학습결과서

이름 : 이승현

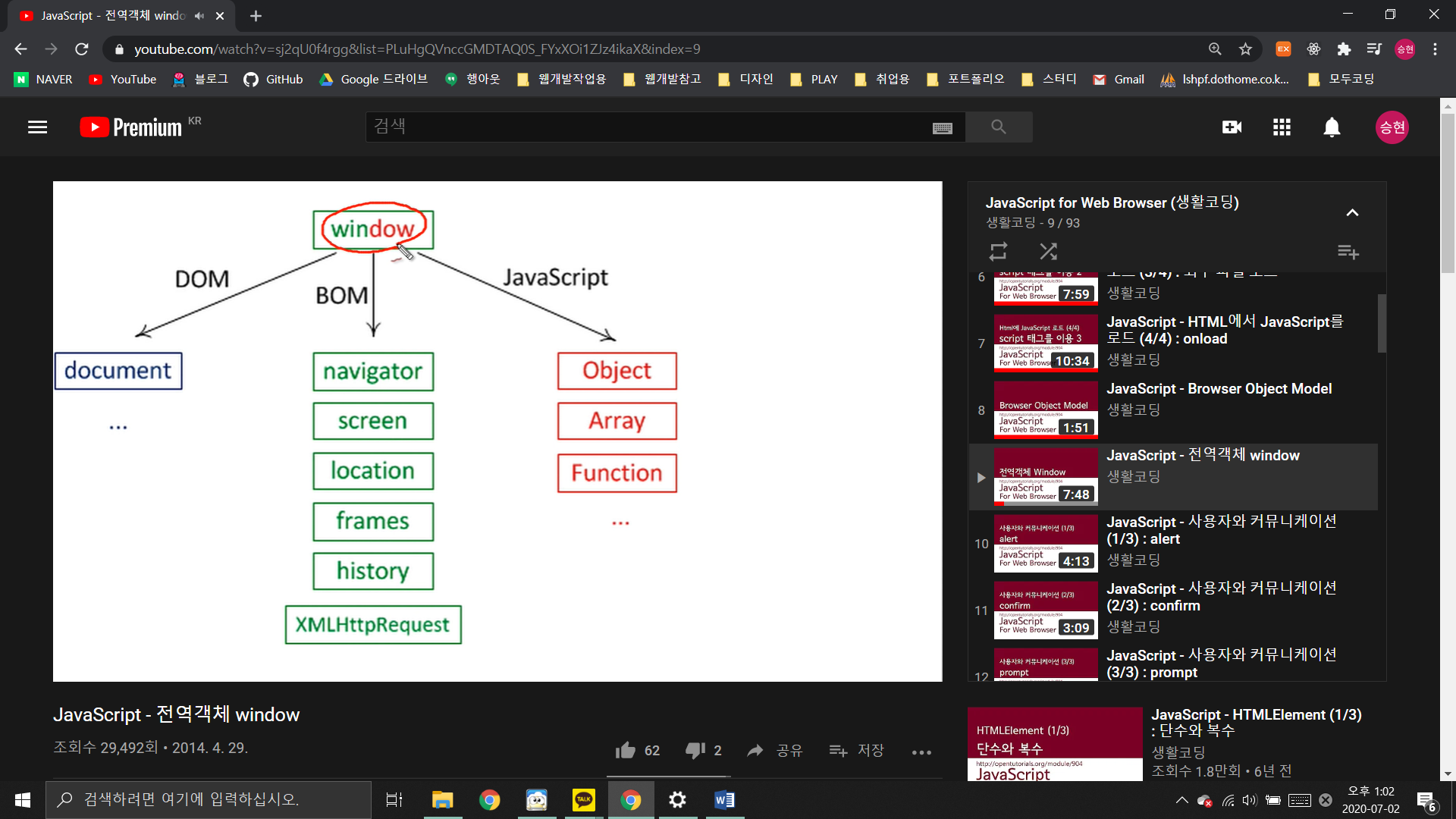
날짜 : 2020.07.02(목)

**오늘 학습 한 내용**

JavaScript for Web Browser (생활코딩)

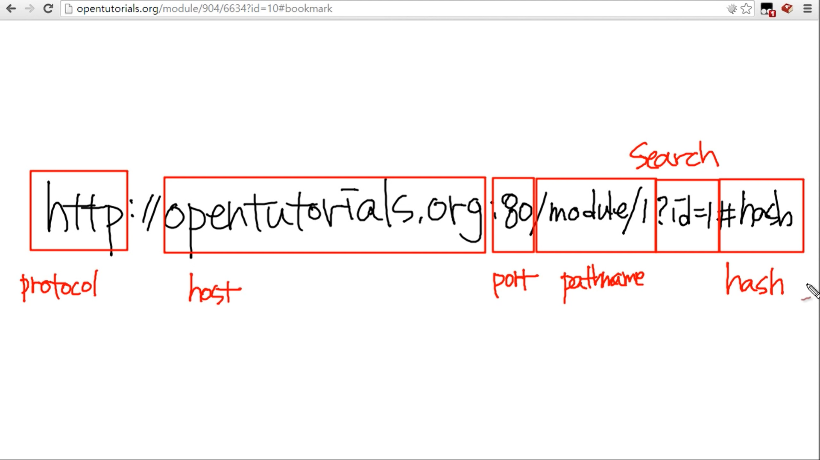
**- BOM (Browser Object Model)**

웹브라우저를 제어하기 위해서 브라우저가 제공하는 객체들. (자바스크립트를 통해 브라우저에 새창을 연다거나, 현재창의 URL을 알아낸다거나, 현재 동작하고 있는 제품명, 버전명 등을 알아낸다거나 를 할수있도록)



**- 사용자와의 커뮤니케이션 내장함수 ( alert , confirm , prompt )**

**- location 객체를 통한 문서의 주소정보 확인 및 변경 (console.log를 통해 확인)**

**location.href : 주소확인**

**location.protocol : 프로토콜 확인**

**location.host : domain 확인**

**location.port : port번호 확인**

**location.pathname : 주소에서 domain을 제외한 구체적인 정보값**

**location.search : 주소에서 ? 뒤에 오는 값**

**location.hash : 주소에서 #뒤에 오는값. Bookmark값**

**- Navigator 객체**

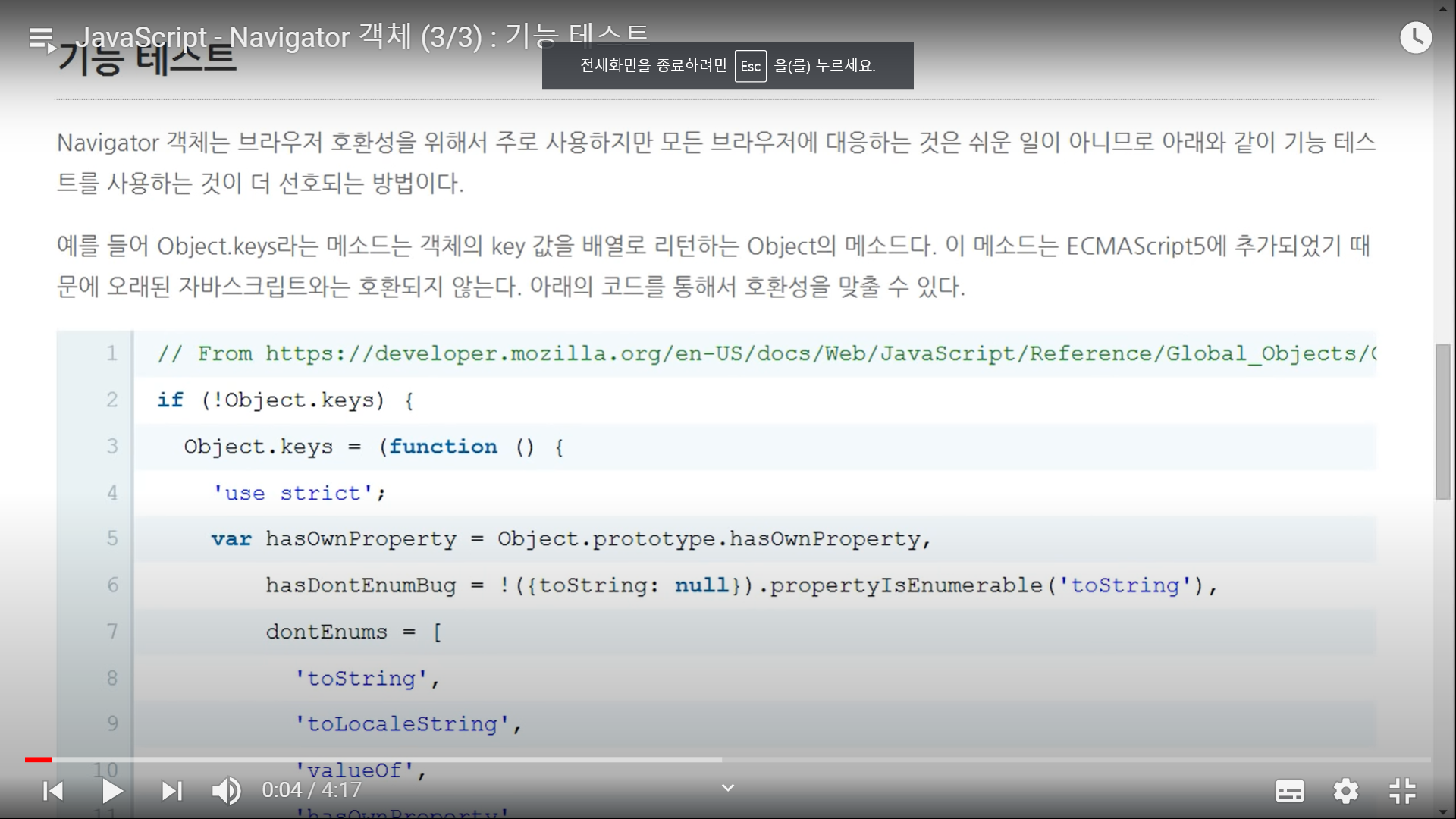
크로스 브라우징을 위해 필요한 정보를 알아오기 위한 객체.

※ console.dir을 통해 해당 객체가 가지고 있는 프로퍼티를 열람할 수 있다.

Navigator.appName / appVersion / userAgent / platform 등

Navigator 객체는 브라우저 호환성을 위해 사용되지만 모든 브라우저에 대응하는것은 쉬운일이 아니므로, 기능테스트를 사용하는것이 더 선호되는 방법이다.

예를들어, object.keys라는 것이 구 버전의 자바스크립트에서는 호환되지 않는데, 모든 브라우저에서 사용되게끔 해주고 싶으면, If( ! object.keys ) 를 이용해, object.keys 가 없다면 if문을 통과해 object.keys= function(){~ } 로 대체되어 사용하도록 설정해 준다.



**- 창 제어**

Window.open(‘주소’, 인자) ※ a태그의 target 에 들어가는 인자와 동일

인자가 없는 경우 : 이름이 붙지 않은 새 창 오픈  
인자가 \_self : 스크립트가 실행되는 창에서 오픈  
인자가 \_blank : 새 창  
인자에 이름을 넣어준다면 : 새로 오픈되는 창이 해당 이름을 갖게 되고, 다시 열릴 경우, 그 이름을 갖고있는 창에서 오픈이 일어난다.

인자를 추가로 넣어줄수 있는데, 새로 열리는 창의 사이즈 등을 지정해 줄 수 있다.

window.open 을 특정 변수에 담아, 변수.document~ 등으로 해당 창의 내용을 제어할 수 있고, 변수.close()등으로 해당 창을 종료시킬 수도 있다.

Do it! 리액트 프로그래밍 정석

**- (리액트 ES6문법 액기스) 04. 클래스**

ECMAScript 5 이전 버전의 JavaScript에는 클래스가 없었습니다. 클래스에 가장 가까운 방법은 생성자를 생성한 다음 생성자의 프로토 타입에 메서드를 할당하는 것으로, 일반적으로 사용자 정의 타입 생성이라고 하는 접근 방식입니다.

클래스는 조금 더 연관있는 데이터(속성값(field) 과 행동 (method))를 하나로 묶어놓는 컨테이너 같은 개념. 클래스 안에서도 내부적으로 보여지는 변수와 외부로 보이는 변수를 나누어서 캡슐화를 해줍니다. 클래스를 통해 상속과 다양화가 일어날 수 있습니다.

Class는 template, 즉 틀과 같은 역할을 해서, 이러이러한 데이터가 들어갈 수 있다고 정의해서 선언해주고, 실제로 데이터를 넣어 만들어 낸 것은 object 라고 합니다.

**- (리액트 ES6문법 액기스) 05. 화살표 함수**

화살표함수는 ES6에 추가된 표현식을 사용하는 함수로, 화살표기호 ( => ) 로 함수를 선언합니다.  
함수를 선언할때, function 키워드를 생략하고 인자블록 ( ) 과 본문블록 { } 사이에 =>를 표기합니다.

ex) 기존 : var add = function(first,second){ } 화살표함수 : var add = (first,second) => { }

화살표 함수를 통해, 계단형 함수 구조가 만들어지지 않게 할 수 있고, 콜백함수의 this 범위로 생기는 오류를 피하기 위해 bind() 함수를 사용하여 this 객체를 전달하는 과정을 포함하고 있어서, 해당 과정을 생략해서 사용할 수 있습니다.

**오늘 알게된 사실**

- BOM과 window내장객체에 대해 조금 알게 되었습니다. 기능테스트를 크로스브라우징에 적용시키는 법에 대해 알게 되었습니다.

**내가 부족하다고 느낀 부분**

- 아직 react공부할게 산더미입니다.

- ES6문법 중 클래스에 대해 공부해보았는데, 프로토타입이나 클래스 모두 써본적이 없는 것들이라서 개념적으로 이해도 되지않고, 클래스가 어떤면에서 좋은건지 잘 와닿지 않았습니다.

**수업에 바라는 점**

-발표형식이 통일되었으면 좋겠습니다. (공부한 모든내용을 공유하는건지, 공부한 내용중에 중요한것만 알려주는건지, 기본개념들까지 알려줘야하는건지) 다들 공통으로 준비한 내용은 제외하는게 서로의 시간을 아낄수 있을것 같습니다.