

# ОБЪЕКТЫ

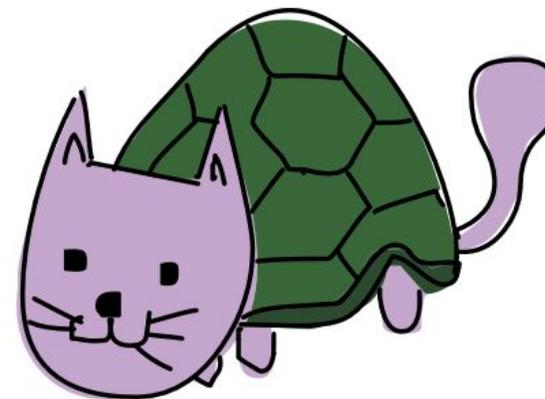
# ОБЪЕКТЫ

- Объекты JavaScript очень похожи на массивы, но для доступа к элементам объектов используются строки, а не числа
- Эти строки называют *ключами*, или *свойствами*, а элементы, которые им соответствуют, — *значениями*
- Вместе эти фрагменты информации образуют пары «ключ-значение»
- Причем если массивы используются главным образом как списки, хранящие множество элементов, то объекты часто применяют как одиночные сущности с множеством характеристик, или атрибутов

# Создание объектов

- Для хранения всевозможной информации об одном животном подойдет JavaScript-объект
- Вот пример объекта, где хранятся сведения о трехногой кошке по имени Гармония

```
var cat = {  
  "legs": 4,  
  "name": "Гармония",  
  "color": "Черепеховый"  
};
```



# Создание объектов

- Мы создали переменную под названием `cat` и присвоили ей объект с тремя парами «ключ-значение» (лапы, имя, окрас)
- При создании объекта используются фигурные скобки `{ }` вместо квадратных, к которым мы привыкли, создавая массивы

```
{ "key1": 99 }
```

Ключ, обязательно  
строковый

Значение  
любого типа

# Ключи без кавычек

- Создавая первый объект, мы писали имена ключей в кавычках, однако это не обязательно
- Следующая запись тоже является допустимой:

```
var cat = {  
  legs: 4,  
  name: "Гармония",  
  color: "Черепеховый"  
};
```

# Ключи без кавычек

- JavaScript знает, что ключи всегда строковые, поэтому можно обходиться без кавычек
- В этом случае имена ключей должны соответствовать тем же правилам, что и имена переменных: например, в них не должно быть пробелов
- Но если ключ указан в кавычках, пробелы в его имени допустимы:

```
var cat = {  
  legs: 4,  
  "full name": "Гармония Филомена Уси-Пусечка Морган",  
  color: "Черепеховый"  
};
```

# Ключи без кавычек



- Помните, что, хотя ключ всегда является строковым (в кавычках он записан или без), значение, соответствующее этому ключу, может быть любого типа — даже переменной, в которой хранятся данные
- Кроме того, весь объект можно записать одной строкой, хотя читать такую программу будет, пожалуй, не слишком удобно:

```
var cat = { legs: 4, name: "Гармония", color: "Черепеховый" };
```

# Доступ к значениям внутри объектов

- Хранящиеся в объектах значения можно получить с помощью квадратных скобок — так же, как элементы массива
- Единственное различие в том, что вместо индекса (число) используется ключ (строка)

```
cat["name"];
```

```
"Гармония"
```

# Доступ к значениям внутри объектов

- Точно так же, как необязательны кавычки при записи литерала объекта, их можно опускать и при доступе к значениям по ключу
- Однако в этом случае код будет немного другим:

```
cat.name;
```

```
"Гармония"
```

---

## Что такое класс

Для определения классов существует несколько вариантов. Самый простой способ описания классов - это пользовательский тип данных.

По другому же - класс, некая конструкция, шаблон, по которому создаются объекты.



## Объект

Мы уже работали с объектами, поэтому вы имеете понятие об объектах.

Объекты - это экземпляры класса, то есть представители класса.

---

## Классы и объекты

### Класс Кошка:

- 4 лапы
- хвост
- усы
- мурчит

### Объект МояКошка

- Имеет конкретное имя
- конкретный цвет
- уникальные особенности

---

## Конструктор

Конструктор - функция, запускающаяся в момент создания объекта. Обычно используется для инициализации свойств объекта.

Конструктор можно обнаружить по ключевому слову `new`

---

## Описание класса

```
class myClass {  
    constructor(){  
  
    }  
}
```

# Задание

- Предположим, у вас список любимых фильмов.
- Правда было бы здорово хранить информацию об этих фильмах на компьютере, чтобы в случае чего быстро найти сведения о том или ином фильме?
- Для этого можно создать объект, каждый ключ в котором — это название фильма, а каждое значение — другой объект, в котором содержится информация об этом фильме.
- Да, хранящиеся в объекте значения тоже могут быть объектами!

# Задание

```
var movies = {  
  "В поисках Немо": {  
    releaseDate: 2003,  
    duration: 100,  
    actors: ["Альберт Брукс", "Эллен Дедженерес",  
            "Александр Гоулд"]  
  },  
  "Звездные войны: Эпизод VI – Возвращение джедая": {  
    releaseDate: 1983,  
    duration: 134,  
    actors: ["Марк Хэмилл", "Харрисон Форд", "Кэрри Фишер"]  
  },  
  "Гарри Поттер и Кубок огня": {  
    releaseDate: 2005,  
    duration: 157,  
    actors: ["Дэниел Рэдклифф", "Эмма Уотсон", "Руперт Гринт"]  
  }  
};
```



# Домашнее задание

Представьте, что вы играете в какую-нибудь игру со своими друзьями и вам нужно вести счет.

Создайте для этого объект и назовите его `scores`.

Пусть ключами будут имена ваших друзей, а значениями — набранные ими очки (0 или больше).

Счет игроков надо будет увеличивать по мере того, как они зарабатывают новые очки.