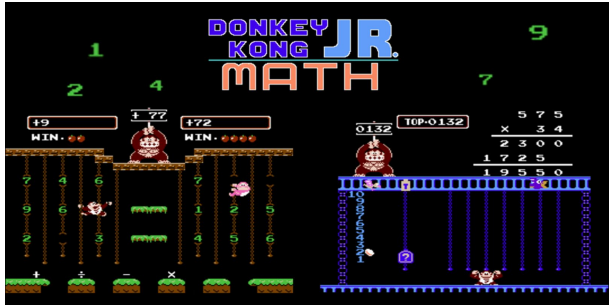
JavaScript

Введения и основные понятия

Что такое JavaScript?

JavaScript язык программирования, который позволяет Вам создать динамически обновляемый контент, управляет мультимедиа, анимирует изображения, впрочем, делает всё, что угодно. Окей, не все, что угодно, но все равно, это удивительно, что можно достичь с помощью нескольких строк JavaScript кода.

```
10
11  xhr.open('POST', loginUrl, true);
12  xhr.setRequestHeader('Content-Type', 'application/json; charset=UTF-8');
13  xhr.addEventListener('load', function() {
14      var responseObject = JSON.parse(this.response);
15      console.log(responseObject);
16      if (responseObject.token) {
17          tokenElement.innerHTML = responseObject.token;
18      } else {
19          tokenElement.innerHTML = "No token received";
20      }
21  });
22
23  var sendObject = JSON.stringify({name: user, password: password});
24
25  console.log('going to send', sendObject);
26
27  xhr.send(sendObject);
```




```
// Run time
Math.pow(99,99) === 99**99 // FALSE !

// Compile time
;(function (num = 99) {
  return (num ** 99) === Math.pow(99,99) // TRUE !
})()
```

Подключение JavaScript

```
<!DOCTYPE html>
<html>
    <head>
        <title>Заголовок страницы</title>
    </head>
    <body>
        <script>
            console.log("Привет  Вася");
        </script>
    </body>
</html>
```

Подключение JavaScript

```
<!DOCTYPE html>
<html>
    <head>
        <title>Заголовок страницы</title>
    </head>
    <body>
        <script src="scripts.js"></script>
    </body>
</html>
```

Переменные в JavaScript

В JavaScript можно объявлять переменные для хранения данных. Существует три типа переменных **var, let, const**

```
var lesson = "JavaScript"; // Создание переменной и присвоение ей значения

lesson = "JavaScript Уровень 1"; // Переопределение переменной

var TITLE_COLOR = 32;

let day = "16 Апреля"; // Такая же переменная, но область видимости отличается

const FOOTER_COLOR = "#cccccc"; // Константа
```

Имена переменных

На имя переменной в JavaScript наложены всего три ограничения.

- Имя может состоять из: букв, цифр, символов $ и _ ;
- Нельзя использовать символы: %, ^, & и тому подобные;
- Первый символ не должен быть цифрой.

```
var firstName = ""; // можно
var day100 = ""; // можно
var $ = ""; // можно
var _ = ""; // можно
```

```
var first-name = ""; // нельзя
var 100day = ""; // нельзя
```

Имена переменных

Регистр букв имеет значение

Переменные **firstname** и **firstName** – две разные переменные.

Русские буквы допустимы, но не рекомендуются

```
var первое_имя = "";
```

Зарезервированные имена

Существует список слов которыми нельзя называть переменные **var, class, return, export** и другие.

Типы данных

Число / number

Тип **number** используется как для целых, так и для дробных чисел. Существуют специальные числовые значения **Infinity** (бесконечность) и **NaN** (ошибка вычислений).

```
var count = 58;

count = 58.678;

console.log( 1 / 0 ); // Infinity

console.log( "текст" * 2 ); // NaN
```

Типы данных

Строка / string

В JavaScript можно использовать одинарные или двойные кавычки для строк.

```
var txt = "Текст в двойных кавычках";

txt = 'Текст в одинарных кавычках';
```

Логический / boolean

Такой тип используется для хранения значения типа да/нет или ложь/правда.

```
var hasFocus = true;

hasFocus = false;
```

```
var hasFocus = 1;

hasFocus = 0;
```

Типы данных

Специальное значение null

Это специальное значение, которое образует свой собственный тип и имеет смысл «ничего» или «значение неизвестно».

```
var cars = null;
```

Специальное значение undefined

Значение **undefined** так же образует свой собственный тип, состоящий из одного значения. Оно имеет смысл «значение не присвоено».

```
var cars;

console.log(cars);  // выведет undefined
```

Типы данных

Объект / object

Используется для коллекций данных и инструкции по работе с ними. Объявляются объекты при помощи фигурных скобок **{...}**

```
var car = { name: "LADA" };
```

Операторы

умножение * `var res = 45 * 2;`

деление / `var res = 29 / 2;`

сложение + `var res = 78 + 34;` `var text = "Мой " + "дом"; // "Мой дом"`

вычитание - `var res = 78 - 34;`

присваивание = `var number = 34;`

остаток от деления % `5 % 2; // 1 остаток от деления 5 на 2;`

инкремент ++ `i++; // i = i + 1;`

декремент -- `i--; // i = i - 1;`

унарный плюс + `+2; // число 2` `var textToNumber = +"20" + +"5"; // число 25`

унарный минус - `var x = 5; // число 5` `var x = -x; // число -5`

Условные операторы

if/else

```
if (условие) {
   выражение
}

if (условие) {
   выражение
} else {
   выражение 2
}
```

else if

```
if (условие) {
   выражение
} else if (условие 2) {
   выражение 2
} else if (условие 3) {
   выражение 3
} else {
   выражение 4
}
```

switch

```
switch (n) {
  case 1:
      //выполнить Б1
      break;
  case 2:
      //выполнить Б2
      break;
  default
      //выполнить Б2
      break;
}
```

Тернарный оператор

Тернарный оператор используется как сокращение оператора if … else.

(условие) ? выражение когда true : выражение когда false

```
var message = 'Привет ' + (isUser ? userName : 'друг');


var message = 'Привет ';
if (isUser) {
        message += userName;
} else {
        message += 'друг';
}
```

Массивы

Массив - это упорядоченная коллекция значений. Значения в массиве называются элементами, и каждый элемент характеризуется числовой позицией в массиве, которая называется индексом.

```
var empty = [];                                  // Создание пустого массива
var numbers = [2, 3, 5, 7, 11];                  // Массив с пятью числовыми элементами
var misc = [ 1.1, true, "a", ];                  // 3 элемента разных типов + завершающая запятая

var base = 1024;
var table = [base, base+1, base+2, base+3];      // Массив с переменными

var arrObj = [
        [1, {x:1, y:2}],
        [2, {x:3, y:4}]
];              // 2 массива внутри, содержащие объекты
```

Функции

Пример функции:

```
function showMessage() {
        console.log('Привет');
}
```

```
showMessage();            // Привет
showMessage('Вася');  //  Привет Вася
```

Пример функции с параметром

```
function showMessage(name) {
        console.log('Привет' + name);
}
```

Циклы

Цикл while

```
while(выражение) {
        инструкция;
}
```

Цикл do/while

```
do {
        инструкция;
}
while(выражение);
```

Цикл for

```
for (инициализация; проверка; инкремент;) {
        инструкция;
}
```

Цикл for/in

```
for (переменная in object) {
        инструкция;
}
```

Взаимодействие с пользователем

alert(message) - выводит на экран модальное окно с сообщением и приостанавливает выполнение скрипта, пока пользователь не нажмет **ok**.

```
alert ('Привет Вася!');
```

Взаимодействие с пользователем

prompt(title, default) - выводит сообщение и ждет, пока пользователь введет текст.

Возвращает введенное значение или null, если ввод отменён.

title - добавляет заголовок окну.

default - добавляет в поле ввода текст по умолчанию.

Пользователь должен либо что-то ввести и нажать OK, либо отменить ввод кликом на CANCEL или нажатием Esc на клавиатуре.

```
var days = prompt('Сколько дней у вас отпуск?', 21);
alert('Количество дней ' + days + ' записано!');
```

Взаимодействие с пользователем

confirm(text) - выводит сообщение и ждет, пока пользователь нажмет **ok** или **cancel** и возвращает true/false.

```
var agree = confirm("Вы согласны на обработку данных?");

alert( agree );
```

Объекты

Сложный тип данных который имеет набор свойств, и каждое свойство состоит из имени и значения, ассоциированного с этим именем.

Объекты в JavaScript, как и во многих других языках программирования, похожи на объекты реальной жизни. Концепцию объектов JavaScript легче понять, проводя параллели с реально существующими в жизни объектами.

Реальный объект из жизни

Как мы можем описать машину?

- марка
- цвет
- тип
- максимальная скорость
- размер бака для бензина
- новая или нет
- количество дверей

Создание объекта

Создается с помощью фигурных скобок.

```
var car = {}
```

Элементы объекта представляют собой набор пар ключ-значение, которые отделяются между собой двоеточием. Пары отделяются запятыми.

```
var car = {
  name: "LADA",
  color: "red"
};
```

Доступ к элементам объекта возможен через точку или квадратные скобки.

```
console.log(car.name, car["color"]);
```

Создание объектов через "new"

Мы только что рассмотрели как создать одиночный объект, но зачастую нужно создать много однотипных объектов. Это делается в два шага.

Первый. Необходимо описать функцию конструктор.

```
function Car(name, color, year) {
  this.name = name;
  this.color = color;
  this.year = year;
}
```

Создание объектов через "new"

Второй. Создайте экземпляр объекта с помощью ключевого слова **new.**

```
var car1 = new Car("Nissan", "green", 1992);

var car2 = new Car("Mazda", "red", 1990);
```

Эта инструкция создает объекты типа Car и присваивает определенные значения их свойствам.

Мы можем добавить или изменить свойство объекта в любое время

```
car1.color = "black";

car2.maxSpeed = 180;
```

Определение методов в объекте

Метод - это свойство объекта, являющееся функцией. Методы определяются так же, как и обычные функции, за тем исключением, что они присваиваются свойству объекта.

```
function Car(name, color, year) {
  // определение свойств
  displayCarName: function(params) {
  }
}
var car1 = new Car("Nissan", "green", 1992);
car1.displayCarName();
```

Формат JSON

JSON - это текстовый формат данных который используют при взаимодействии с JavaScript. Отличный пример это общение клиентского приложения и сервера. Мы можем получить данные в JSON и так же передать.

```
{
  "model": "lada",
  "color": "red",
  "price": 12000,
  "condition":  "new"
}
```

JSON.parse – используйте если хотите получить доступ к данным JSON.
JSON.stringify – превращает объекты в строку в формате JSON, используется, когда нужно из JavaScript передать данные по сети.

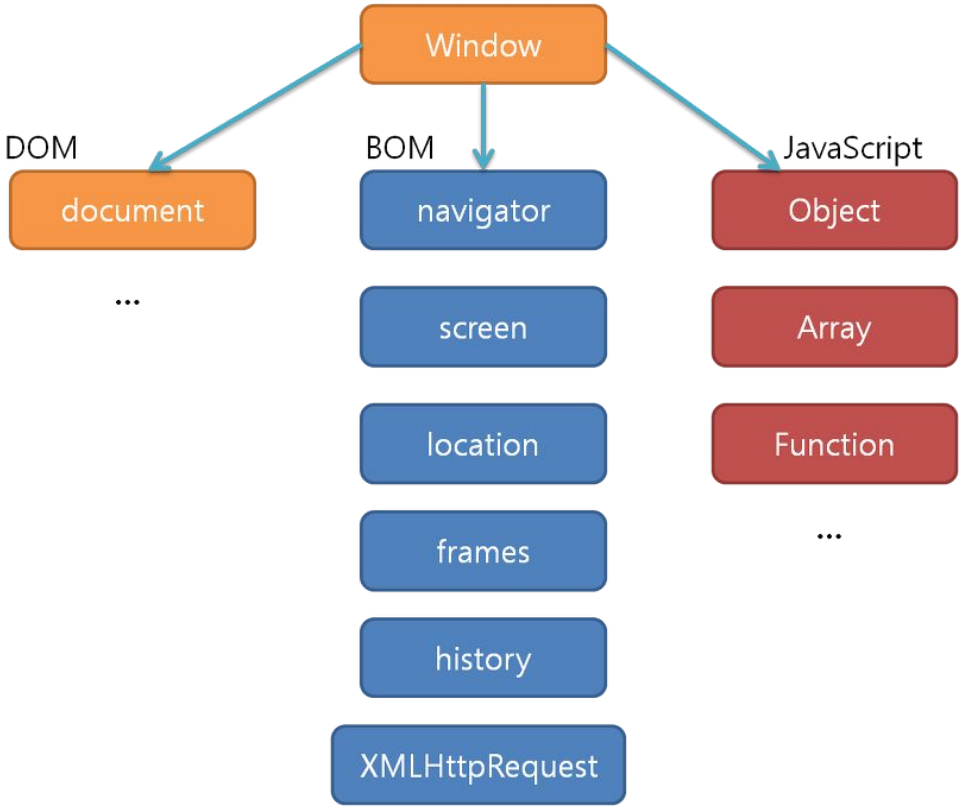
Пример JSON.parse

```
var json = '{"result":true, "count":42}';
var obj = JSON.parse(json);

console.log(obj.count);
// выведет: 42

console.log(obj["result"]);
// выведет: true
```

Пример JSON.stringify

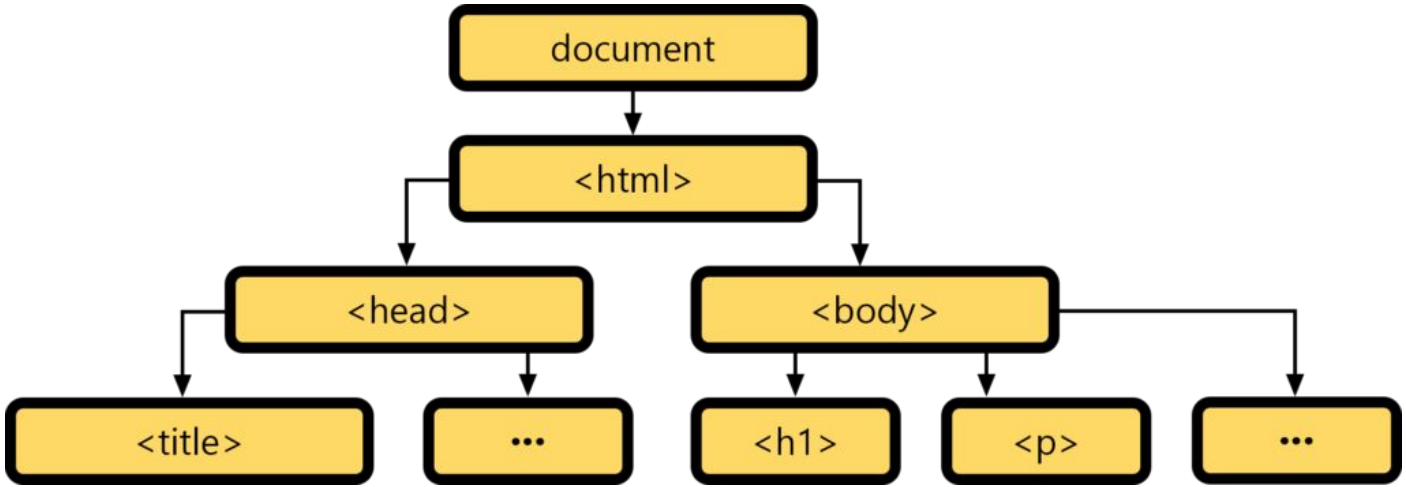
```
var myObj = {
  name: 'Ivan',
  age: 22,
  favoriteFood: 'Beef'
};

var myObjStr = JSON.stringify(myObj);

console.log(myObjStr);
// "{"name":"Ivan", "age":22, "favoriteFood":"Beef"}"

console.log(JSON.parse(myObjStr));
// Object {name:"Skip", age:22, favoriteFood:"Beef"}
```

Окружение: DOM, BOM и JS



DOM - объектная модель документа

DOM описывает структуру страницы объектами javascript.



Поиск элементов в DOM

У объекта document есть множество свойств и методов для поиска элементов на странице

```
// поиск элементов через свойство
document.children[0];

// поиск элемента по имени тега
document.getElementsByTagName('li');

// поиск элемента по атрибуту id
document.getElementById('car');

// поиск элемента по атрибуту class
document.getElementsByClassName('text');
```

JavaScript

Увидимся на практике!