Javascript Objects

- description : Javascript Object Properties
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-29

The source of this article

Javascript Object Properties
속성(properties)은 JavaScript 오브젝트에서 가장 중요한 부분입니다.

JavaScript Properties

속성(properties)은 JavaScript 오브젝트와 관련된 값입니다.

JavaScript 오브젝트는 정렬되지 않은 속성의 모음입니다.

속성은 일반적으로 변경, 추가 및 삭제할 수 있지만, 일부는 읽기 전용입니다.

Accessing JavaScript Properties

오브젝트 속성에 엑세스하기 위한 구문은 다음과 같습니다:

```
objectName.property     // person.age
```

또는

```
objectName["property"]  // person["age"]
```

또는

```
obejctName[expression]  // x = "age"; person[x]
```

```
표현식(expression)은 속성 이름으로 평가되어야 합니다.
```

Example 1

```
let person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};

document.getElementById("demo").innerHTML =
  person.firstName + " is " + person.age + " years old."; // John is 50
years old.
```

Example 2

```
let person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};

document.getElementById("demo").innerHTML =
person["firstName"] + " is " + person["age"] + " years old."; // John is 50
years old.
```

JavaScript for...in Loop

JavaScript `for...in` 스테이트먼트는 오브젝트 속성을 반복합니다.

Syntax

```
for (variable in object) {
  // code to be executed
}
```

`for...in` 루프 내부의 코드 블록은 오브젝트의 속성을 반복하면서 각 속성에 대해 한 번씩 실행됩니다.

Example

```
let txt = "";
```

```
let person = { fname: "John", lname: "Doe", age: 25 };
let x;
for (x in person) {
  txt += person[x] + " ";
}
document.getElementById("demo").innerHTML = txt;  // John Doe 25
```

Adding New Properties

단순히 값을 제공하여 기존 오브젝트에 새로운 속성을 추가할 수 있습니다.

person 오브젝트가 이미 존재한다고 가정합니다. 그런 다음 새 속성을 부여할 수 있습니다.

Example

```
let person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyecolor: "Blue"
};

person.nationality = "English";
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.nationality + ".";

console.log(person);
// {firstName: "John", lastName: "Doe", age: 50, eyecolor: "Blue",
nationality: "English"}
```

Deleting Properties

delete 키워드는 오브젝트에서 속성를 삭제합니다.

Example

```

```

delete 키워드는 속성 값과 속성 자체를 모두 삭제합니다.

삭제 후에는, 다시 추가하기 전에는 속성을 사용할 수 없습니다.

`delete` 연산자는 오브젝트 속성에 사용하도록 설계되었습니다. 변수 또는 함수에는 영향을 주지 않습니다.

`delete` 연산자는 미리 정의된 JavaScript 오브젝트 속성에 사용하면 안됩니다. 응용 프로그램이 충돌할 수 있습니다.

Property Attribute

모든 속성에는 이름이 있습니다. 또한 속성에는 값(value)이 있습니다.

값은 속성의 속성 중 하나입니다(The value is one of the property's attributes)

기타 속성(attributes)은 열거 가능(enumerable), 구성 가능(condfigurable) 및 쓰기 가능(writable)입니다.

이러한 속성(attributes)은 속성(property)에 액세스 할 수 있는 방법을 정의합니다 (읽기 가능합니까?, 쓰기 가능합니까?)

JavaScript에서는 모든 속성(attributes)을 읽을 수 있지만, value 속성(attribute) 만 변경할 수 있습니다 (속성(property)이 쓰기 가능한 경우에만).

(ECMAScript 5에는 모든 속성 속성을 가져오고 설정하는 방법이 있습니다)

Prototype Properties

JavaScript 객체는 프로토타입의 속성(properties)을 상속받습니다.

`delete` 키워드는 상속된 속성을 삭제하지 않지만, 프로토타입 속성을 삭제하면, 프로토타입에서 상속된 모든 객체에 영향을 줍니다.

오션, Javascript, Objects, Properties

From:
http://rwiki.repia.com/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다.** - 2023.12

Permanent link:
http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_object_properties&rev=1619674917

Last update: **2022/03/10 19:52**