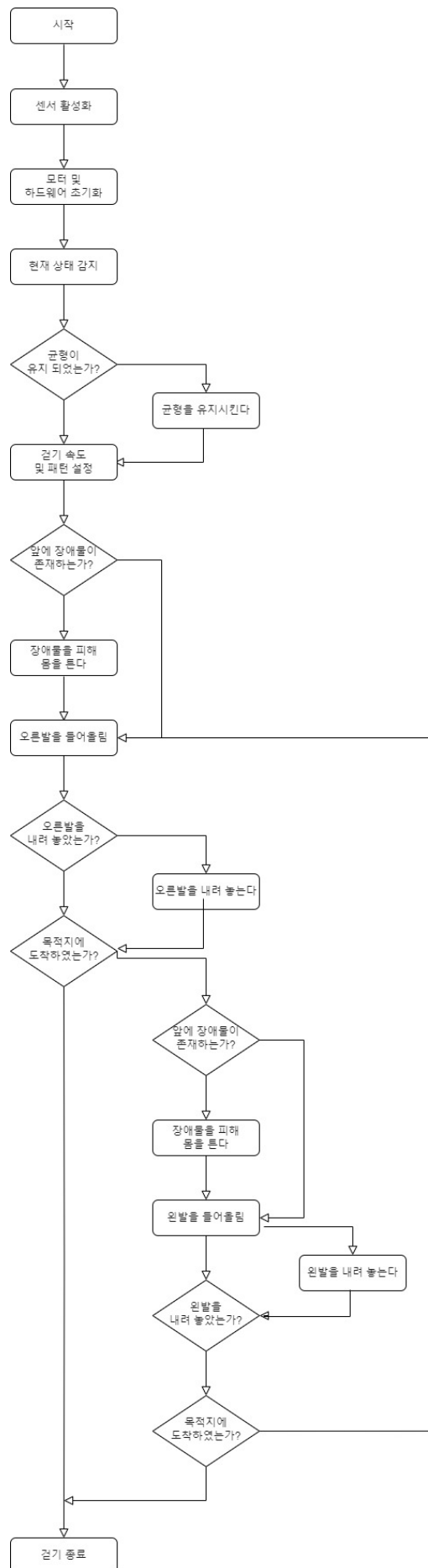


1주차 1차시

1. 2족 보행 로봇이 있습니다. 해당 로봇에 대한 걷다라는 기능을 디자인 해보세요.



2.1 Ajax에 대해서 알아보시오

- Asynchronous JavaScript and XML(비동기식 자바스크립트와 xml)의 약자로 자바스크립트의 라이브러리중 하나이다.
- 서버와 비동기적으로 데이터를 주고받는 자바스크립트 기술을 의미한다.
- 새로고침 하지 않고 단순히 웹에서 무언가를 부르거나 어떠한 데이터를 조회 하고 싶은 경우에 주로 사용한다.
- 웹 페이지의 필요한 일부분만을 갱신해 시간과 자원을 아낄 수 있다.
- Ajax 요청을 만들고 서버와 통신하기 위해 주로 XMLHttpRequest 객체가 사용되는데 이 객체를 사용하여 서버로 데이터를 보내고 서버로부터 데이터를 받아올 수 있다.
- 초기에 Ajax는 XML을 주로 사용하여 데이터를 교환하는 데에 사용되었으나, 현재는 JSON, HTML, 텍스트 등 다양한 데이터 형식을 처리할 수 있다.

2.2 jQuery에 대해서 알아보시오

- 자바스크립트 프로그래밍을 보다 쉽고 효율적으로 수행하기 위한 라이브러리다. (명령어 간소화)
- DOM 조작, 이벤트 처리, Ajax 요청, 애니메이션 효과 등에 사용된다.
- 코드가 간결하고 가독성이 좋아 유지보수에 유리하다.
- 최근에는 다른 라이브러리, 프레임워크를 사용한다.

2.3 V8 엔진에 대해서 알아보시오

- Google에서 개발한 오픈 소스 자바스크립트 엔진으로 C++로 개발하였고 자바 스크립트 전용의 웹 어셈블리 엔진이다.
- 인터프리터를 사용하는 대신 JIT(Just In Time) 컴파일러를 구현함으로써 코드 실행 시에 자바스크립트 코드를 바이트 코드로 컴파일한다.
- Node JS 런타임 및 Chrome 브라우저에서 사용되고 있다.

2.4 Nodejs에 대해서 알아보시오

- Chrome V8 JavaScript 엔진으로 빌드 된 JavaScript 오픈 소스 런타임 환경이다.
- 다양한 자바스크립트 애플리케이션을 실행할 수 있고 서버에서 많이 사용된다.
- 내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도 소프트웨어 없이 동작하는 것이 가능하다.
- 웹 서버의 동작에 있어 더 많은 통제에서 벗어나 여러 가지 기능을 가능하게 한다.
- 과거에는 Server-Client 웹사이트를 만들 때 웹에서 표시되는 부분은 JavaScript를 사용하여 만들어야만 했으며, 서버는 Reby, Java 등 다른 언어를 써서 만들었어야 했는데 한 가지 언어로 전체 웹 페이지를 만들 수 있다.

2.5 SPA 프레임워크의 뜻을 알아보시오

- 과거에는 어떠한 페이지에서 버튼을 클릭했을 때 해당 페이지의 일부 영역만 변경되어도 완전히 새로운 페이지를 재렌더링하여 전송해주는 방식이었지만 이것이 버거워지게 되면서 이러한 문제를 해결하기 위해 등장한 기술이다.

- Single Page Application의 약자로 단일 페이지 어플리케이션이라는 의미다.
- 어떠한 웹 사이트의 전체 페이지를 하나의 페이지에 담아 동적으로 화면을 변경해 표시하는 기술이다.
- 최초에 한 번 서버에 요청하여 페이지 전체를 로드하고, 이후부터는 특정 변경 사항이 일어나는 부분만 AJAX와 같은 기술을 통해 데이터를 바인딩하는 방식으로 동작한다.
- DOM 조작이 빈번하게 일어나 브라우저의 성능을 저하하는 문제가 있어 Virtual DOM이라는 개념이 생겼다
- V가상의 DOM 트리로 HTML 정보를 저장하고 있다가, 이 트리에서 변경이 발생하면 모든 변화를 모아 한 번 브라우저를 호출해 갱신하는 방법이다.
- 사용자가에게 빠른 인터랙션을 제공해야하는 서비스는 SPA 프레임워크로 개발하는 것이 좋고, 간단한 정보를 제공하는 서비스라면 기존의 웹사이트 개발 방식으로 개발하는 것이 좋다.
- SPA 프레임워크에는 Angular, React, Vue 등이 있다.

3. V8 엔진에 대해서 정리해보시오

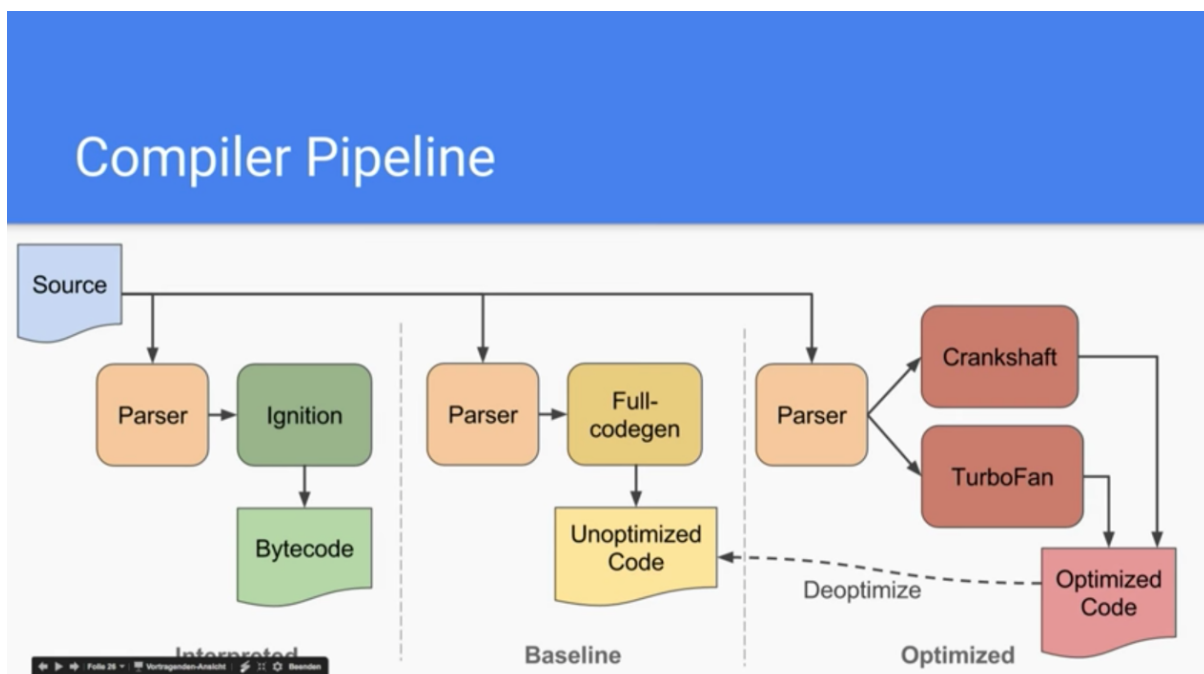
3.1 Ignition

- 자바스크립트 코드를 바이트 코드로 변환하는 인터프리터다.
- 원본 소스 코드보다 컴퓨터가 해석하기 쉬운 바이트 코드로 변환하여, 수시로 코드를 파싱하는 작업을 최소화하고 코드의 양도 줄임으로 메모리 공간도 효율적으로 관리할 수 있다.

3.2 TurboFan

- 바이트 코드를 최적화된 코드로 컴파일하는 컴파일러다.
- 바이트 코드로 수시로 변환하는 과정을 최소화하기 위해 사용된다.
- V8은 런타임 중에 Profiler에게 함수나 변수들의 호출 빈도와 같은 데이터를 모으고 자기 기준에 맞는 코드를 가져와 최적화 시킨다.
- 대표적인 기법으로 Hidden Class와 Inline Caching 기법이 있다.
- Hidden Class는 객체의 프로퍼티가 저장된 주소에 대한 상대적인 위치 정보를 기억하고 있다가 프로퍼티가 동적으로 생성되거나 변경될 때, 함께 업데이트되는 숨겨진 클래스다.
- Inline Caching은 반복문 내의 객체 접근 시 '조회' 작업을 생략하게 함으로 성능 향상을 도모하는 기법이다.

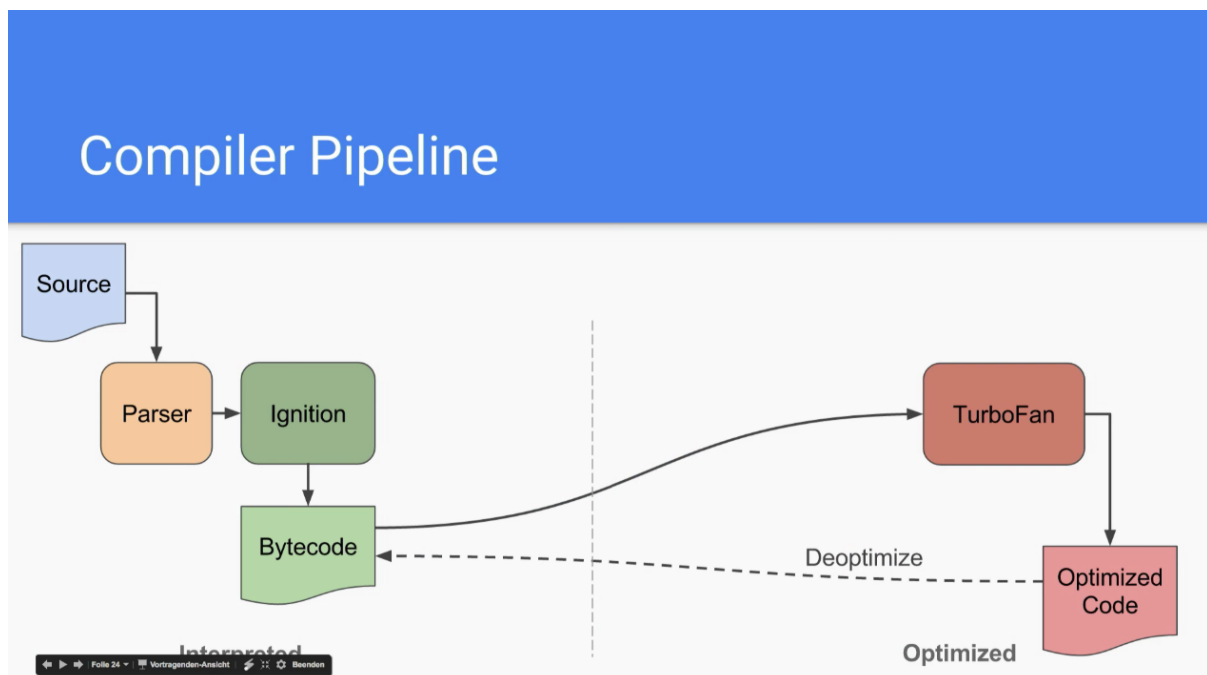
3.3 5.9 이하 V8 엔진 구조



- Full-codegen과 crankshaft라는 컴파일러가 존재했다.
- Full-codegen은 Ignition과 같이 자바스크립트 코드를 바이트 코드로 변환하여 실행 속도를 빠르게 하기 위해 사용했던 컴파일러다.

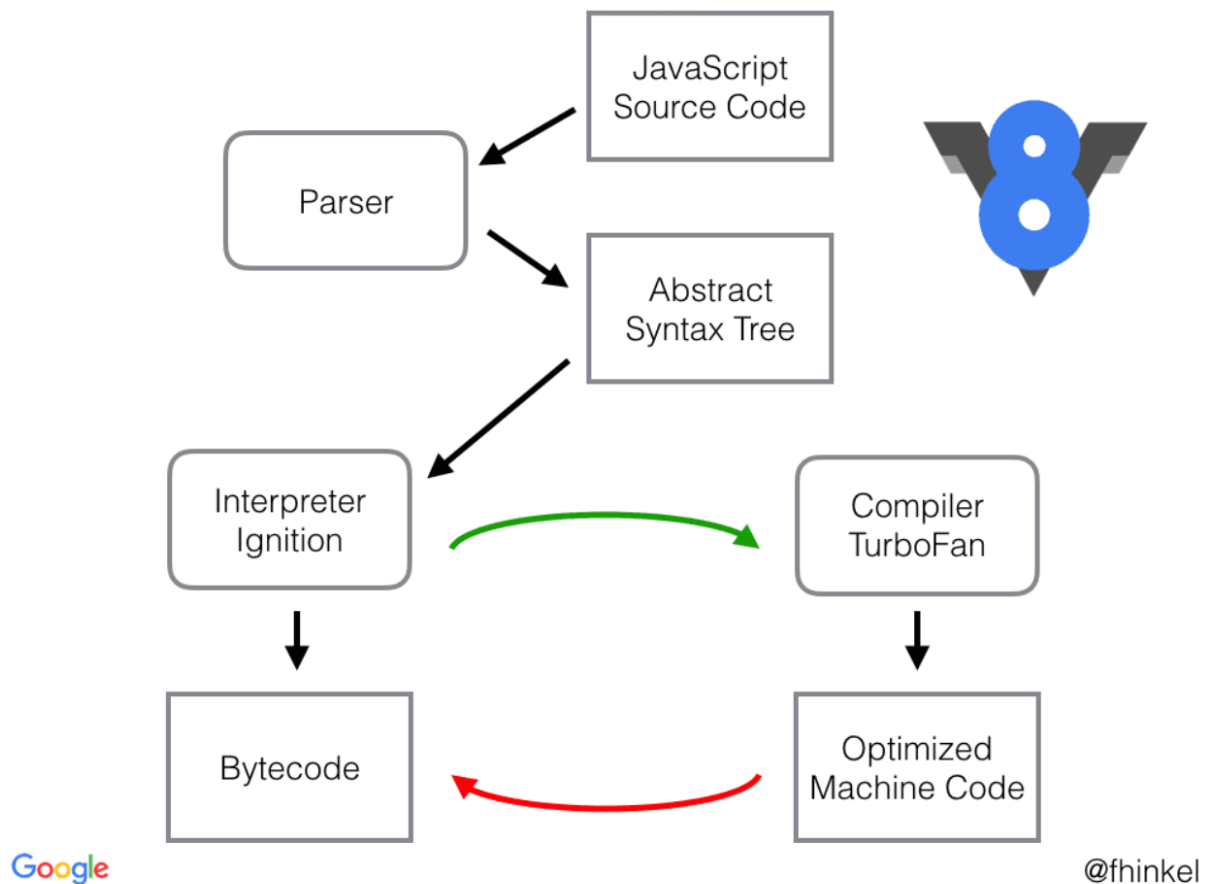
- Crankshaft는 TurboFan과 같이 바이트 코드를 최적화된 코드로 컴파일을 하는 컴파일러다.
- 원래 목적은 Ignition과 TurboFan만 이용하려고 했지만 성능이 많이 떨어져 어쩔 수 없이 함께 사용하게 됐다.

3.4 5.9 이상 V8 엔진 구조



- Ignition과 TurboFan의 성능이 좋아지면서 Full-codegen과 Crankshaft가 성능이 못 받쳐줘 제거하게 됐다.

3.5 V8 엔진 동작 과정



JSConf EU 2017에서 발표한 Franziska Hinkelmann님의 자료

- 소스 코드를 가져와서 파서에게 넘긴다. 파서는 소스 코드를 분석 후 AST(추상 구문 트리)로 변환하게 된다.
- 그 후 AST를 Ignition에 넘긴다. (Ignition은 자바스크립트를 바이트 코드로 변환하는 인터프리터)
- 이후 바이트 코드를 실행함으로 소스 코드가 실제로 작동하게 되고, 자주 사용되는 코드는 TurboFan으로 보내져 최적화된 기계어 코드(Optimized Machine Code)로 다시 컴파일 된다.
- 자주 사용되지 않으면 Deoptimizing 하기도 한다.

4. 아래의 코드들을 실행한 값을 알아오시오. 그리고 그 이유도

```
// 각각의 console.log의 출력 값을 알아오시오.  
console.log(a);  
var a;  
a= 11;  
console.log(a);
```

- 출력 결과

```
undefined  
11
```

- 이유

- `var a;` 선언문이 코드의 맨 위로 끌어올려진 것처럼 보이게 된다. 이때 `a`는 선언되었지만 초기화되지 않았으므로 값은 `undefined`이다
- `console.log(a);` 코드가 실행되며, `a`의 현재 값은 `undefined`이므로 `undefined`이 출력된다.
- `a = 11;` 코드가 실행되어 `a`에 11이 할당된다.
- `console.log(a);` 코드가 실행되며, 이번에는 `a`의 값이 11이므로 11이 출력된다.

```
// 각각의 console.log의 출력 값을 알아오시오.  
console.log(a);  
a = 10;  
var a;  
console.log(a);
```

- 출력결과

```
undefined  
10
```

- 이유

- `var a;` 선언문이 코드의 맨 위로 끌어올려진 것처럼 보이게 된다. `a` 는 선언되었지만 초기화되지 않았으므로 값은 `undefined` 이다.
- `console.log(a);` 코드가 실행되며, `a` 의 현재 값은 `undefined` 이므로 이 값이 출력된다.
- `a = 10;` 코드가 실행되어 `a` 에 10이 할당된다.
- `console.log(a);` 코드가 실행되며, 이번에는 `a` 의 값이 10이므로 10이 출력된다.

5. 헝가리안 케이스에 대해서 알아보시오

- C언어 진영에서 주로 사용했던 방식으로 변수명 앞에 자료형을 붙이는 것이다.
- C언어의 경우 메모리를 할당 및 해제가 가능하기 때문에 자료형이 명확한 것이 중요해 해당 표기법을 사용했다.
- 현재는 자료형을 쉽게 알아 볼 수 있는 다양한 방법들이 많기 때문에 권장하지 않는다.
- 최근에는 카멜이나 스네이크와 같은 표기법으로 넘어가는 추세다.
- 예시

```
String strName;
int iNum;
```