Object Literal

Rolo - Nomad Coders Napolitan

객체

0개 이상의 프로퍼티로 구성된 집합

프로퍼티

키와 값으로 구성됨 { 프로퍼티 키 : 프로퍼티 값 }

프로퍼티 키 : 빈 문자열을 포함하는 모든 문자열 또는 심벌값

프로퍼티 값 : 자바스크립트에서 사용할 수 있는 모든값,

* 함수도 프로퍼티 값으로 사용할 수 있다. (메서드)

메서드

객체에 묶여있는 함수. 프로퍼티 값 위치에서 생성된 함수

프로퍼티를 참조하고 조작할 수 있는 동작

다음 중 에러가 나는 객체 생성 코드를 모두 고르시오

여러개의 프로퍼티를 사용할시,
콤마로 구분해줘야 한다.

```
const me1 = {
  name: "rolo"
  job: "student"
};
```

```
Uncaught SyntaxError: Unexpected identifier 'job'
```

```
const me2 = {
  name: "rolo",
  job: "student",
};
```

```
▶ {name: 'rolo', job: 'student'}
```

```
const me3 = {
  name: "rolo",
  job: "student"
};
```

```
▶ {name: 'rolo', job: 'student'}
```

마지막 프로퍼티 값은 콤마를 넣어주지 않아도 동작한다.

```
const me4 = {
    "": "",
};
```

```
▶ {"": ''}
```

빈프로퍼티키, 빈키값으로 객체를
생성해도 에러가 발생하지 않는다.

다음 객체들의 console.log 결과를 예상하시오

```
const me5 = {
  firstName: "rolo",
  "last-name": "choi",
  본업: "student",
  SecondJob: "cafeOwner",
};
console.log(me5);
```

- 프로퍼티 키에는 식별자 생성규칙이 적용됨
- 카멜케이스, 스네이크헤이스, 파스칼 케이스 모두 사용가능
- 프로퍼티 키에 한국어 사용가능
- 프로퍼티 키에 일반 -을 사용할 경우,
  마이너스 기호로 인지하므로, 식별자규칙에 위배되는
  문구를 사용할 경우, "" 따옴표로 묶어줘야 한다.

```
▶ {firstName: 'rolo', last-name: 'choi', 본업: 'student', SecondJob: 'cafeOwner'}
```

```
const me6 = {};
const key = "name";
me6[key] = "rolo";
console.log(me6);
```

동적으로 객체를 생성하는 방법

먼저 빈 객체를 생성하고, 프로퍼티 key 값을 만든다.
키에 들어갈 값을 [key]로 준뒤, 프로퍼티 값을 넣어준다.

```
▶ {name: 'rolo'}
```

다음 객체들의 console.log 결과를 예상하시오

```
const me7 = {
  1: "rolo",
  2: "student",
  3: "cafeOwner",
};
console.log(me7);
```

프로퍼티 키값으로 숫자가 들어오면 암묵적으로 타입이 변경된다.
숫자를 키값으로 사용하여 객체를 생성할 수 있지만,
혼동을 유발할 여지가 있으므로, 사용 하지 않는 것이 좋다.

```
▶ {1: 'rolo', 2: 'student', 3: 'cafeOwner'}
```

```
const me8 = {
  name: "jieun",
  name: "rolo",
};
console.log(me8);
```

동일한 프로퍼티 키로 다른 프로퍼티 값을 주면,
자동으로 재할당된다. (덮어쓰기 된다)

```
▶ {name: 'rolo'}
```

다음 객체의 "last-name"과 본업을 console.log 하시오

```
const me5 = {
  firstName: "rolo",
  "last-name": "choi",
  본업: "student",
  SecondJob: "cafeOwner",
};
```

식별자 규칙을 따르지 않는 키. 즉, 생성할때 따옴표안에 생성한 키 값은 대괄호안에 따옴표 그대로 넣어줘야 값을 불러올 수 있다.

한국말은 키값으로 인식을 하고, 불러올때는 대괄호나 마침표 방식 둘다 불러올 수 있다.

```
> console.log(me5[last-name])
✕ ▶Uncaught ReferenceError: last is not defined
      at <anonymous>:1:17
> console.log(me5["last-name"]);
  choi
‹· undefined
> console.log(me5.last-name)
  NaN
‹· undefined
> console.log(me5."last-name")
✕ Uncaught SyntaxError: Unexpected string
```

```
> console.log(me5[본업]);
✕ ▼Uncaught ReferenceError: 본업 is not defined
      at <anonymous>:1:17
    (anonymous) @ VM1553:1
> console.log(me5["본업"]);
  student
‹· undefined
> console.log(me5.본업);
  student
‹· undefined
> console.log(me5."본업");
✕ Uncaught SyntaxError: Unexpected string
```

다음 console.log 결과들을 예상하시오

```
const counter = {
  num: 0, //프로퍼티
  //메서드
  increase: function () {
    this.num++;
  },
};
counter.increase();
counter.increase();
console.log(counter.num);
```

2

```
const person_hello = {
  name: "Rolo",
  sayHello: function () {
    console.log(`Hello! My name is ${this.name}.`);
  },
};
console.log(person_hello.sayHello());
```

Hello! My name is Rolo.

다음 객체의 SecondJob을 삭제하시오

```
const me5 = {
  firstName: "rolo",
  "last-name": "choi",
  본업: "student",
  SecondJob: "cafeOwner",
};
```

```
delete me5.SecondJob;
console.log(me5);
```

```
▶ {firstName: 'rolo', last-name: 'choi', 본업: 'student'}
```

다음 코드를 ES6 방식으로 변경하시오

```es5
//ES5
var x = 1, y = 2;
var obj = {
    x: x,
    y: y,
};
```

```es6
//ES6
let x = 1, y = 2;
const obj = {x, y};
```

```es5
//ES5
var obj = {
    name: "Lee",
    sayHi : function() {
        console.log("Hi " + this.name);
    }
};
console.log(obj.sayHi());
```

```es6
//ES6
const obj = {
    name: "Lee",
    sayHi(){
        console.log(`Hi ${this.name}`);
        }
    };

    console.log(obj.sayHi());
```

Extra : Destructuring assignment (구조 분해 할당 36장)

```
const user = { firstName: "Rolo", lastName: "Choi" };
const { lastName, firstName } = user;
console.log(firstName, lastName);
```

```
Rolo Choi
```

```
const member = {
  name: "Choi",
  address: { zipCode: "66646", city: "Apodaca"},
};
const { address: { city }, name} = member;
console.log(city);
console.log(name);
```

```
Apodaca
Choi
```