

JavaScript Naming Conventions

- description : JavaScript Naming Conventions
- author :
- email : shlim@repia.com
- lastupdate : 2022-04-06 Wed

The source of this article

[JavaScript Naming Conventions](#)

10 JavaScript Naming Conventions Every Developer Should Know

가 . 가

10가 JavaScript .

1. Naming Convention for Variables

JavaScript

```
var DogName = 'Scooby-Doo';
var dogName = 'Droopy';
var DOGNAME = 'Odie';
console.log(DogName); // "Scooby-Doo"
console.log(dogName); // "Droopy"
console.log(DOGNAME); // "Odie"
```

JavaScript 가 camel case .

JavaScript camel case , 가

가

```
// bad
var dogname = 'Droopy';
// bad
var dog_name = 'Droopy';
// bad
var DOGNAME = 'Droopy';
// bad
var DOG_NAME = 'Droopy';
// good
var dogName = 'Droopy';
```

가 (, Name) 가 dogName .

```
// bad
var d = 'Scooby-Doo';
// bad
var name = 'Scooby-Doo';
// good
var dogName = 'Scooby-Doo';
```

2. Naming Convention for Booleans

Boolean , is has Boolean 가 , hasOwner .

```
// bad
var bark = false;
// good
var isBark = false;
// bad
var ideal = true;
// good
var areIdeal = true;
// bad
var owner = true;
// good
var hasOwner = true;
```

3. Naming Convention for Functions

JavaScript

camel case

,
, getName 가

```
// bad
function name(dogName, ownerName) {
  return `${dogName} ${ownerName}`;
}

// good
function getName(dogName, ownerName) {
  return `${dogName} ${ownerName}`;
}
```

4. Naming Convention for Constants

JavaScript

```
var LEG = 4;
var TAIL = 1;
var MOVABLE = LEG + TAIL;
```

가 , UPPER_SNAKE_CASE

```
var DAYS_UNTIL_TOMORROW = 1;
```

5. Naming Convention for Classes

JavaScript

Pascal case

```
class DogCartoon {
  constructor(dogName, ownerName) {
    this.dogName = dogName;
    this.ownerName = ownerName;
  }
}

var cartoon = new DogCartoon('Scooby-Doo', 'Shaggy');
```

6. Naming Convention for Components

JavaScript

React

Pascal case

DOM

```
// bad
function dogCartoon(roles) {
  return (
    <div>
      <span> Dog Name: { roles.dogName } </span>
      <span> Owner Name: { roles.ownerName } </span>
    </div>
  );
}

// good
function DogCartoon(roles) {
  return (
    <div>
      <span> Dog Name: { roles.dogName } </span>
      <span> Owner Name: { roles.ownerName } </span>
    </div>
  );
}
```

HTML

```
<div>
  <DogCartoon
    roles={{ dogName: 'Scooby-Doo', ownerName: 'Shaggy' }}
  />
</div>
```

7. Naming Convention for Methods

, JavaScript

JavaScript

≠ camel case

```

class DogCartoon {
  constructor(dogName, ownerName) {
    this.dogName = dogName;
    this.ownerName = ownerName;
  }

  getName() {
    return `${this.dogName} ${this.ownerName}`;
  }
}

var cartoon= new DogCartoon('Scooby-Doo', 'Shaggy');

console.log(cartoon.getName());
// "Scooby-Doo Shaggy"

```

8. Naming Convention for Denoting Private Functions

(_) MySQL PHP
JavaScript private

toonName

(_toonName)

가

```

class DogCartoon {
  constructor(dogName, ownerName) {
    this.dogName = dogName;
    this.ownerName = ownerName;
    this.name = _toonName(dogName, ownerName);
  }
  _toonName(dogName, ownerName) {
    return `${dogName} ${ownerName}`;
  }
}

var cartoon = new DodCartoon('Scooby-Doo', 'Shaggy');

```

```
// good
var name = cartoon.name;
console.log(name);
// "Scooby-Doo Shaggy"

// bad
name = cartoon._toonName(cartoon.dogName, cartoon.ownerName);
console.log(name);
// "Scooby-Doo Shaggy"
```

9. Naming Convention for Global Variables

JavaScript

가 (mutable) camel case , 가 (immutable)

10. Naming Convention for File Names

flower.jpg *Flower.jpg*가 (Apache, Unix)

Flower.jpg *flower.jpg* IIS *Flower.jpg*

가

(lowercase)

Conclusion

가 10가 JavaScript

., [JavaScript](#), [Naming](#), [Conventions](#)

From:

<http://rwiki.repia.com/> -

. - 2023.12

Permanent link:

http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:javascript_naming_conventions&rev=1649287931 

Last update: **2022/04/07 08:32**