**front\_study\_day6(JavaScript)**

**신찬우**

**기본 문법**

1. **클래스(Class)**
   1. **Class declaration(클래스 선언)**
      1. class name { constructor() { fields…} methods …}
   2. **Getter and Setter**
      1. get func(), set func(val) this.\_name (이런 식으로 작성해야함)
   3. **Fields (public, private)**
      1. **#**private = 0; (#으로 private 표현)
   4. **Static properties and methods**
      1. e.g. **static** publisher = "chan"; ( object가 아니고 class 자체에 할당이 됨.)
      2. object 생성 후 호출 시 : objectName(x), className(o)
         1. Article.publisher (class 이름)
   5. **Inheritance( 상속 .. +다형성 )**
      1. e.g.

class Shape {}

class Triangle **extends** Shape {} // Shape 클래스를 그대로 상속받음

* + 1. **Class checking (부모 클래스에서 상속된 인스턴스가 맞는지)**
  1. e.g. Triangle **instanceof** Shape ; => boolean 값 출력(true)

1. **오브젝트**
   1. **Literals and properties (오브젝트 기본 문법)**
      1. object literal

obj1 = {};

1. object constructor

obj2 = new Object();

1. const name = { key1: value1, key2: value2 }
2. **Computed properties**
   1. 일반적으로 object.key 로 사용(여기에서 object는 class에서 생성된 object, key는 object에 포함되어있는 key)
   2. computed properties 는 [] 사용, object[key]
   3. 실시간으로 오브젝트를 추가하거나, key, value를 확인하고 싶을 때
3. **Property value shorthand**
   1. 오브젝트를 만들어서 리턴하는 함수를 작성

const obj1 = funcName(val1, val2);

function funcName(val1, val2) {

return { // 오브젝트 생성

val1,

val2,

};

}

1. **Constructor Function**
   1. 함수가 아닌, 클래스를 만들어서 사용

cosnt obj2 = new funcName(val1, val2);

function funcName(param1, param2) {

this.key1 = param1;

this.key2 = param2;

}

1. **in operator(오브젝트에서 key값을 확인 할 때)**
   1. 사용법 : key in object; -> boolean 값으로 출력
2. **for .. in, for .. of (key와 value 관계)**
   1. for (key in obj)
   2. for (val of iterable) (배열을 사용)
3. **cloning (오브젝트 복사)**
   1. obj2 = obj1; // obj1은 ref(주소)를 가지고 있음, obj2도 주소를 복사 받음
   2. obj2 = {};

for (key in obj1) { obj2[key] = obj1[key]; } // 반복으로 오브젝트 생성

1. obj2 = Object.assign({}, obj1); // Object 클래스 사용

Object.assign({}, obj1, obj2, obj3 …) // 뒤에 나오는 키 값으로 덮어씀