**Пользовательские объекты JS.  
Специальные операторы.**

Цель: изучить способы создания пользовательских объектов, познакомиться с правилами применения специальных операторов: delete, in, instanceof, typeof.

Пользовательские объекты в JavaScript можно создать несколькими способами.

Один из способов основан на функции, в теле которой описываются все свойства и методы создаваемого объекта. Ее называют функцией-конструктором или просто конструктором объекта. Имя функции-конструктора объекта является одновременно и именем создаваемого объекта. Свойства и методы создаваемого объекта задаются в теле функции-конструктора с помощью операторов присваивания, имена переменных-свойств записываются с ключевым словом **this** (этот).

Пример создания и использования объекта, созданного с помощью конструктора.

//Конструктор для создания объекта Gruppa со свойствами n, spec, kolich и методом add\_stud

**function Gruppa(n,spec,kolich) {**

**//Свойства(номер, специальность, количество)**

**this.n=n;**

**this.spec=spec;**

**this.kolich=kolich;**

**//Метод (добавить в группу k студентов)**

**this.add\_stud=function add\_stud(k) {**

**this.kolich+=k;**

**WScript.echo('В группу ' + this.n + ' добавили '+ k + ' студентов.');**

**}**

**}**

**//Теперь можно создавать экземпляры объекта**

**gr1=new Gruppa(2, 'ИСиТ', 28);**

**//Вызов метода созданного объекта (добавить 2 студента)**

**gr1.add\_stud(2);**

**//C помощью prototype добавим объекту свойство kurs,**

**//экземпляры объекта будут иметь это свойство**

**Gruppa.prototype.kurs=this.kurs**

**gr1.kurs=2**

Обращаться к свойствам объектов можно двумя способами:

- используя точку после имени объекта, например **gr1.kurs=2**;

- заключая название свойства в квадратные скобки после имени объекта, например **gr1['kurs']=2**.

С помощью свойства **prototype** можно добавлять новые свойства и методы к конструкторам объектов. Добавленные к конструктору свойства и методы будут также добавлены ко всем объектам, которые были созданы данным конструктором. Например, Gruppa.prototype.kurs=this.kurs.

**Удаление свойств объекта (delete).**

С помощью оператора **delete** можно удалить свойство объекта, а также элемент массива. При удалении элемента массива удаляется и его индекс, но оставшиеся элементы сохраняют свои прежние индексы, а длина массива не изменяется.

Пример, delete mas[2] – удалить 3-й элемент массива.

**Проверка наличия свойств (in).**

Оператор **in** позволяет проверить, имеется ли некоторое свойство или метод у того или иного объекта. Если свойство или метод содержится в объекте, то возвращается true, иначе – false. Отсюда следует, что оператор in можно применять в условных выражениях (в операторах if, switch, for, while, do-while).

Примеры:

WScript.echo('spec' in gr1) // проверить, есть ли свойство spec у объекта gr1;

WScript.echo(1 in mas) //проверить, есть ли элемент с номером 1 в массиве mas.

**Проверка принадлежности объекта модели (instanceof)**

Оператор **instanceof** позволяет проверить, принадлежит ли некоторый объект объектной модели JavaScript. Если они совпадают, метод возвращает true, если нет false.

Выражение с оператором instanceof может использоваться в условных выражениях (в операторах if, switch, for, while, do-while).

Пример, WScript.echo(mas instanceof Array) – проверяет, является ли mas массивом.

**Определение типа (typeof)**

Оператор **typeof** позволяет проверить, относится ли значение к одному из следующих типов: string, number, boolean, object, function или undefined. Значение, возвращаемое оператором typeof, является строковым и содержит одно из перечисленных названий типа.

Пример, WScript.echo(typeof gr1.kurs).

## Проверка типов

Следующий фрагмент кода демонстрирует проверку типов c использованием оператора typeof:

|  |  |
| --- | --- |
|  | console.log(typeof ""); // "string" |
|  | console.log(typeof "hello"); // "string" |
|  | console.log(typeof String("hello")); // "string" |
|  | console.log(typeof new String("hello")); // "object" |
|  |  |
|  | console.log(typeof 0); // "number" |
|  | console.log(typeof -0); // "number" |
|  | console.log(typeof 0xff); // "number" |
|  | console.log(typeof -3.142); // "number" |
|  | console.log(typeof Infinity); // "number" |
|  | console.log(typeof -Infinity); // "number" |
|  | console.log(typeof NaN); // "number" |
|  | console.log(typeof Number(53)); // "number" |
|  | console.log(typeof new Number(53)); // "object" |
|  |  |
|  | console.log(typeof true); // "boolean" |
|  | console.log(typeof false); // "boolean" |
|  | console.log(typeof new Boolean(true)); // "object" |
|  |  |
|  | console.log(typeof undefined); // "undefined" |
|  |  |
|  | console.log(typeof null); // "object" |
|  |  |
|  | console.log(typeof Symbol()); // "symbol" |
|  |  |
|  | console.log(typeof []); // "object" |
|  | console.log(typeof Array(5)); // "object" |
|  |  |
|  | console.log(typeof function() {}); // "function" |
|  | console.log(typeof new Function); // "function" |
|  |  |
|  | console.log(typeof new Date); // "object" |
|  |  |
|  | console.log(typeof /^(.+)$/); // "object" |
|  | console.log(typeof new RegExp("^(.+)$")); // "object" |
|  |  |
|  | console.log(typeof {}); // "object" |
|  | console.log(typeof new Object); // "object" |

Обратите внимание на то, что все значения созданные с помощью ключевого слова new всегда имеют тип “object”. Исключением из этого является только конструктор Function.

**Задание 1.** Создать пользовательский объект Gruppa (использовать пример, представленный выше). Добавить метод sub\_stud (исключить из группы k студентов). Создать несколько экземпляров объекта Gruppa (gr2, gr3, gr4). Применить методы add\_stud и sub\_stud к каждому экземпляру. Вывести на страницу количество студентов в каждой группе.

**Задание 2.**

Создайте функцию AddString(obj), которая к строковым свойствам объекта obj добавляет строку «Это строка.».

Например:

// до вызова функции

let menu = {

width: 200,

height: 300,

title: "My menu"

};

AddString (menu);

// после вызова функции

menu = {

width: 200,

height: 300,

title: "My menu Это строка."

};

Обратите внимание, что AddString не нужно ничего возвращать. Следует напрямую изменять объект.

P.S. Используйте typeof для проверки, что значение свойства строковое.

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

**Задание 3.**

Выполните вариант задания.

Создать объект (способом, основанном на функции) с заданными свойствами и определить методы:

|  |  |
| --- | --- |
| **N** | **Задание.** |
| **0**  Ахматова А. Р.  Нгуен Динь Куан  Тетюков А. Н. | Объект – обыкновенная дробь.  Свойства: числитель, знаменатель.  Методы:   1. сокращение дроби; 2. перевод дроби в десятичную дробь; 3. выделение целой части. |
| **1**  Ахматова К. Р.  Никагосов М. В.  Тетюцких Д. А. | Объект – вектор.  Свойства: координаты начала, координаты конца.  Методы:   1. нахождение проекции вектора на ось hello_html_fbf90c6.gif; 2. нахождение длины вектора; 3. умножение вектора на число. |
| **2**  Ахмедьянов А. А.  Новиков А. С.  Тихонов К. А. | Объект – цилиндр.  Свойства: радиус основания, высота.  Методы:   1. нахождение объема цилиндра; 2. нахождение площади основания цилиндра; 3. нахождение площади боковой поверхности цилиндра. |
| **3**  **Ахметгалиев Р. С.**  Охотников М.А.  Туровинина М. А. | Объект – параллелепипед.  Свойства: сторона 1, сторона 2, сторона 3.  Методы:   1. нахождение длины диагонали параллелепипеда; 2. нахождение площади основания; 3. нахождение объема параллелепипеда. |
| **4**  Банда Джошуа  Паршинцев Р. В.  Урусов Н. Е.  Шувалов Д. А. | Объект – треугольник.  Свойства: стороны треугольника.  Методы:   * 1. периметр треугольника;   2. площадь треугольника;   3. радиус вписанной окружности. |
| **5**  Баранов М А  Подболотов М. А.  Федоров Г. С. | Объект – конус.  Свойства: радиус основания, высота.  Методы:   1. нахождение объема конуса; 2. нахождение площади основания конуса; 3. нахождение площади боковой поверхности конуса. |
| **6**  Ворзопова Е. В.  Ромашов Р. В.  Фёдоров Л. В. | Объект – равнобедренная трапеция.  Свойства: основание 1, основание 2, высота.  Методы:   1. нахождение средней линии трапеции; 2. нахождение площади трапеции 3. нахождение боковой стороны трапеции. |
| **7**  Е Цзиняо  Свиридов А.Д.  Хмылин Г. Ю.  Юркевич М. А. | Объект – окружность.  Свойства: координаты центра, радиус.  Методы:   1. нахождение длины окружности; 2. нахождение площади круга; 3. определение положения точки с координатами hello_html_39bd0abf.gif и hello_html_4afad5f8.gif относительно окружности. |
| **8**  Ильиных А. И.  Создыков Э. Г.  Чащин В. С.  Янычев В. В. | Объект – ромб.  Свойства: диагонали.  Методы:   1. периметр ромба; 2. площадь ромба; 3. радиус вписанной окружности. |
| **9**  Мартьянов С. М.  Сюзёв М. А.  Чезганов А. А.  Шипицын П. Е. | Объект – радиус-вектор.  Свойства: координаты радиус-вектора.  Методы:   1. нахождение полярных координат радиус-вектора ; 2. нахождение длины вектора; 3. нахождение вектора, перпендикулярного данному вектору. |