

**Web
Assembly 란?**

WA

주제를 선정하게 된 이유

SW개발

Front-end

웹 기술 언어(HTML, CSS, JavaScript, WebAssembly)를 활용하여
사용자와 서비스 사이의 다리를 놓습니다. 사용자와 시스템의 상호 작용을 돕기 위해
웹 브라우저 등 다양한 플랫폼과의 웹 호환이 가능한 어플리케이션을 개발합니다.

평소에 관심있던 기술

+

2024 팀네이버 신입 공채 직무소개에
WebAssembly가 들어있어서

WebAssembly란?

웹 브라우저에서 C, C++, RUST와 같은
처리 속도가 빠른 low-level 언어의 코드를
사용할 수 있도록 지원하는 도구

이미 컴파일 된 상태로 사용하기 때문에
브라우저 엔진이 Javascript를 실행하기 위해 진행하는
파싱, 컴파일, 최적화 등의 과정들을 거의 거치지 않는다

WebAssembly란?

```
#include <emscripten.h>
```

```
int Increment(int value) {  
    return (value + 1);  
}
```

```
emcc side_module.c -s SIDE_MODULE=2 -O1 -s EXPORTED_FUNCTIONS=['_Increment'] -o side_module.wasm
```

```
WebAssembly.instantiateStreaming(fetch("side_module.wasm"), importObject).then((result) => {  
    const value = result.instance.exports.Increment(17);  
    console.log(value.toString());  
})
```

C언어로 작성한 Increment 함수를
WebAssembly 모듈로 추출하고
사용하는 과정

Speed comparison of various programming languages

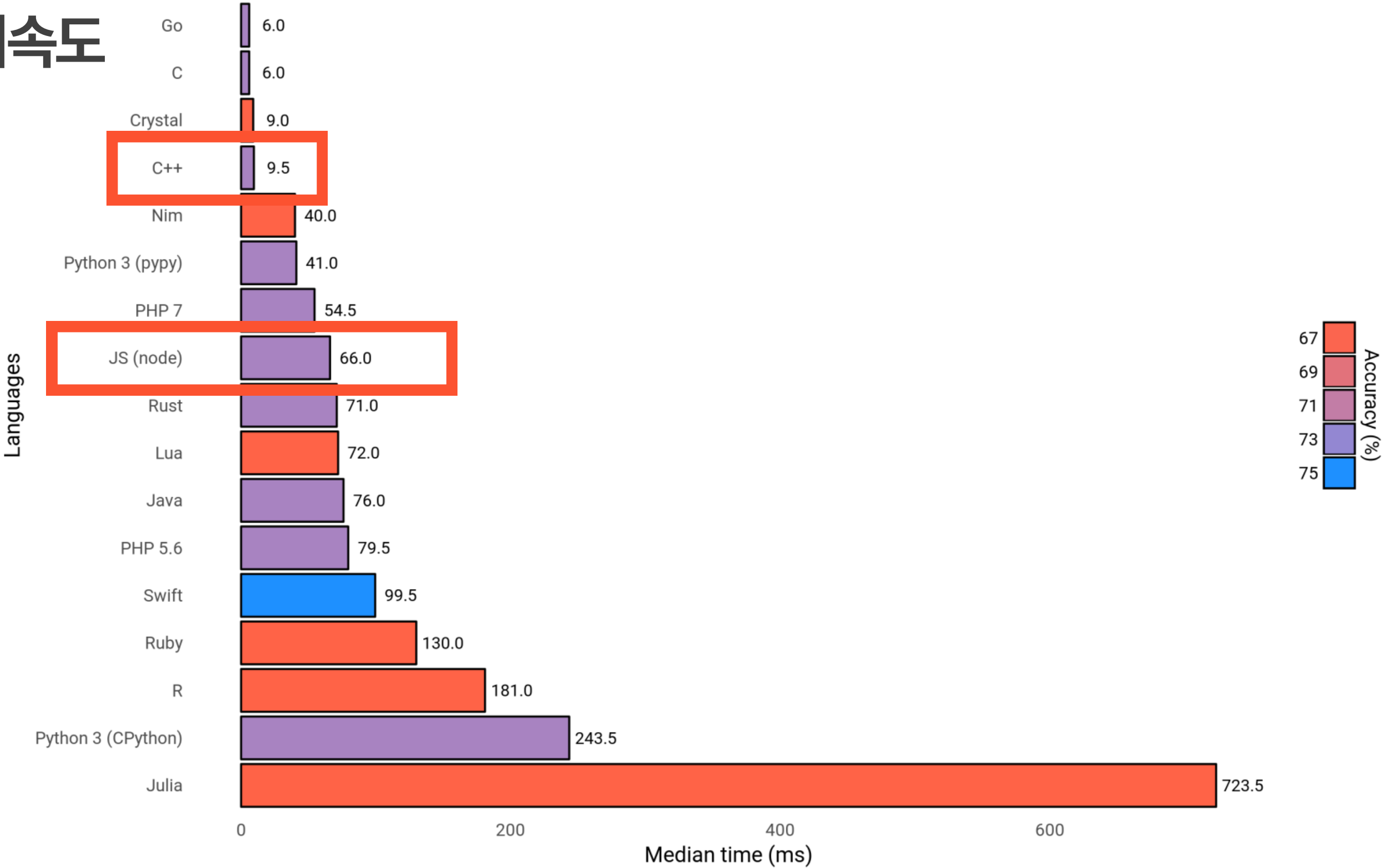
Method: calculating π through the Leibniz formula x times

Javascript의 성능 - 처리속도

C : 6ms

C++ : 9.5ms

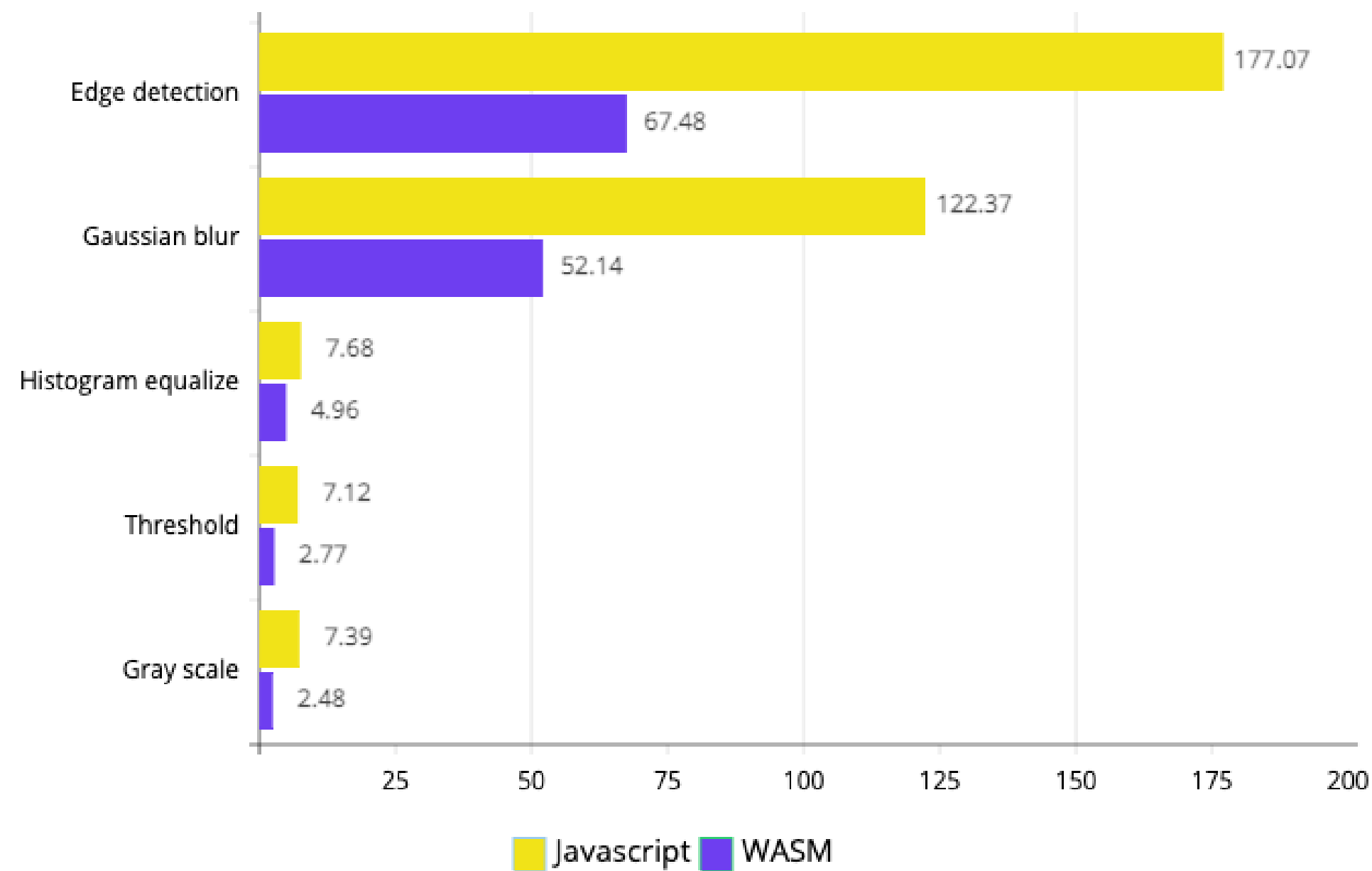
JS(node) : 66ms



<https://github.com/niklas-heer/speed-comparison>

라이프니츠 원주율 공식으로 원주율의 근사값을 각 언어별로 구했을 때 결과

Javascript의 성능 - 처리속도



canvas에서 이미지 필터링
과정을 기준으로 성능 측정한 결과

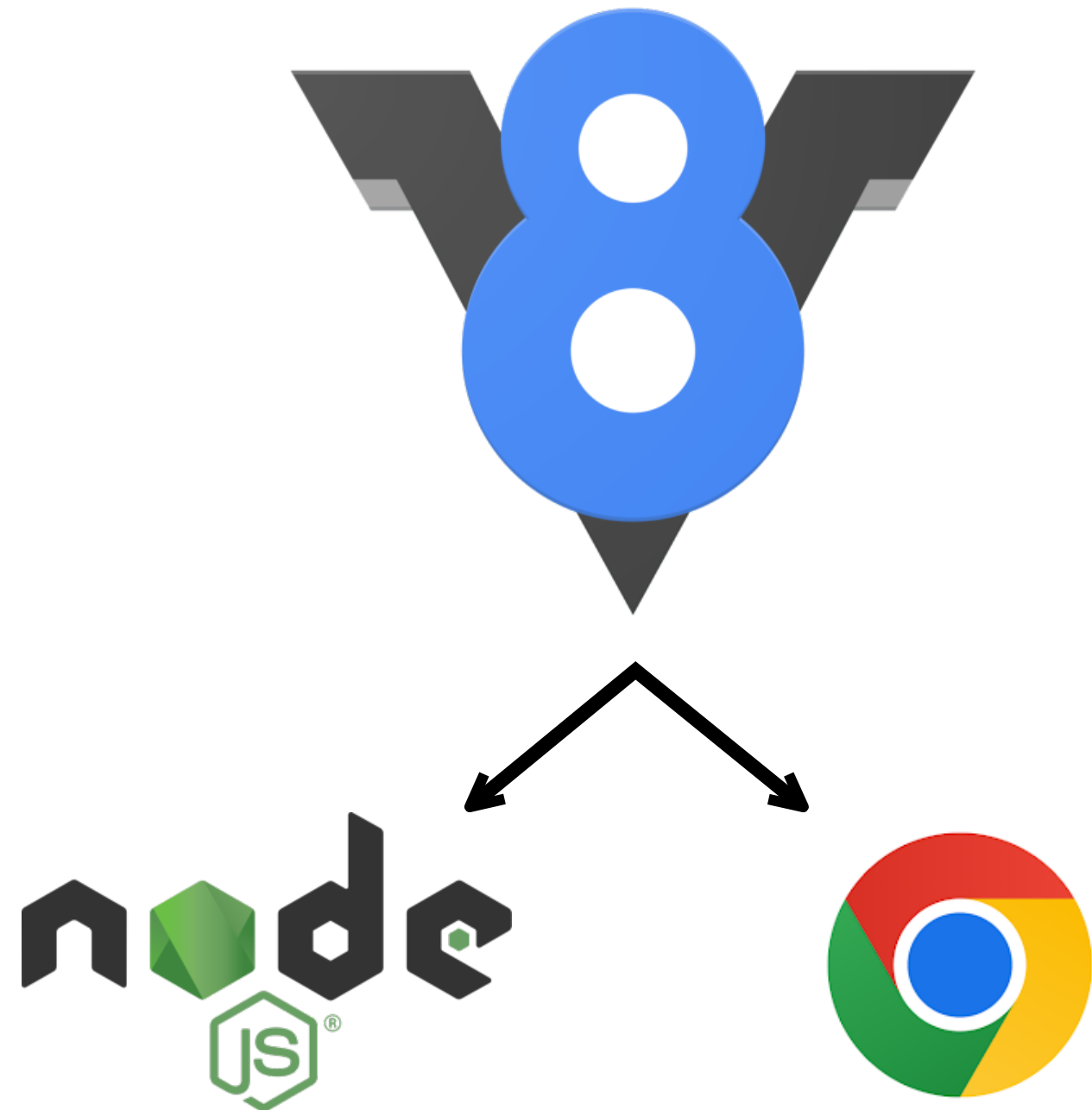
Javascript의 성능 - 로딩시간

Javascript는 인터프리터 언어?

Javascript의 성능 - 로딩시간

Javascript는 인터프리터 언어가 맞지만
Javascript를 실행하는 환경에 따라 다르다

Javascript의 성능 - 로딩시간



Javascript는 프로그래밍 언어
Javascript를 실행시키는
Javascript 엔진에 따라 동작이 달라진다

Node.js와 구글 크롬은
V8이라는 Javascript 엔진을 사용한다

Javascript의 성능 - 로딩시간



V8엔진은 최적화를 위해서
Javascript를 인터프리터 언어처럼 사용하지 않는다

Javascript의 성능 - 로딩시간

JIT(just-in-time) 컴파일 = 인터프리터 + 정적컴파일

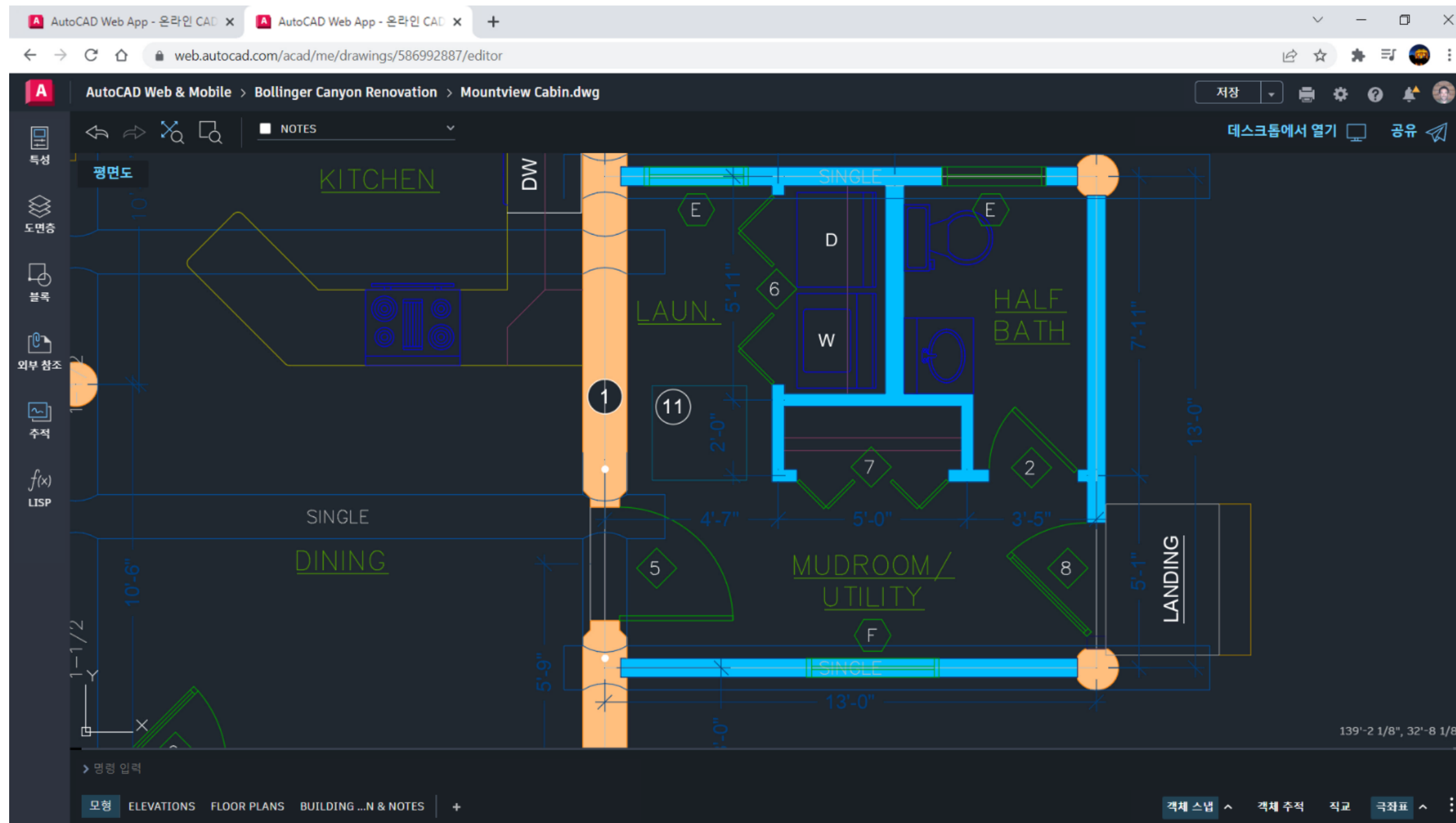
코드를 실행하는 동안에 실시간으로
전체 코드 중 필요한 부분만을
기계어로 변환하는 방식

변환된 코드는 캐시에 저장되기 때문에
코드를 재사용하면 다시 컴파일 할 필요가 없다

WebAssembly가 더 빠른 이유

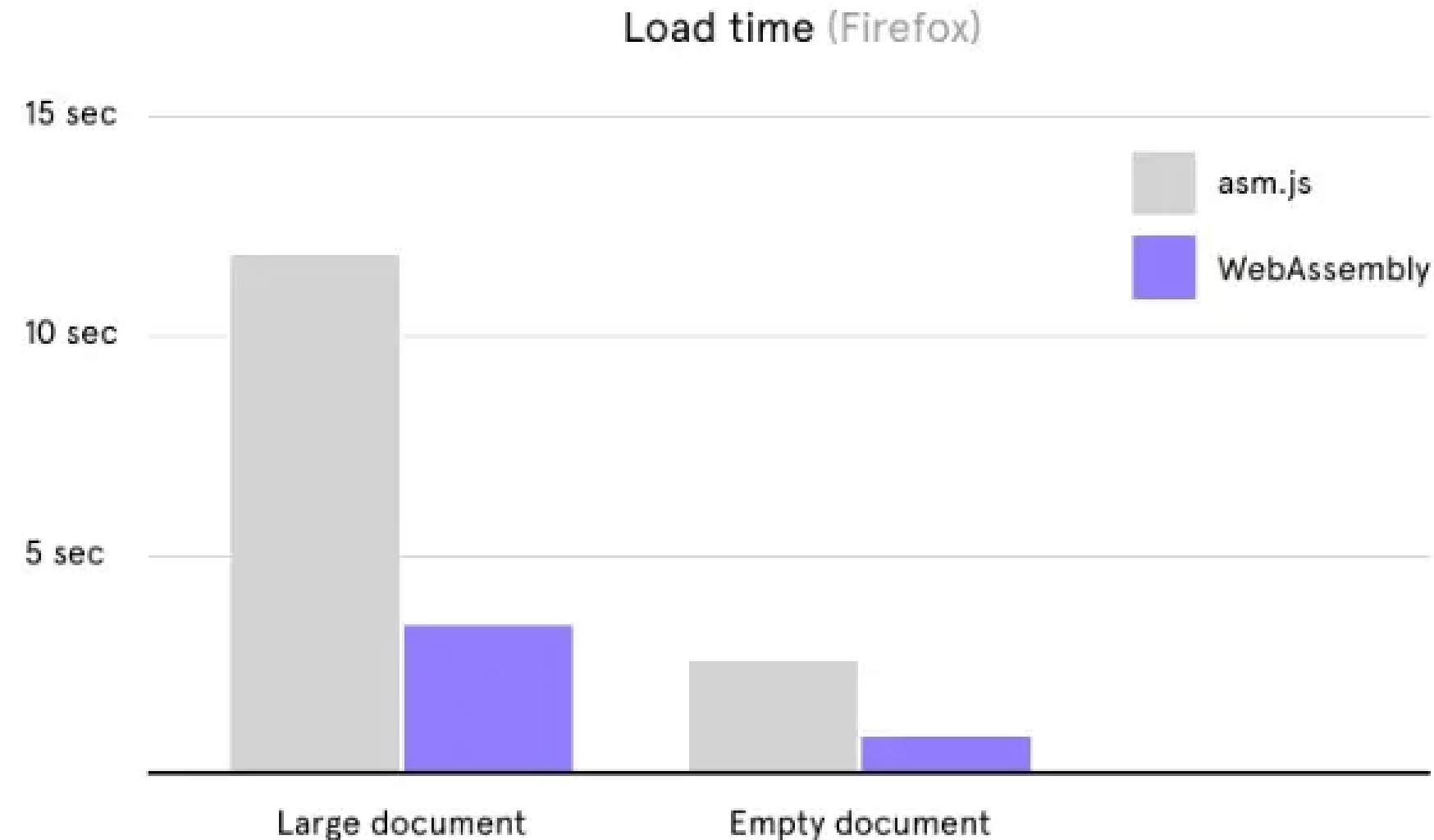
이미 컴파일 된 상태로 사용하기 때문에
브라우저 엔진이 Javascript를 실행하기 위해 진행하는
파싱, 컴파일, 최적화 등의 과정들을 거의 거치지 않는다

WebAssembly 도입 사례 - AutoCAD Web App



AutoCAD Web App과 같이 웹 편집 툴의 경우
WebAssembly를 사용하고 있다.

WebAssembly 도입 사례 - Figma



Figma는 WebAssembly이전에 asm.js를 사용했다

WebAssembly를 도입하고 문서를 로드하는 시간이 3배 정도 빨라졌다

참고자료

<https://d2.naver.com/helloworld/8257914>

<https://developer.mozilla.org/ko/docs/WebAssembly/Concepts>

<https://medium.com/watcha/wasm-%EA%B0%84%EB%8B%A8-%EC%82%AC%EC%9A%A9%EA%B8%B0-d2e14c760a0d>