학습결과서

이름 : 이승현

날짜 : 2020.07.01(수)

**오늘 학습 한 내용**

JavaScript for Web Browser (생활코딩)

**- HTML에서 JavaScript를 로드하는 방법 4가지 ( inline / script태그 / 외부파일로드 / onload )**

**inline방식** : 정보와 제어가 같이 있기 때문에 유지보수 문제, 검색엔진이 해석하기 어려워지는 문제등이 발생할 수 있다. 이를 해결하기 위한 방법이 script태그.

**scrpit태그 방식** : inline태그보다는 정보와 제어를 구분하지만, 아직은 html안에 포함되어있는 방식.

**외부파일 로드 방식** : HTML파일이 정보’만’을 갖게됨. JS가 여러개의 HTML파일을 공통으로 사용할 수 있게 작업할 수 있음. 중복제거, 유지보수의 편의성 등의 장점 그리고 브라우저가 캐시를 통해 같은파일을 요청받았을때는 빠르게 불러오기 때문에 네트워크의 대역폭 지연현상 감소, 서버측 부담 감소, 전송과정에서 발생하는 비용감소 등 여러 장점이 있다.

**onload방식** : window.onload 가 하는 역할 => 웹페이지가 모두 읽히고, 모든 소스들이 다운로드가 끝난 후에, onload라는 메소드를 호출한다. 조금 더 빠르게 로딩되기 위해서는, body태그 끝나는 곳에 script를 호출하는것이 바람직하다.

Do it! 리액트 프로그래밍 정석

**- 리액트의 장점**

**1. 컴포넌트로 화면 구성을 효율적으로 할 수 있습니다.**

**2. 게임엔진 원리를 도입하여 화면 출력 속도가 빠릅니다.**

다음에 나타날 화면의 일부(노드)를 미리 그려놓고, 변경된 화면의 일부(노드)만 수정하는 가성화면 기술-virtual DOM을 이용하여 화면 출력 속도를 높이고 코드의 복잡성도 줄였습니다

**-웹팩의 기본개념**

프로젝트에 사용된 파일을 분석하여 기존 웹 문서 파일로 변환하는 도구입니다. 프레임워크가 js,css같은 기존 웹문서 파일을 사용하지 않기 때문에 웹팩이 필요합니다. 또한 웹팩은 js,png,jpg같은 파일을 적절한 크기로 자르거나 묶어주어 불필요한 파일을 제외하거나 압축하여 프로젝트 용량을 줄여줍니다. 그리고 웹팩은 간단한 노드기반의 웹 서버를 구동하기도 합니다.

**- (리액트 ES6문법 액기스) 01. 템플릿문자열**

기존 자바스크립트에서는 문자열과 문자열 또는 문자열과 변수를 연결하려면 병합연산자(+)를 사용해야 했습니다. 병합 연산자를 사용했을때의 코드의 복잡성을 해결하기 위해 ES6에서는 템플릿 문자열을 도입했습니다. 템플릿 문자열을 작은따옴표 대신 백틱( ` ) 으로 문자열을 표현합니다. 또한 템플릿 문자열에 특수기호 $를 사용하여 변수 또는 식을 포함할 수도 있습니다.

ex)

**(병합연산자방식)** ‘장바구니에’ + cart.name + ‘가 있습니다. 총 금액은’ + getTotal(cart) + ‘입니다.’;  
**(템플릿문자열방식)** `장바구니에 ${cart.name}가 있습니다. 총 금액은 ${getTotal(cart)} 입니다.`

**- (리액트 ES6문법 액기스) 02. 전개연산자**

전개연산자는 함수를 호출하는 인자로 배열을 사용하고 싶을 때나 배열을 정의하는 리터럴 내에서 사용할 수 있습니다. 사용 방법은 배열이나 객체, 변수명 앞에 마침표 세개 (…) 를 입력합니다. 다만, 배열, 객체, 함수 인자 표현식 ( [ ].{ },( ) ) 안에서만 사용해야 합니다.

S6이전 문법에서는 배열의 일부 요소만 잘라내거나 연결하려면 배열 인덱스와 함께 배열 내장함수들을 사용해야 했고, 객체또한 객체의 키나 값을 추출할때 객체 내장 함수를 사용했습니다. 이제는 전개연산자를 이용하여 코드를 간결하게 작성 할 수 있습니다.

ex)

**1) 전개연산자의 함수호출용의 용도**

const add = (a, b, c) => {

    return a+b+c;

}

var arr = [2, 4, 5];

add(...arr); // 11이 출력

**2) 배열 리터럴에서 전개연산자를 활용하는 방법**

var arr1 = [3, 4, 5];

var arr2 = [1, 2, ...arr1, 6, 7];

console.log(arr2);

// [1, 2, 3, 4, 5, 6, 7]

참고 : <https://pro-self-studier.tistory.com/13?category=659080>

**\*추가개념 : Rest Parameter ( 나머지 매개변수)**

함수를 선언 할 때, 해당 함수가 호출될 때 값으로 들어올 인자를 정하는 것을 매개변수라고 합니다. 이 매개변수를 지정할 때 매개변수의 이름 앞에 ...을 붙이면 이는 나머지 매개변수를 의미합니다. 그리고 이는 함수내에서 배열로 인식됩니다. 나머지 매개변수는 arguments와 다르게 유사 배열이 아닌 자바스크립트 표준 배열로 대체되고, 마지막 파라미터만 Rest 파라미터가 될 수 있습니다. 그리고 나머지 매개변수는 반드시 하나여야 합니다. 또한, arguments 객체는 호출될 때 들어온 모든 인자를 의미하지만 나머지 매개변수는 사용자가 원하는 인자만 배열로 정의할 수 있습니다.

function add2(a, b, ...rest) {

    console.log(arguments);

    console.log(rest);

}

add2(2,3,4,5,6);

첫 번째 콘솔은 유사배열로 호출될때 들어온 2, 3, 4, 5, 6 모두를 찍지만,

두 번째 콘솔은 나머지 매개변수로 정의한 4, 5, 6 만을 찍습니다.

나머지 매개변수의 효용은 인자로 들어온 값에 대해서 배열의 매서드를 이용해 어떤 작업을 하려고 할 때나 정해지지 않은 부정수인 인자들을 배열로 처리하고자 할 때 사용됩니다. arguments 객체와 분명히 다른, 좀 더 활용성이 높은 표현이다 라고 생각하시면 좋을 것 같습니다.

참고 : <https://pro-self-studier.tistory.com/15>

**- (리액트 ES6문법 액기스) 03. 가변 변수와 불변 변수**

기존 자바스크립트 문법은 변수선언에 var 키워드를 사용했지만, ES6에서는 값을 수정할 수 있는 가변 변수를 위한 let 키워드와, 값을 수정할 수 없는 불변 변수를 위한 const키워드를 사용합니다.

**가변 변수 사용법** : **let**키워드로 선언합니다. let으로 선언한 변수는 읽거나 수정할 수 있습니다.

**불변 변수 사용법** : **const**키워드로 선언합니다. const로 선언한 변수는 읽기만 가능합니다.   
불변 변수는 값을 다시 할당할 수 없는 것이지, 값을 변경할 수는 있습니다. 하지만 불변 변수로 정의된 배열이나 객체를 수정하는 것은 ‘무결성 제약조건에 위반되었다’ 라고 합니다. 무결성을 유지하면서 불변 변수의 값을 수정해야 하는 경우가 있을때에는, 수정할 불변 변수를 새로 만들어 새 값을 할당하는 방법으로 수정해야 합니다. 불변 변수를 사용하면 무결성 제약 규칙에 의해 변수가 변하는 시점을 쉽게 파악할 수 있고, 수행 전과 후의 변수값을 비교할 수 있어 가변 변수보다 더 유용합니다.

**무결성 제약조건** : 데이터베이스 상태를 만족시켜야하는 조건이고, 사용자에 의한 데이터베이스 갱신이 데이터베이스의 일관성을 깨지 않도록 보장하는 수단이다.

**오늘 알게된 사실**

- JS를 로드하는 여러 방식에 대해 알게 되었고, 왜 대부분이 외부파일 로드 방식을 쓰는지 알게되었습니다.

- ES6 문법인 템플릿 문자열 / 전개연산자 / 가변변수와 불변변수에 대하여 공부하였습니다.

-

**내가 부족하다고 느낀 부분**

- 아직 react공부할게 산더미입니다.

- ES6문법 중 가변변수/불변변수 중에서 불변변수를 사용했을때의 장점 (변수가 변하는 시점을 어떻게 아는건지, 수행 전과 후의 값을 어떻게 아는건지) 이 잘 이해가 되지 않았습니다. 추가로 찾아서 공부할 예정입니다.

**수업에 바라는 점**

- JS200 관련 책이 없어서 예제 js파일만 가지고는 어떤 공부를 해야할지 잘 모르겠습니다.