



Лекция 4

Объекты

Цели на сегодня

- Понять, что такое объект и узнать области использования объектов;
- Научиться отличать методы и свойства объектов;
- Узнать, как получать, добавлять, изменять и удалять значения;
- Разобрать написание небольшой консольной игры.

Типы данных

1. Прimitives

- a. Числа (Numbers);
- b. Строки (Strings);
- c. Логические (Boolean)
- d. Символ (Symbol);
- e. null;
- f. undefined.

2. Сложные

- a. Объекты (Objects);
- b. Массивы (Arrays);
- c. Функции (Functions).

NB!

```
console.log(typeof null) // object
```

```
console.log(typeof {name: 'Peter'}) // object
```

Объект --

это сложный тип данных, который способен хранить любые данные в формате "ключ - значение". При этом по ключу всегда можно получить доступ к значению.

Как создать объект?

Для создания объектов можно использовать два основных способа:

- 1) Фигурные скобки (литеральная нотация);**
- 2) Функцию-конструктор.

objects.js x

```
1 // Создаем пустой объект
2
3 const obj = {};
4 console.info(obj) // Object {  }
5
6
7 // Второй синтаксис менее лаконичен
8
9 const secondObj = new Object();
10 console.info(secondObj); // Object {  }
11
```

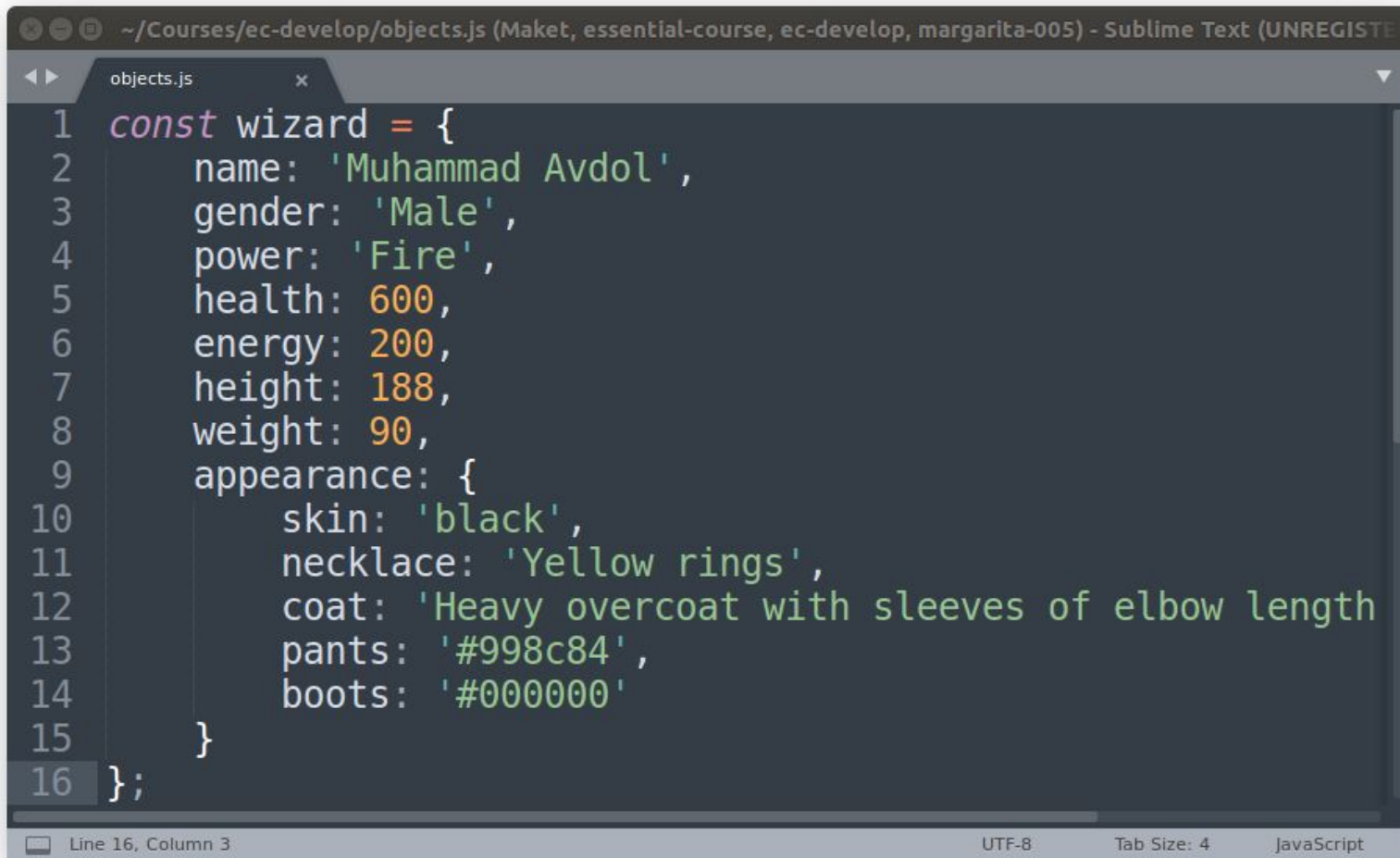


Первый способ

```
~/Courses/ec-develop/objects.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGISTE
objects.js x
1  const wizard = {};
2
3  wizard.name = 'Muhammad Avdol',
4  wizard.gender = 'Male',
5  wizard.power = 'Fire',
6  wizard.health = 600,
7  wizard.energy = 200,
8  wizard.height = 188,
9  wizard.weight = 90,
10 wizard.appearance = {
11     skin: 'black',
12     necklace: 'Yellow rings',
13     coat: 'Heavy overcoat with sleeves of elbow length and
14     pants: '#998c84',
15     boots: '#000000'
16 }
```

Line 7, Column 19 UTF-8 Tab Size: 4 JavaScript

Второй способ



```
~/Courses/ec-develop/objects.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGISTE
objects.js x
1  const wizard = {
2      name: 'Muhammad Avdol',
3      gender: 'Male',
4      power: 'Fire',
5      health: 600,
6      energy: 200,
7      height: 188,
8      weight: 90,
9      appearance: {
10         skin: 'black',
11         necklace: 'Yellow rings',
12         coat: 'Heavy overcoat with sleeves of elbow length
13         pants: '#998c84',
14         boots: '#000000'
15     }
16 };
```

Line 16, Column 3 UTF-8 Tab Size: 4 JavaScript

Третий способ



```
~/Courses/ec-develop/objects.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGISTERED)
objects.js x
1  const name = 'Muhammad Avdol';
2  const gender = 'Male';
3  const power = 'Fire';
4  const health = 600;
5  const energy = 200;
6
7  const wizard = {
8      name,
9      gender,
10     power,
11     health,
12     energy
13 };
14
```

Line 13, Column 3 UTF-8 Tab Size: 4 JavaScript

Деструктуризация

ES6

```
~/Courses/ec-develop/objects.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGISTE
objects.js x
1  const wizard = {
2      name: 'Muhammad Avdol',
3      gender: 'Male',
4      power: 'Fire',
5      health: 600,
6      energy: 200,
7  }
8
9  const {name, gender, power, health, energy} = wizard;
10
11  // Имя теперь можно получить из свойства объекта:
12  console.log(wizard.name); // "Muhammad Avdol"
13
14  // И из переменной name:
15  console.log(name); // "Muhammad Avdol"
```

Line 15, Column 39 UTF-8 Tab Size: 4 JavaScript

Функции-конструкторы используются, когда нужно создать много похожих объектов.

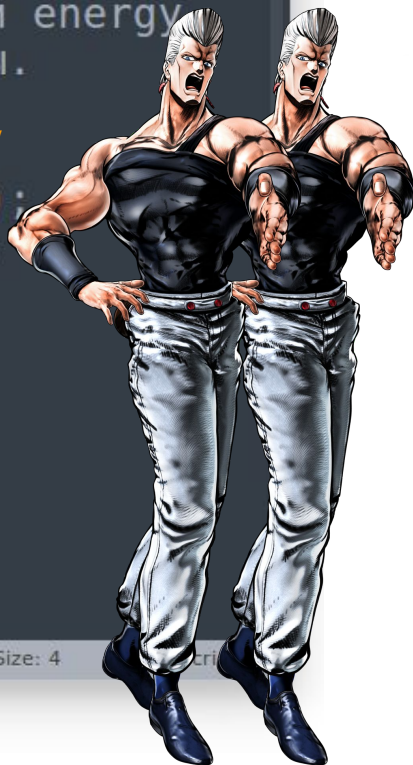
В таком случае создается функция-конструктор, а объекты из нее образуют с помощью ключевого слова *new*.

objects.js

```
1 function Knight(id, health, energy) {  
2     this.id = id;  
3     this.health = health;  
4     this.energy = energy;  
5     this.name = 'Jean Pierre Polnareff';  
6     this.gender = 'Male';  
7     this.power = 'Rapier master';  
8     this.height = 185;  
9     this.weight = 78;  
10    this.clothes = {  
11        necklace: false,  
12        coat: 'Fitted sleeveless black top',  
13        pants: 'Yellow ones',  
14        boots: 'Ordinary black boots'  
15    };  
16 }
```

objects.js

```
18  /*
19     Каждое присваивание переменной создает нового
20     рыцаря, которому мы передаем id, health и energy.
21     Остальные параметры у них будут одинаковы.
22  */
23                                     id, health, energy
24  const firstKnight = new Knight(001, 200, 120);
25  const secondKnight = new Knight(002, 60, 10);
```



Самостоятельная работа

Создать объект ***room*** (используя литеральную нотацию и функцию-конструктор), описывающий комнату вокруг вас:

{

- 1) Стены: { количество, цвет в HEX },
- 2) Окна: { количество, цвет рамы, чистые или нет? },
- 3) Освещение: { количество ламп, свет включен или нет? },
- 4) Проектор: включен или нет?,
- 5) Столы: количество,
- 6) Люди: количество душ

}

Методы

Метод --

это свойство-функция в объекте.

Ключевое слово ***this*** (оно еще называется контекстом вызова) ссылается на текущий объект или функцию и используется для доступа к текущему объекту из метода.



objects.js



thegame.js



```
1  const obj = {  
2      two: 2,  
3      addTwo: function(num) {  
4          return num + this.two;  
5      }  
6  }  
7  
8  obj.addTwo(10) // 12  
9  obj.addTwo(5) // 7
```



Line 9, Column 19

UTF-8

Tab Size: 4

JavaScript

ES6

```
~/Courses/ec-develop/objects.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGISTERED)
objects.js x thegame.js x
1  /*
2      В ES-2015 можно использовать сокращенную запись
3      методов.
4
5      Работает так же, но записывается лаконичнее.
6  */
7
8  const obj = {
9      two: 2,
10     addTwo(num) {
11         return num + this.two;
12     }
13 }
14
15 obj.addTwo(10) // 12
16 obj.addTwo(5) // 7
```

Line 5, Column 49 UTF-8 Tab Size: 4 JavaScript

```
~/Courses/ec-develop/thegame.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGIST
thegame.js x
13 pants: #998C84 ,
14 boots: '#000000'
15 },
16 beatSomeone: function(person) {
17   if (person.name === 'Muhammad Avdol') {
18     return 'No!';
19   } else {
20     person.health -= 50;
21     person.energy -= 10;
22   }
23   if (person.health <= 0) {
24     return 'Person died'
25   }
26   if (person.energy <= 0) {
27     return 'Person can\'t fight'
28   }
29   return `${person.name} (${person.id}) lost 50 HP points
30 }
31 ,
```

Line 18, Column 22

UTF-8 Tab Size: 4 JavaScript



```
~/Courses/ec-develop/thegame.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGIS...
thegame.js x
50     this.beatSomeone = function(person) {
51         if (person.name === 'Jean Pierre Polnareff') {
52             return 'Nani?';
53         } else {
54             person.health -= 20;
55             person.energy -= 30;
56
57             if (person.health <= 0) {
58                 return 'Person died'
59             }
60
61             if (person.energy <= 0) {
62                 return 'Person can\'t fight'
63             }
64
65             return `${person.name} lost 20 HP points and 30 energy po
66         }
67     };
68 }
```

Line 50, Column 42

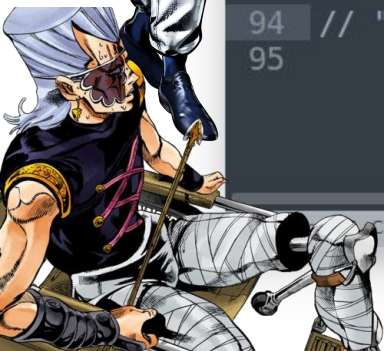
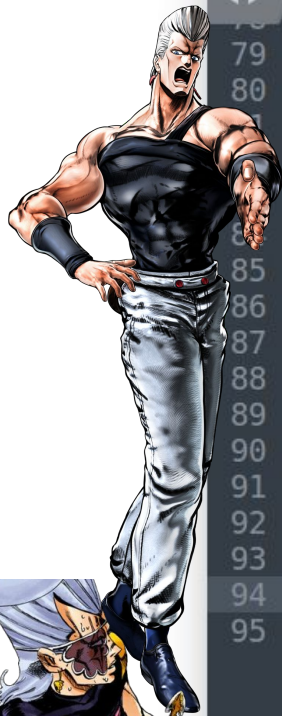
UTF-8 Spaces: 4 JavaScript



**Как получить значение свойства /
вызвать метод?**


```
~/Courses/ec-develop/thegame.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGIS...  
thegame.js x  
79 /*  
80 Получить свойство: названиеОбъекта.названиеСвойства  
81 Получить метод: названиеОбъекта.названиеМетода  
82 Вызвать метод: названиеОбъекта.названиеМетода()  
83 */  
84  
85 console.info(wizard.health); // 600  
86 console.info(firstKnight.energy); // 120  
87  
88 wizard.beatSomeone(secondKnight); // "Person can't fight"  
89  
90 firstKnight.beatSomeone(wizard);  
91 // "Muhammad Avdol lost 20 HP points and 30 energy points"  
92  
93 wizard.beatSomeone(firstKnight);  
94 // "Jean Pierre Polnareff (1) lost 50 HP points and 10 energy points"  
95
```

Column 70 UTF-8 Spaces: 4 JavaScript



Как изменять значения?

```
~/Courses/ec-develop/thegame.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGIST
thegame.js x
86  /*
87     Для уверенности в своей победе Рыцарь (2) решил
88     увеличить запас здоровья Рыцаря (1).
89
90     Для этого он получил значение здоровья первого
91     рыцаря и переписал значение на большее
92  */
93
94  // Столько было
95  console.info(firstKnight.health); // 60
96
97  firstKnight.health = 300;
98
99  // Столько стало
100 console.info(firstKnight.health); // 300
101
```

Line 100, Column 41 UTF-8 Spaces: 4 JavaScript

Как удалять свойства и методы?

```
~/Courses/ec-develop/thegame.js (Maket, essential-course, ec-develop, margarita-005) - Sublime Text (UNREGIS
thegame.js x
104  /*
105     Маг оказался хитрее, поэтому решил запретить
106     рыцарям сопротивляться, удалив у них метод
107     beatSomeone
108  */
109
110  delete firstKnight.beatSomeone; // true
111  delete secondKnight.beatSomeone; // true
112
113  // При попытке ударить мага, появится ошибка:
114
115  firstKnight.beatSomeone(wizard);
116  // TypeError: firstKnight.beatSomeone is not a function
117  secondKnight.beatSomeone(wizard);
118  // TypeError: secondKnight.beatSomeone is not a function
```

Line 118, Column 57 UTF-8 Spaces: 4 JavaScript

Самостоятельная работа

Написать объект `R2D2`, в котором будут следующие методы:

- 1) Метод для сложения двух чисел (`R2D2.add(a, b)`);
- 2) Метод для нахождения большего из двух чисел (`R2D2.max(a, b)`);
- 3) Метод для возведения в степень (`R2D2.pow(число, степень)`);
- 4) Метод для нахождения дискриминанта (`R2D2.disk(a, b, c)`);
- 5) Метод для нахождения диаметра (`R2D2.diam(radius)`);

При выполнении домашнего задания нельзя использовать готовые методы объекта `Math`

- + Додавання властивостей через [] + ES6
- + Деструктуризація.