

| 카카오톡 봇 강좌 | >

객체(Object)

 | 화베 | 챗봇 고수 🏆 1:1 채팅

2020.02.11. 06:25 조회 221

💬 댓글 22

🔗 URL 복사

⋮

자바스크립트에서 객체란, 키(속성)과 값의 쌍들을 가지고 있는 구조를 말합니다.

자바스크립트 객체의 종류에는 크게 세가지가 있습니다.

- 1. 생성자 함수
- 2. 객체 리터럴
- 3. 배열객체

※ 생성자 함수는 심화 과정이니 정의만 배우고, 자세한건 객체 리터럴만 배우겠습니다. 배열 객체는 객체 리터럴을 할때 같이 설명하겠습니다.

1. 생성자 함수

생성자 함수란 이름에서 볼 수 있듯이 함수를 객체로써 사용하는것을 말합니다.

우리가 흔히 사용하지만 생성자 함수인지 몰랐던 생성자 함수에는 "Date"가 있습니다.

"Date" 생성자 함수를 쓸때는 new Date().속성 같은 형식으로 쓰죠?

이렇듯이 생성자 함수를 사용할때는 앞에 new 연산자를 붙입니다. new 연산자는 "뒤에있는 객체를 인스턴스화 시키겠다." 라는 기능을 가진 연산자로,

"인스턴스"라는것은 객체와 비슷하지만 조금 다릅니다.

객체란 코드상의 구조 그 자체를 말하고,

인스턴스란 코드상에서 사용할 수 있는 실체를 말합니다.

즉 인스턴스화 한다는것은, "실제로 사용할 수 있는 형태로 바꾼다" 라고도 말할 수 있습니다.

2. 객체 리터럴

객체 리터럴은 그냥 줄여서 객체라고도 부릅니다. 왜냐하면 그 형태가 기본적인 객체의 형태이기 때문이죠.

객체 리터럴은 변수로 선언하거나, ()로 감싸면 그 자체로 인스턴스화 되기 때문에 new 연산자를 사용해서 인스턴스화 할 필요가 없습니다.

객체 리터럴을 만드는 방법은

{ "키" : 값, "키": 값 } 과 같은 형태로 만들면 됩니다.

예를들어서 A라는 변수에 B:1, C:2 의 (키,값)쌍이 있는 객체를 저장하고 싶다면,

```
var A = {  
  B: 1,  
  C: 2  
};
```

처럼 하면 됩니다.

2-1. 값을 불러오거나 저장하기

A라는 객체에 있는 값들을 가져오려면

A.키 또는 A["키"] 와 같이 사용하시면 됩니다.

추가적으로 객체에 특정 키에 대한 값을 저장하고 싶다면

A.키 = 값; 또는 A["키"] = 값;

그런데, 2번째 형태를 어디서 많이 보지 않았나요?

그렇습니다. 바로 배열에서 보았죠.

배열 또한 "객체" 입니다.

0,1,2,3 과 같은 "인덱스"와

push, shift, slice와 같은 배열 관련 함수들을 "키" 로 가지고 있는 "객체"죠.

그렇기 때문에 `var A = [];` 라고 해놓고
`A.B = 1;` 와 같이 사용할 수 있습니다.
우리는 이것을 "배열 객체" 라고 부릅니다.

3. 관련 함수들

객체와 관련된 함수들에는 뭐가 있을까요?

대표적으로는 `Object.defineProperty`, `Object.getOwnPropertyNames`,
`Object.keys` 등이 있습니다.

`Object.defineProperty`는 이름에서 볼 수 있듯이

특정 객체에 특정 속성을 저장하는 함수입니다.

그런데 그냥 객체.속성 = 값; 을 사용하면 되지 왜 이런 복잡한 함수를 사용할까요?

그 이유는 `Object.defineProperty`는 속성을 정의할 때 삭제가 가능한가? 수정이 가능한가? 등의 세부적인 조건들을 달 수 있기 때문입니다.

사용법은 `Object.defineProperty(객체, "속성", 세부조건 객체 리터럴)`

입니다. 값이 들어가지 않는 이유는 세부조건 안에서 값을 지정할 수 있기 때문입니다.

각각의 세부조건들은 기본(Default)값들이 있는데, 그중 하나가 "수정 불가" 입니다. 이때문에 유용하게 쓰입니다.

마치 상수처럼 값을 바꾸지도, 값을 삭제하지도 못하는거죠.

값을 지정하는 세부조건들의 이름은 `value`입니다.

예를 들어서 A라는 객체에 B:2 라는 속성과 값을 지정하고 싶다면

`Object.defineProperty(A, "B", {value: 2});`

와 같이 사용하면 됩니다.

`Object.getOwnPropertyNames`와 `Object.keys`는 둘다 객체가 가지고 있는 속성들을 배열에 담아 가져옵니다.

그런데 왜 똑같은 기능의 함수가 2개나 있을까요?

아까 `Object.defineProperty` 를 배웠죠?

3번째 인자에 들어가는 세부조건중에는 `enumerable`이라는 것이 있습니다. 기능은 "속성 열거(읽기) 가능 여부" 이며, 기본 값은 거짓(false)입니다.

즉, `Object.keys`는 `enumerable`이 true인 속성들만 가져오고, `Object.getOwnPropertyNames`는 모든 속성을 가져온다는 차이가 있습니다.

이상 객체(Object) 강의였습니다.



| 화베 | 님의 게시글 더보기 >

❤️ 좋아요 1 💬 댓글 22

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐



| 화베 | 작성자

와!

2020.02.11. 06:57 답글쓰기



줄러

빨랑 `Array.prototype.reduce` 써오세웁

2020.02.11. 06:57 답글쓰기



| 화베 | 작성자

빨글줄 올려줘요 그럼


2020.02.11. 06:58 답글쓰기




줄러

| 화베 | 그러기엔 제 글의 대부분이 빨글이라 양심에 찔려요


2020.02.11. 06:58 답글쓰기




| 화베 | 작성자
줄러 그러면 Array.isArray 강의 올려요
2020.02.11. 06:58 답글쓰기




줄러
| 화베 | 어라?
2020.02.11. 06:59 답글쓰기




마른얼음
configurable writable도 써줘요 음음
2020.02.11. 06:59 답글쓰기




| 화베 | 작성자
생략
2020.02.11. 06:59 답글쓰기



마른얼음
| 화베 | 나!
2020.02.11. 06:59 답글쓰기



| 화베 | 작성자
마른얼음 이전글도 댓글좀 ㅠ
2020.02.11. 07:00 답글쓰기



CodeMonkey
let a = [];

var b= {
0: 'a',
1: 'b',
2: 'c',
length: 3
};

a.0 = 'T';
a[0] = 'a';
a[1] = 'b';
a[2] = '3';

원래 배열이 자바 스크립트에선 객체라서 프로퍼티로 접근할때 만들어지는것으로 알고있습니다

그래서 a를 출력하면

0: "a"
1: "b"
2: "c"
b: "T"
length: 3
__proto__: Array(0)

이런식으로 뜨죠 화베님이 했던말과 같은 말이긴합니다

하지만 b처럼 리터럴 객체를 생성할때 배열인것처럼 한다면 이것을 유사배열 또는 연관 배열이라고 불립니다.
실제로 쓰는 경우도 종종 있고 Dom인 브라우저 .querySelectorAll 또는 .children 에서도 이것은 배열이 아니라 유사배열이죠
b를 찍어보면

0: "a"
1: "b"
2: "c"
3: "d"
length: 3
__proto__: Object

확인해보면 Object라고 되어있는것처럼 배열이 아닌 객체가 찍혀있는거 확인할수있습니다
2020.02.11. 09:22 답글쓰기



| 화배 | 작성자

객체.키 로 접근할때
변수 선언과 마찬가지로 앞글자 숫자는 안됩니다.

2020.02.11. 15:05 답글쓰기



| 화배 | 작성자

a.0 오류나요

2020.02.11. 15:05 답글쓰기



AlphaDo

| 화배 | 대괄호..

2020.02.11. 17:52 답글쓰기



CodeMonkey

| 화배 | 아 a.0 잘못된 코드입니다 원래 a.b 적었는데 잘못된코드 적고 후에 고쳤는데 그걸 반영안했네요

2020.02.11. 18:30 답글쓰기



AlphaDo

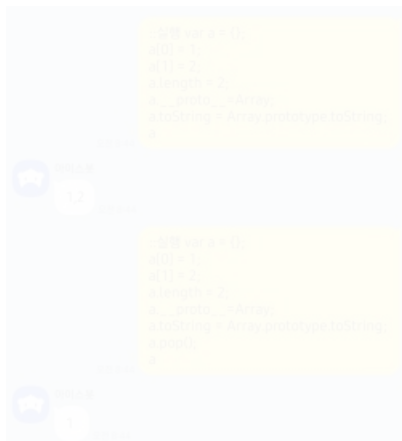
CodeMonkey 억 까르륵

2020.02.11. 22:54 답글쓰기



| 화배 | 작성자

그냥 배열로 만들어 버리기 ㅎㅎ



2020.02.12. 08:45 답글쓰기



AlphaDo

| 화배 | ㅋㅋㅋㅋㅋㅋㅋㅋ

2020.02.12. 10:49 답글쓰기



CodeMonkey

| 화배 | 확인결과 __proto__에 직접 Array를 넣는다고 배열이 되진않습니다

그 증거로 값을 추가하여 넣을경우 length는 변하지않으며 __proto__를 확인해보면 array가 아닌 function이 붙어있을것입니다. 그리고 그 function 안에 prototype에 배열의 함수들이 있습니다.
다시말해 Array.prototype.~~ 하는것과 크게 다를바가 없을뿐 아니라 위에서 직접 넣은 __proto__ = Array 의 결과는 a는 함수가 되었다는점이죠

2020.02.12. 11:12 답글쓰기



CodeMonkey

CodeMonkey 다면 __proto__에서 new Array() 로 생성한다면 배열과 동일하진않지만 배열 거의 동일한 놈으로 변한 합니다

2020.02.12. 11:26 답글쓰기



AlphaDo

아 defineProperty가 저거였군

2020.02.11. 14:39 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'카카오봇 강좌' 게시판 글

이 게시판 새글 구독하기 ☐

Node.js Setting - Install	CodeMonkey	2020.02.11.
Node.js ajax 통신	CodeMonkey	2020.02.11.
객체(Object)	화베	2020.02.11.
&와 &&, 와 의 차이	화베	2020.02.11.
[팁] console.log 를 카톡에서도!	jomin398	2020.02.03.

1 2 3

전체보기

이 카페 인기글



태양, 달 정보 구현 완료

Dark Torado

마녀사냥 금지

AlphaDo

틱택토 (Player vs Player)

안녕하세요

기 나무 곡경에[노말]의
피막의 문들 중 하나를

5000분의 1을 얻은 사람

분초보안니당

eval 질문

Kiri

kakao.py에 대해서

undefined

[카카오봇] 양산형 자동학습 봇 소스 공유

에뮬레이터로 카톡 돌리면 정지되나요?

공돌이

1 2 3 4 5