객체 및 호이스팅 정리

*//호이스팅은 피해야만 하는 메커니즘이다. var변수 선언과 함수선언에서 일어난다.*

*//변수 사용, 함수 호출이 변수 선언이나 함수 선언 보다 일찍 일어나는 경우*

*//자바스크립트가 알아서 선언부분을 최상단으로 일으키는 메커니즘*

x;*//undefined*

var x = 4; *// 4*

x; *// 4*

hoistingFunc();

function hoistingFunc(){

console.log("hi");

}

*//위의 예제를 자바스크립트가 실행할 때 해석하는 코드*

var x;

x;

var x = 4;

x;

function hoistingFunc(){

console.log("hi");

}

hoistingFunc();

*//let과 const , 함수표현식으로 할당된 에로우 펑션은 호이스팅이 안 일어난다.*

*//객체는 데이터(프로퍼티)와 기능(메서드)를 논리적으로 묶어 놓은 것을 말함*

*//생성 방법*

*// 1. 객체리터럴을 이용한 객체 직접 할당*

let leejongho = {

*//객체의 프로퍼티는 키와 밸류의 쌍으로 이루어져있다. 프로퍼티는 key : value의 형태로 작성하며,*

*//' , '쉼표로써 프로퍼티를 구분 한다.*

eye : 2,

name : "leejongho",

sayMyName : function(){

*//메서드는 메서드명 : 함수의 쌍으로 되어있다.*

*//기명함수 보다는 익명 함수를 할당하는 것이 좋다.*

console.log("hi ! my name is leejongho");

}

}

*// 2. 함수를 이용한 객체 생성 방법*

function Tv(*display*,*speaker*,*name*){

*//객체 리터럴을 반환한다.*

return {

display : display,

speaker : speaker,

modelId : name,

chanel : 7,

up : function(){

*//this라는 키워드는 객체 내부에서 자신을 가르킨다.*

*//samsungTv라면 this.chanel === samsungTv.chanel*

console.log(++*this*.chanel);

}

}

}

let samsungTv = Tv("삼성","하만카돈","hi-20");

let lgTv = Tv("lg","애플","lg-29");

*// 3. 클래스를 이용한 객체 생성 방법*

class Tv{

constructor(*display*,*speaker*,*name*){

*this*.display = display;

*this*.speaker = speaker;

*this*.name = name;

*this*.chanel = 7;

}

*//메서드 선언*

up(){

console.log(++*this*.chanel);

}

*//메서드 표현*

down = ()=>{

console.log(--*this*.chanel);

}

}

let appleTv = new Tv("apple","j&b","apple-20");

let nobrandTv = new Tv("?","?","?");

*//객체 프로퍼티에 접근 하려면 다음과 같이 객체명 뒤에 '.'을 붙여 프로퍼티명을 적는다.*

console.log(appleTv.chanel) *// 7*

console.log(appleTv.display)*// apple*

*//객체 메서드를 호출 하려면 다음과 같이 객체명 뒤에 '.'을 붙여 메서드명을 적고 호출자를 적는다.*

appleTv.up(); *// console.log(8);*

appleTv.down() *// console.log(7);*

appleTv.down() *//console.log(6);*