Javascript Objects

- description : Javascript Object Accessors
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-05-03

The source of this article

Javascript Object Accessors

JavaScript Accessors (Getters and Setters)

ECMAScript 5 (2009)는 Getter 및 Setters를 도입했습니다.

Getter 및 Setter를 사용하면 오브젝트 접근자(계산된 프로퍼티)를 정의할 수 있습니다.

JavaScript Getter (The get Keyword)

다음 예제에서는 `lang` 속성을 사용하여 `language` 프로퍼티의 값을 가져옵니다(`get`).

Example

```
// Create an object
let person = {
  firstName: "John",
  lastName: "Doe",
  language: "en",
  get lang() {
    return this.language;
  }
};
// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;  // en
```

JavaScript Setter (The set Keyword)

다음 예제에서는 lang 속성을 사용하여 language 프로퍼티의 값을 설정합니다(set).

Example

```
// Create an obejct
let person = {
  firstName: "John",
  lastName: "Doe",
  language: "No",
  set lang(value) {
    this.language = value;
  }
};
console.log(person);  // {firstName: "John", lastName: "Doe", language:
"No"}
// Set a property using set:
person.lang = "French";
// Display data from the object:
document.getElementById("demo").innerHTML = person.language;  // French
console.log(person);  // {firstName: "John", lastName: "Doe", language:
"French"}
```

JavaScript Function or Getter?

다음 2 예제의 차이점은 무엇입니까?

Example 1

```
// Create an object;
let person = {
  firstName: "Emile",
  lastName: "Darwin",
  fullName: function () {
    return this.firstName + " " + this.lastName;
  }
};
// Display data from the object using a method;
document.getElementById("demo").innerHTML = person.fullName();  // Emile
Darwin
```

Example 2

```
// Create an object;
let person = {
  firstName: "Emile",
  lastName: "Darwin",
  get fullName() {
    return this.firstName + " " + this.lastName;
  }
};
// Display data from the object using a getter;
document.getElementById("demo").innerHTML = person.fullName;  // Emile
Darwin
```

Example 1은 함수: person.fullName()로 fullName에 액세스합니다.

Example 2는 프로퍼티: person.fullName로 fullName에 액세스합니다.

두 번째 예제는 더 간단한 구문을 제공합니다.

Data Quality

JavaScript는 getter 및 setter를 사용할 때 더 좋은 데이터 품질을 보장할 수 있습니다.

다음 예제에서, `lang` 프로퍼티를 사용하면 대문자로 `language` 프로퍼티 값을 반환합니다.

Example

```
// Create an object:
let person = {
  firstName: "Elizabeth",
  lastName: "Taylor",
  language: "en",
  get lang() {
    return this.language.toUpperCase();
  }
};
// Display dafta from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;  // EN
```

다음 예제에서는 `lang` 프로퍼티를 사용하여 `language` 속성에 대문자 값을 저장합니다.

Example

```
// Create an object:
let person = {
  firstName: "Elizabeth",
  lastName: "Taylor",
  language: "",
  set lang(lang) {
    this.language = lang.toUpperCase();
  }
};
// Set a property using set:
person.lang = "en";
// Display data from the object:
document.getElementById("demo").innerHTML = person.language;  // EN
```

Why Using Getters and Setters?

- 더 간단한 구문을 제공합니다.
- 프로퍼티 및 메서드에 대해 동일한 구문을 허용합니다.
- 더 좋은 데이터 품질을 확보할 수 있습니다.
- behind-the-scenes에서 작업하는 데 유용합니다.

Object.defineProperty()

`Object.defineProperty()` 메서드를 사용하여 Getter 및 Setter를 추가할 수도 있습니다:

Example

```
```

오션, Javascript, Objects, Accessors

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:js_object_accessors&rev=1620006393

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