



Об'єкти



План

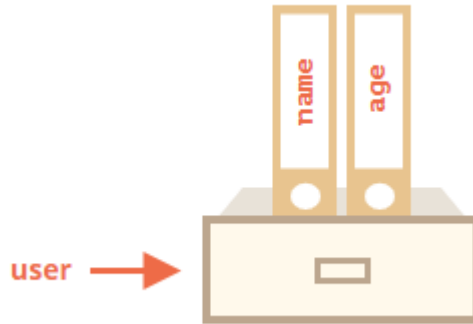
- Оголошення об'єктів
- Перебір об'єктів
- Видалення властивостей
- Прототипи

Оголошення об'єктів

```
var ob = {  
  name: 'Some name',  
  age: 25  
}
```

Доступ до властивостей
`ob.name;` // 'Some name'
`ob['age'];` // 25

Можна передавати властивість через змінну



Робота з властивостями

```
let user = {  
  name: "John",  
  age: 30,  
  "likes birds": true // назва властивості з кількох слів повинна бути в  
лапках  
};
```

Присвоювання значення

```
user.likes birds = true // помилка  
user["likes birds"] = true;
```

Робота з властивостями

```
let user = {  
  name: "John",  
  age: 30  
};
```

Доступ до властивості через змінну

```
let key = "name";  
alert( user[key] ); // John
```

```
let key = "name";  
alert( user.key ); // undefined
```

Передача по силці

При присвоєнні строки, числа чи булевого значення змінній створюється нова комірка пам'яті з цим значенням

```
var message = "Hello!";  
var phrase = message;
```



В змінну в яку присвоєний об'єкт, зберігається не сам об'єкт, а адреса на комірку - "силка"

```
var user = { name: "Bob" }; // в змінній силка  
var admin = user; // скопіювали силку
```



Копіювання об'єкта

```
let user = {  
  name: "John",  
  age: 30  
};
```

```
let clone = {}; // пустий об'єкт
```

```
// скопіюємо всі властивості user в нього
```

```
for (let key in user) {  
  clone[key] = user[key];  
}
```

```
// тепер в змінні clone знаходиться абсолютно незалежний клон об'єкта.
```

```
clone.name = "Pete"; // змінимо дані
```

```
alert( user.name ); // в оригінальному об'єкті значення властивості `name` осталось - John.
```

Копіювання об'єкта

```
Object.assign(dest, [src1, src2, src3...])
```

властивості всіх перерахованих об'єктів, починаючи з другого, копіюються в перший об'єкт. Після копіювання метод повертає об'єкт dest.

```
let user = {
```

```
  name: "John",
```

```
  age: 30
```

```
};
```

```
let clone = Object.assign({}, user);
```

змінні clone знаходиться абсолютно незалежний клон об'єкта.

Копіювання об'єкта

```
let b = {a: 123};  
let a = JSON.stringify(b);  
a = JSON.parse(a);
```

Перевірка чи властивість є в об'єкті

- `"prop" in obj` - `true`, якщо властивість `prop` є в об'єкті `obj`

```
if ("name" in ob) {  
    alert ("Властивість name існує!");  
}
```

Видалення властивості з об'єкта

- `delete ob.age;` - видаляє властивість `age` з об'єкта `ob`

Перебір властивостей

```
let ob = {  
  name: 'Some name',  
  age: 25  
}  
for (let key in ob) {  
  alert( "Ключ: " + key + " значення: " + ob[key] );  
}
```

<https://goo.gl/9gLV3L>

Методи об'єкта

```
let user = {  
  name: "John",  
  age: 30  
};  
  
user.sayHi = function() { // властивістю є функція (метод)  
  alert("Hello!");  
};  
  
user.sayHi(); //Hello!
```

Такий же запис

```
user = {  
  sayHi: function() {  
    alert("Привет");  
  }  
};
```

this в об'єктах

- вказує на поточний об'єкт

```
let person = {  
    firstName: "John",  
    lastName : "Doe",  
    id        : 5566,  
    fullName : function() {  
        return this.firstName + " " + this.lastName;  
    }  
};
```

fullName() - метод об'єкта

person.fullName(); - виклик методу

Практика

- дано об'єкт

```
let skill = {  
  name: 'html',  
  level: 5  
}
```

Написати метод який поверне строку вигляду “html - 5”

this не є фіксованим

Значення this обчислюється під час виконання коду і залежить від контексту.

```
let user = { name: "Джон" };  
let admin = { name: "Админ" };
```

```
function sayHi() {  
    alert( this.name );  
}
```

```
// використовуємо ту ж функцію в двох об'єктах  
user.f = sayHi;  
admin.f = sayHi;
```

```
// виклик функції мають різне значення this  
user.f(); // Джон (this == user)  
admin.f(); // Админ (this == admin)
```


Створення об'єктів через new

Використовуються для створення безліч однотипних об'єктів. Це можна зробити за допомогою функції-конструктора і оператора "new".

```
function User(name) {  
    this.name = name;  
    this.isAdmin = false;  
}  
  
let user = new User("Вася");  
  
alert(user.name); // Вася  
alert(user.isAdmin); // false
```

Методи в конструкторі

```
function User(name) {  
    this.name = name;
```

```
    this.sayHi = function() {  
        alert( "Меня зовут: " + this.name );  
    };  
}
```

```
let vasya = new User("Вася");
```

```
vasya.sayHi(); // Меня зовут: Вася
```

Практика

- Створіть об'єкт `calculator` з методами:
 - `sum ()` повертає суму цих двох значень
 - `mul ()` повертає добуток цих двох значень
 - ініціалізувати об'єкт через функцію-конструктор

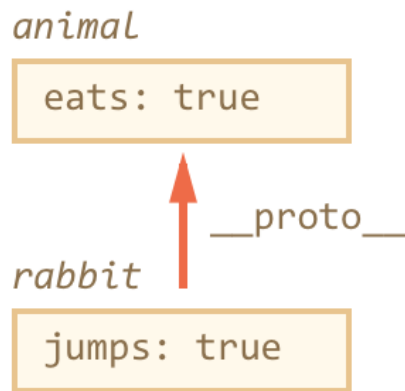
```
let calculator = {  
}
```

```
alert( calculator1.sum() );  
alert( calculator2.mul() );
```

Прототип об'єкта

- Об'єкти в JavaScript можна організувати в ланцюжки так, щоб властивість, що не знайдене в одному об'єкті, автоматично шукалася в іншому
- Сполучною ланкою виступає спеціальна властивість **__proto__**

```
let animal = {  
  eats: true  
};  
let rabbit = {  
  jumps: true  
};  
rabbit.__proto__ = animal;  
alert( rabbit.jumps ); // true  
alert( rabbit.eats ); // true
```



- Об'єкт, на який вказує посилання `__proto__`, називається «**прототипом**». В даному випадку вийшло, що `animal` є прототипом для `rabbit`.
- Також кажуть, що об'єкт `rabbit` «прототипно успадковує» від `animal`.

```
let animal = {  
  eats: true  
};  
let rabbit = {  
  jumps: true,  
  eats: false  
};  
rabbit.__proto__ = animal;  
alert( rabbit.eats ); // false, взято з rabbit
```

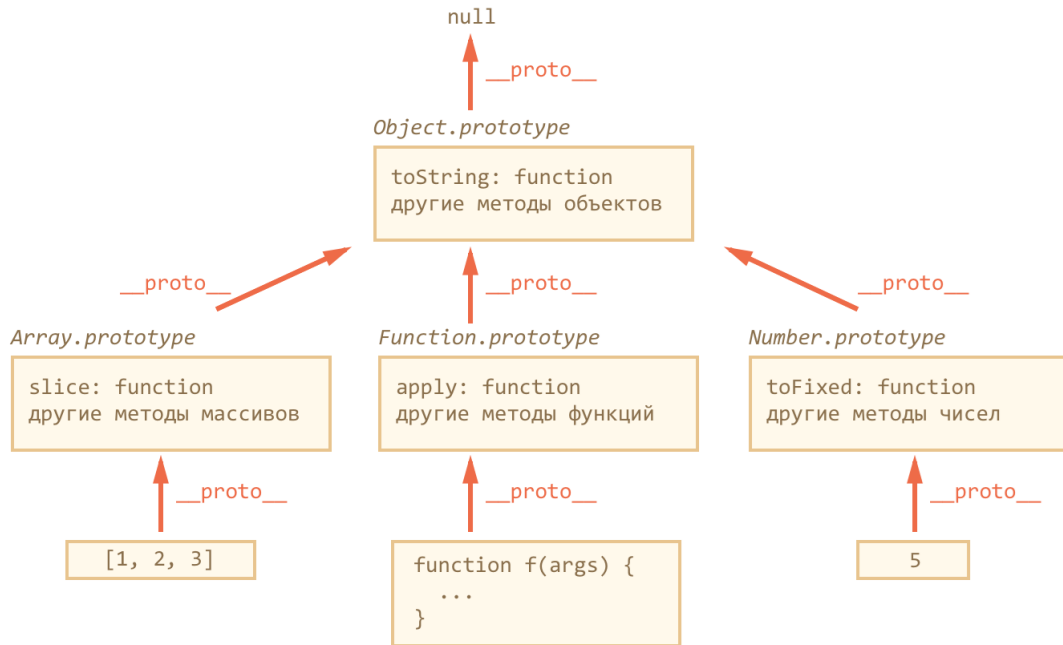
- Прототип - це «резервне сховище властивостей і методів» об'єкта, автоматично використовується при пошуку.

hasOwnProperty

- Звичайний цикл `for..in` не робить різниці між властивостями об'єкта і його прототипу
- Виклик `obj.hasOwnProperty(prop)` повертає `true`, якщо властивість `prop` належить самому об'єкту `obj`, інакше `false`

Вбудовані «класи» в JavaScript

- В JavaScript є вбудовані об'єкти: Date, Array, Object і інші. Вони використовують прототипи і демонструють організацію «псевдокласів» на JavaScript



Посилання

<https://learn.javascript.ru/object>

<https://learn.javascript.ru/object-reference>

<https://learn.javascript.ru/object-for-in>

Відео

<https://www.youtube.com/watch?v=sfqsb6lwx0>

<https://www.youtube.com/watch?v=J1alrZFnGig>