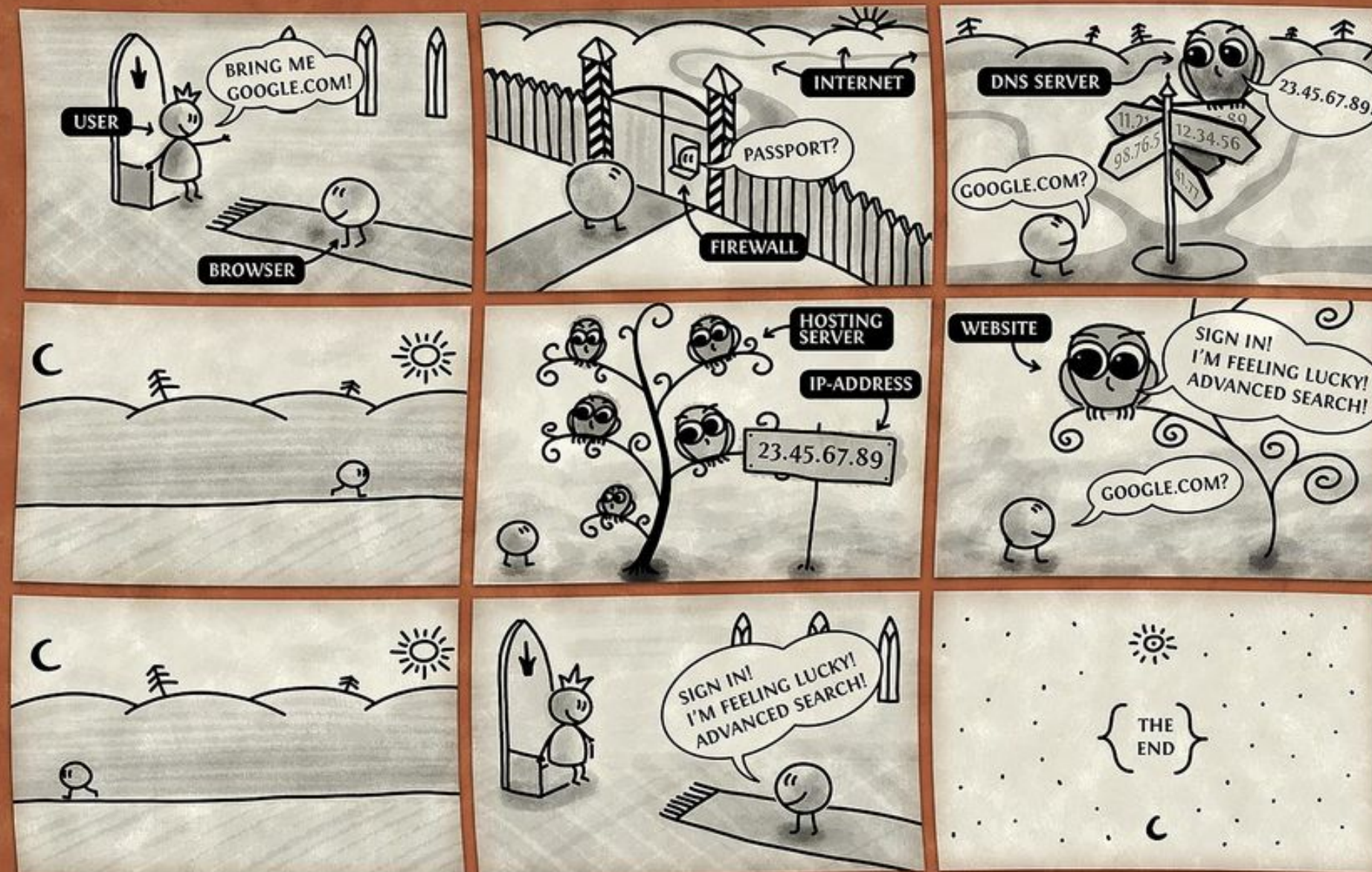
05

웹 브라우저의 동작 방식

브라우저가 하는 일
웹 브라우저

브라우저가 하는 일

웹 브라우저의 동작 방식 05



웹 브라우저

브라우저는 입력된 주소에 데이터를 요청하여 HTML 문서를 렌더링하여 보여줌

- 주소 창에 주소를 입력하면 해당 주소에 GET 방식으로 데이터를 요청함
- 응답 받은 데이터가 HTML 문서가 아니면(파일이면 다운로드 등 다른 행위) 해당 행위를 브라우저가 인지하여 실행
- HTML 문서일 경우, 문서 텍스트 파일을 해석하여 뼈대를 구성함
- 해석 도중 CSS 파일이 있다면 수집하여 외관을 입힘
- 해석 도중 JavaScript 파일이 있다면 해석을 멈추고 소스 코드를 실행함
- GUI 응용 프로그램이므로 최대한 표준에 따라 모습을 구성하지만, 브라우저에 따라 차이가 있을 수 있음



06

크로스 브라우징이란?

크로스 브라우징
브라우저의 점유율

크로스 브라우징

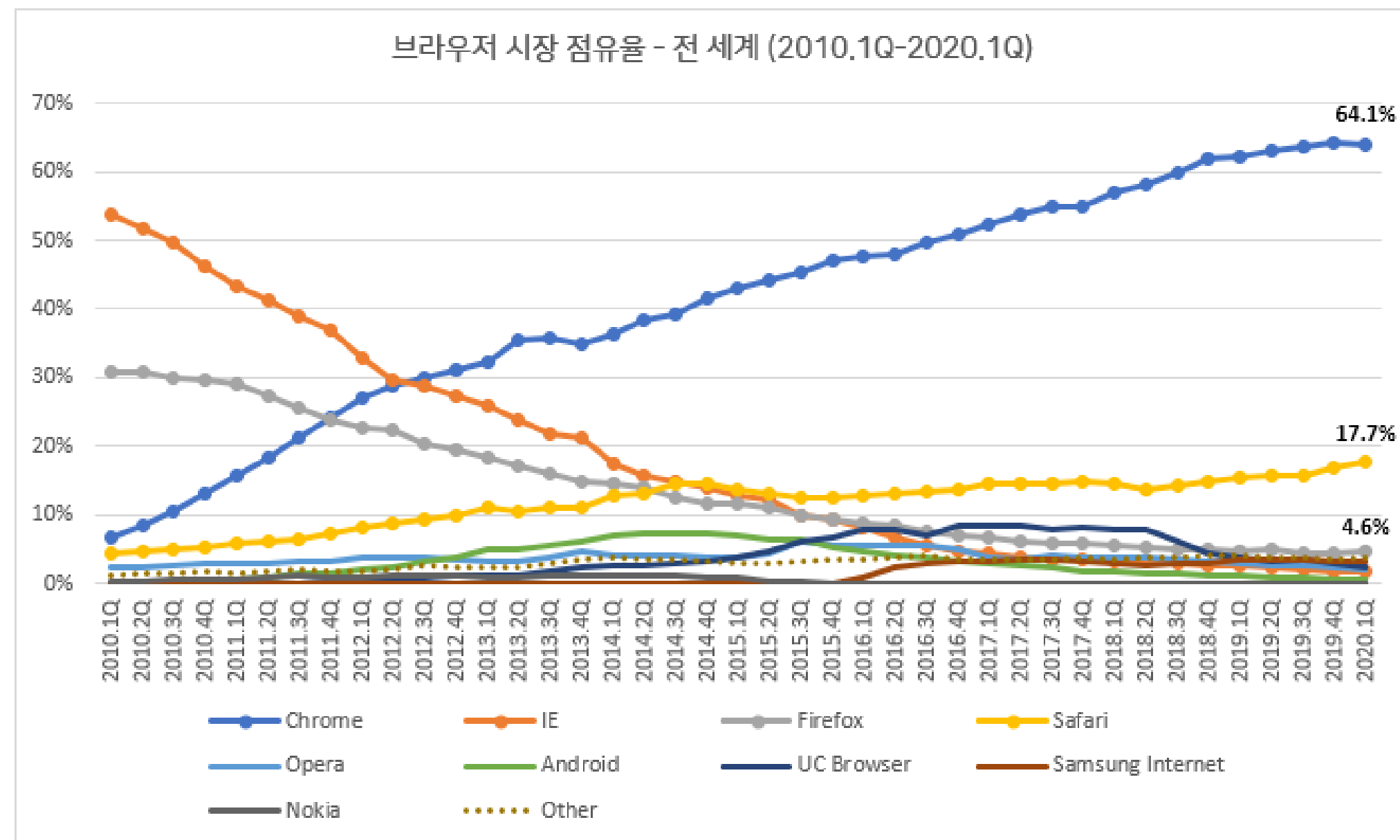
브라우저의 엔진이 다름을 고려하여 모든 브라우저에서 동일하게 동작할 수 있도록 작성하는 기법

- 브라우저마다 엔진이 달라 HTML5, CSS, JavaScript가 작동되지 않는 경우가 생길 수 있음
- 이는 특정 브라우저에서는 JavaScript 코드가 에러가 나거나, CSS의 스타일이 먹히지 않아 깨져 보이게 나올 수 있음을 의미함
- W3C 웹 규격이 제안되었지만, IE 등 구형 브라우저에서는 완벽하게 규격을 따르지 않음
- 때문에 모든 브라우저에서 올바르게 동작하는 호환성 작업 기법이 사용되고 있음
- 완벽하게 동일하게 구현할 수는 없으므로 동일성이 아닌 동등성을 추구
- 한 가지 브라우저가 독점하지 않는 이상 모든 브라우저를 고려해야 함
- Babel 등의 크로스 브라우징 도구가 존재함
- 크로스 브라우징과 비슷하게 응용 프로그램에서는 크로스 플랫폼이라는 이슈가 있음



브라우저의 점유율

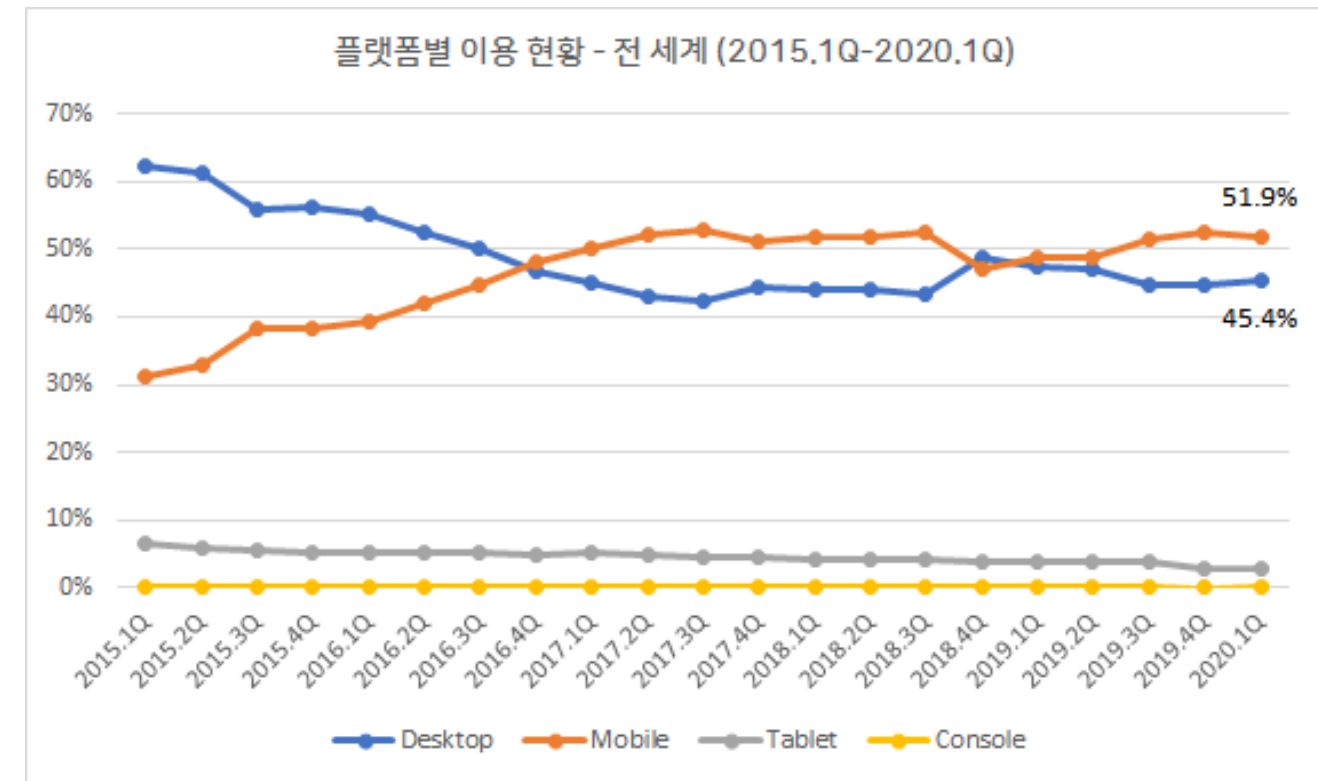
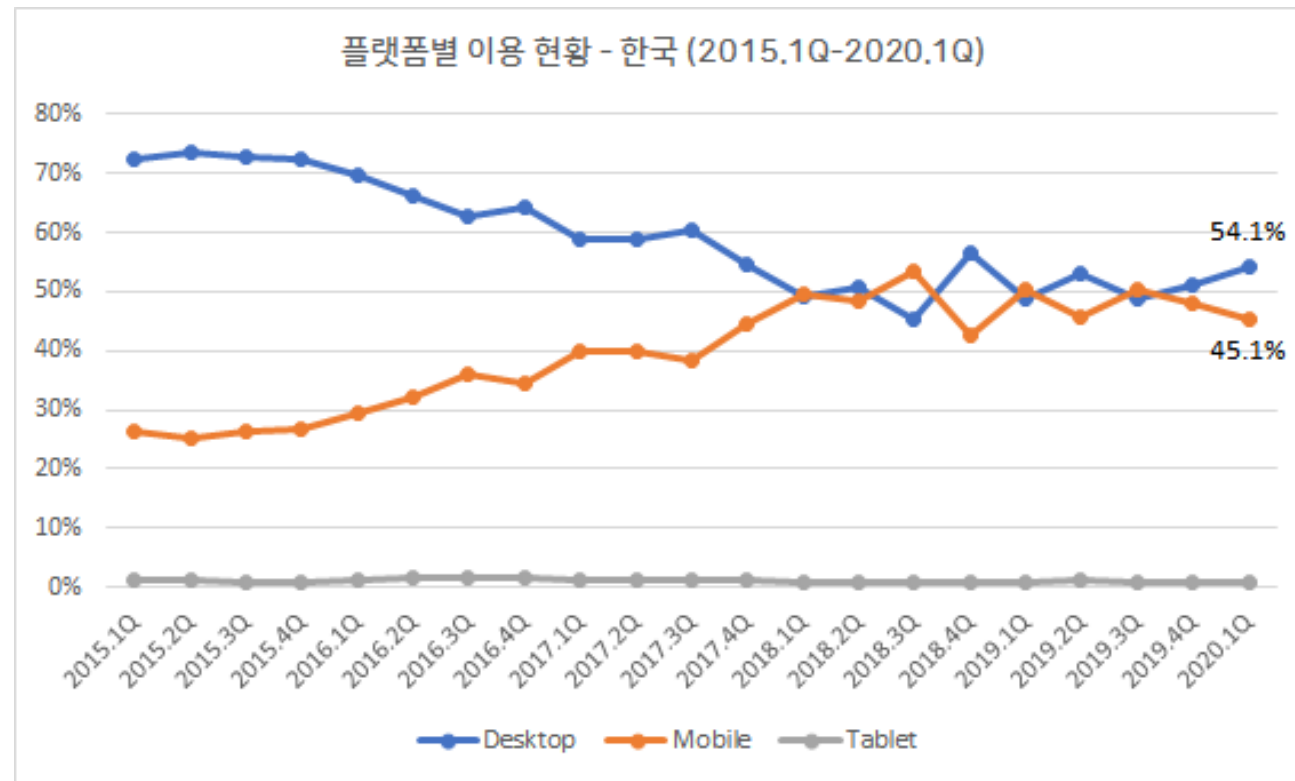
전 세계 브라우저 시장 점유율



참조: <https://presscat.co.kr/blog/browser-market-share-mar2020/>

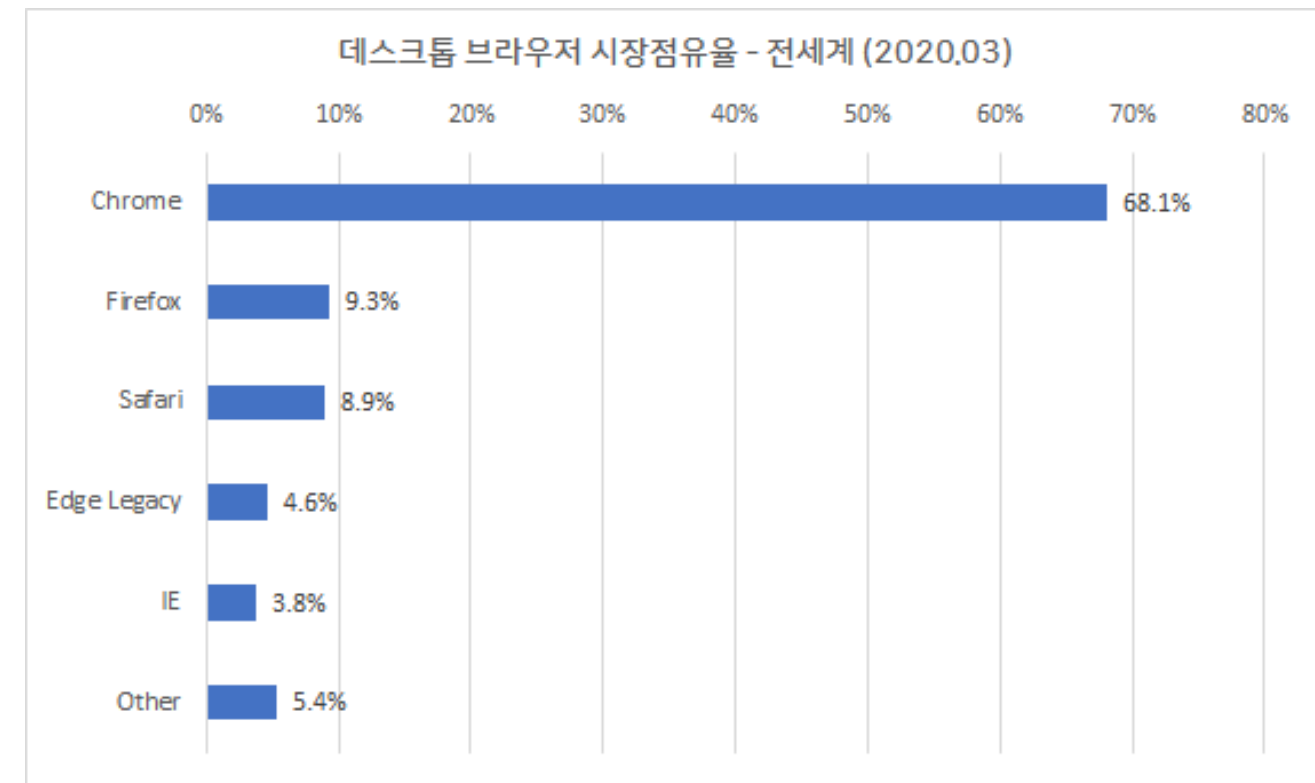
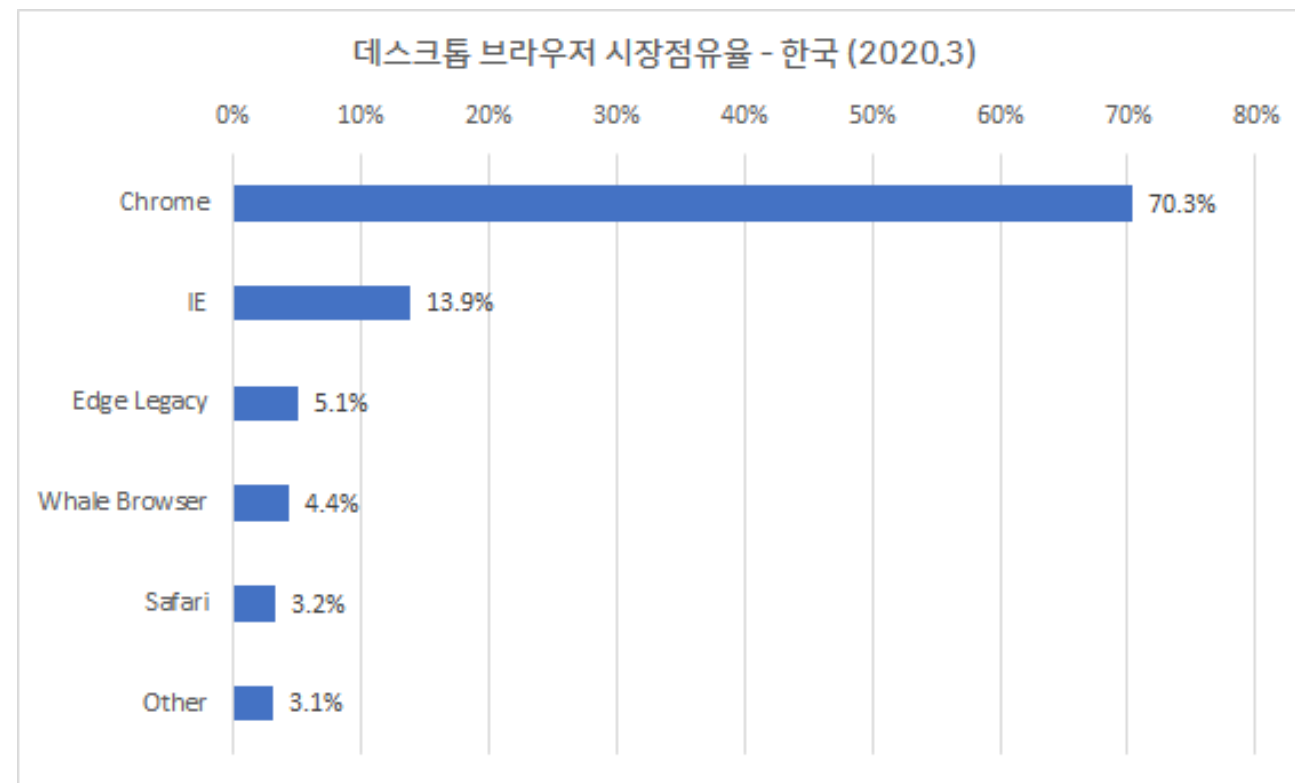
브라우저의 점유율

데스크탑 vs 모바일



브라우저의 점유율

데스크탑 브라우저 점유율



실증 아카데미 리액트 특강

THANK
YOU

(주) 인바이즈 박철현
(주) 인바이즈 개발팀