# Тема 10. Объекты. Деструктуризация


хекслет колледж

# Цель занятия:

Сформировать понимание объектов как ключевой структуры данных в JavaScript, научить создавать объекты, работать со свойствами и применять деструктуризацию для удобного доступа к данным.

хекслет колледж

# Учебные вопросы:

1. Понятие объекта в JavaScript. Создание объектов.
2. Свойства и методы объекта. Доступ к свойствам объекта. Изменение и добавление свойств
3. Перебор свойств объекта
4. Деструктуризация объектов

хекслет колледж

# 1. Понятие объекта в JavaScript. Создание объектов.

Объект в JavaScript — это структура данных, которая хранит набор свойств в формате «ключ — значение».

хекслет колледж

В отличие от массива, где данные упорядочены по индексам, в объекте доступ к данным осуществляется по имени свойства.

```
let user = {
    name: "Анна",
    age: 20,
    isStudent: true
};
```

name, age, isStudent — ключи (свойства)
"Анна", 20, true — значения

хекслет колледж

# Когда использовать объекты?

Объекты применяются, когда данные:

- описывают одну сущность (пользователь, товар, настройки);
- имеют разные типы;
- логически связаны между собой.

Примеры:

- пользователь сайта;
- карточка товара;
- настройки интерфейса.

хекслет колледж

# Создание объектов

1. Литерал объекта {} (основной способ)

```
let book = {
    title: "JavaScript Basics",
    pages: 250,
    available: true
};
```

- Самый удобный и читаемый способ
- Используется в большинстве случаев

хекслет колледж

# 1. Понятие

2. Создание пустого объекта

```
let settings = {};
```

Свойства можно добавлять позже:

```
settings.theme = "dark";
settings.lang = "ru";
```

хекслет колледж

# Свойства объекта

Свойство — это пара ключ: значение

- Ключи обычно записываются как строки

Значением может быть любой тип данных:

- число
- строка
- логическое значение
- массив
- другой объект
- функция

хекслет колледж

# Пример:

```
let student = {
    name: "Иван",
    scores: [5, 4, 5],
    address: {
        city: "Алматы",
        street: "Абая"
    }
};
```

хекслет колледж

# Именование свойств

Имя свойства:

- начинается с буквы, $ или _
- может содержать цифры (не в начале)
- Если имя содержит пробелы или спецсимволы, его нужно заключать в кавычки:

```
let user = {
    "first name": "Олег"
};
```

хекслет колледж

# Вывод:

- Объект — это набор связанных данных в формате «ключ — значение».
- Используется для описания реальных сущностей.
- Основной способ создания — литерал {}.
- Свойства объекта могут хранить значения любых типов.
- Объекты — одна из ключевых структур данных в JavaScript.

хекслет колледж

# 2. Свойства и методы объекта. Доступ к свойствам объекта. Изменение и добавление свойств

Свойство объекта — это пара
ключ : значение.

```
let user = {
    name: "Анна",
    age: 20
};
```

name, age — свойства
"Анна", 20 — значения свойств

хекслет колледж

# Методы объекта

Метод — это свойство, значением которого является функция.

```
let user = {
    name: "Анна",
    age: 20,
    greet: function () {
        console.log("Привет!");
    }
};
```

Вызов метода:
user.greet();

хекслет колледж

# Ключевое слово this

**this** указывает на текущий объект, внутри которого вызывается метод.

```
let user = {
    name: "Анна",
    greet: function () {
        console.log("Привет, " + this.name);
    }
};
```

хекслет колледж

# Доступ к свойствам объекта

Существует два способа доступа к свойствам.

**1. Точечная нотация (основная)**

```
console.log(user.name);
user.age = 21;
```

Используется, когда имя свойства известно заранее.

хекслет колледж

## 2. Доступ через квадратные скобки

```
console.log(user["name"]);
```

Используется, когда:

- имя свойства хранится в переменной;
- имя содержит пробелы или специальные символы.

```
let key = "age";
console.log(user[key]);
```

хекслет колледж

# Изменение свойств объекта

Существующее свойство можно изменить:

```
let user = {
    name: "Анна",
    age: 21
};

user.age = 22;
```

хекслет колледж

# Добавление новых свойств

Если свойства не было, оно будет создано автоматически:

```
let user = {
    name: "Анна",
    age: 21
};

user.city = "Алматы";
```

хекслет колледж

# Удаление свойств

Для удаления используется оператор delete:

```
let user = {
    name: "Анна",
    city: "Алматы"
};

delete user.city;
```

хекслет колледж

# Вывод:

- Объект состоит из свойств и методов.
- Метод — это функция, принадлежащая объекту.
- Доступ к свойствам возможен через точку и квадратные скобки.
- Свойства можно изменять, добавлять и удалять.
- this позволяет обращаться к данным текущего объекта.

хекслет колледж

# 3. Перебор свойств объекта

Перебор свойств объекта используется, когда необходимо просмотреть или обработать все данные, хранящиеся в объекте.

хекслет колледж

# Цикл for...in

Основной способ перебора свойств объекта.

```
let user = {
    name: "Анна",
    age: 20,
    city: "Москва"
};

for (let key in user) {
    console.log(key, user[key]);
}
```

key — имя свойства (ключ)
user[key] — значение свойства

хекслет колледж

Почему используется квадратная нотация? При переборе имя свойства хранится в переменной (key), поэтому доступ через точку невозможен:

```
user.key      // неправильно
user[key]     // правильно
```

хекслет колледж

# Что перебирает for...in?

- перебирает перечисляемые свойства объекта
- порядок перебора не гарантирован
- чаще используется для объектов, а не массивов

хекслет колледж

# Object.keys() — перебор ключей

```
let keys = Object.keys(user);
console.log(keys);
```

Результат:

```
["name", "age", "city"]
```

Использование с циклом:

```
keys.forEach(key => {
    console.log(key, user[key]);
});
```

хекслет колледж

# Object.values() — перебор значений

```
let values = Object.values(user);
console.log(values);
```

Результат:

```
["Анна", 20, "Москва"]
```

хекслет колледж

# Object.entries() — ключи и значения

```
let entries = Object.entries(user);
```

Результат:

```
[
  ["name", "Анна"],
  ["age", 20],
  ["city", "Москва"]
]
```

Использование:

```
entries.forEach(([key, value]) => {
    console.log(key, value);
});
```

хекслет колледж

# Сравнение способов перебора:

| Способ | Что возвращает | Когда использовать |
|---|---|---|
| for...in | ключи | быстрый перебор объекта |
| Object.keys() | массив ключей | когда нужны только ключи |
| Object.values() | массив значений | когда важны значения |
| Object.entries() | массив пар | для полной обработки данных |

хекслет колледж

# Вывод:

Объекты можно перебрать несколькими способами.
for...in — базовый цикл для перебора свойств.
Методы Object.keys, Object.values, Object.entries возвращают массивы.
Для доступа к значениям при переборе используется квадратная нотация.
Выбор метода зависит от поставленной задачи.

хекслет колледж

# 4. Деструктуризация объектов

Деструктуризация объекта — это удобный способ извлечь значения свойств объекта в отдельные переменные.

хекслет колледж

Вместо многократного обращения к объекту:

```
let user = {
    name: "Арина",
    age: 20,
    city: "Алматы"
};

let name = user.name;
let age = user.age;
```

используется деструктуризация:

```
let { name, age } = user;
```

хекслет колледж

# Синтаксис деструктуризации:

Синтаксис деструктуризации

**let { свойство1, свойство2 } = объект;**

- Имена переменных должны совпадать с именами свойств объекта.

хекслет колледж

# Пример:

```
let car = {
    brand: "Toyota",
    year: 2020,
    color: "red"
};

let { brand, color } = car;

console.log(brand); // Toyota
console.log(color); // red
```

- Порядок не имеет значения

хекслет колледж

# Переименование переменных:

Можно задать другое имя переменной:

```
let { brand: carBrand, year: carYear } = car;

console.log(carBrand); // Toyota
console.log(carYear);  // 2020
```

хекслет колледж

# Значения по умолчанию

Если свойства нет, можно задать значение по умолчанию:

```
let { model = "unknown" } = car;

console.log(model); // unknown
```

хекслет колледж

# Деструктуризация параметров функции

Часто используется при передаче объекта в функцию:

```
let car = {
    brand: "Toyota",
    year: 2020,
    color: "red"
};

function printCar({ brand, year }) {
    console.log(brand, year);
}

printCar(car);
```

хекслет колледж

Важно помнить:

- Деструктуризация не копирует объект, а только извлекает значения.
- Если свойства не существует и нет значения по умолчанию, переменная будет undefined.

хекслет колледж

# Вывод:

- Деструктуризация упрощает работу с объектами.
- Извлечение происходит по именам свойств.
- Поддерживается переименование и значения по умолчанию.
- Часто применяется в функциях и при работе с результатами API (в дальнейшем).

хекслет колледж

# Контрольные вопросы:

- Что такое объект в JavaScript?
- Чем объект отличается от массива?
- Из каких элементов состоит объект?
- Какие типы данных могут быть значениями свойств объекта?
- Можно ли создавать вложенные объекты?
- Что называется свойством объекта?
- Что называется методом объекта?
- В чём разница между доступом к свойствам через . и через []?
- В каком случае обязательно использовать доступ через []?
- Как изменить значение существующего свойства объекта?
- Как добавить новое свойство объекту?
- Как работает цикл for...in?
- Для чего используются методы Object.keys(), Object.values() и Object.entries()?
- Что такое деструктуризация объекта?

хекслет колледж

# Домашнее задание:

https://ru.hexlet.io/courses/js-basics

хекслет колледж

хекслет колледж

@HEXLY.KZ