Hello world, Preparation, async versus defer

Console API는 웹 API 중에 하나다.

API는 이다.

프로그램을 쉽게 제작할 수 있게 미리 만들어 놓은 것들의 모음이라고 할 수 있다.

웹 API는 자바스크립트 언어에 포함된 것이 아니라 브라우저가 이해할 수 있는 함수들

<https://developer.mozilla.org/ko/docs/Web/API/Console_API>

MDN web docs의 Console API

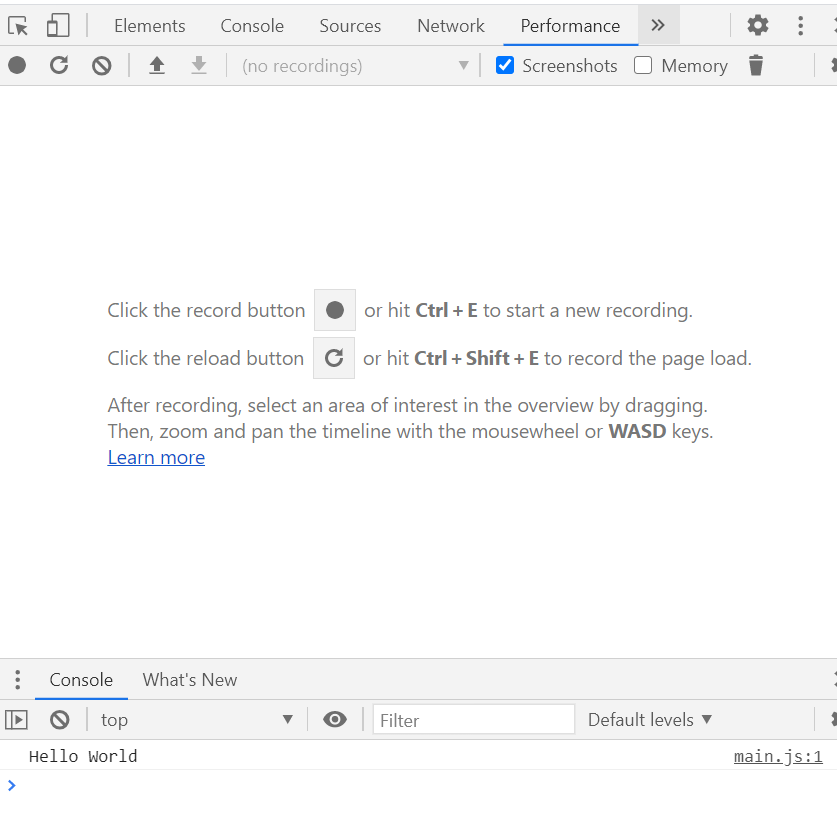
<https://console.spec.whatwg.org/>

console api spec

콘솔은 node.js 와 web api 둘 다에 포함되어 있음. 둘 다 작동 가능

Web의 console 창에서도 간단한 자바스크립트 실행 가능

Ctrl shift I 해서



자바스크립트의 공식 사이트는 ecma script이나 조금 난해할 수 있다.

그래서 수많은 개발자들이 가는 곳이 <https://developer.mozilla.org/ko/>

MDN이 근본입니다

html에서 자바스크립트를 포함하는 방법

parsing HTML (한줄씩) -> blocked [fetching js -> executing js] -> parsing html

**제일 좋은 것은 body 안의, 가장 끝부분에 js를 배치하는 것**

Parsing HTML -> fetching js -> executing js

Page가 준비된 다음 js를 받는다.

장점은 html을 빨리 볼 수 있으나, 전체 구성이 js에 의존적이라면 많이 부족한 것을 보는 셈.

의미가 없다.

**Head 안에 async 라는 속성 쓰는 방법**



Parsing 하다가 async가 있으면 병렬로 js 파일 다운 받자 명령해놓고 다시 parsing하다가 다운로드 완료되면 parsing 멈추고 js 실행한 다음 나머지 html parsing 한다.

장점은 시간 절약 가능

단점은 Html 요소를 js 코드가 요구할 때 html이 다 준비가 안 되어있으면 곤란

사용자가 페이지 보는 것에 시간 오래 걸릴 수 있다.

**defer가 가장 좋다.**



Async 옵션은 다운 받는 순으로 js를 실행하기 때문에 b가 a보다 빨리 다운되면 코드 순서 상관없이 b를 먼저 실행. 그러나 만약 b가 a를 선행적으로 필요로 한다면 문제 발생.

