

## Лабораторная работа № 4. Пользовательские объекты JS. Специальные операторы

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

### Теория

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;
document.write('Вгруппу ' + this.n + ' добавили'+ k + '
студентов.<br\>');
}
}
//Теперь можно создавать экземпляры объекта gr1=newGruppa(2,
'ИСиТ', 28);
//Вызов метода созданного объекта (добавить 2 студента)
gr1.add_stud(2);
//С помощью prototype добавим объекту свойство kurs,
//экземпляры объекта будут иметь это свойство
Gruppa.prototype.kurs=this.kurs
gr1.kurs=2
document.write(gr1.n,gr1.spec, gr1.kolich, gr1.kurs)
</script>
```

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:

```
document.write('spec' in gr1) // проверить, есть ли свойство
spec у объекта gr1;
document.write(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, `document.write(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, `document.write(typeof gr1.kurs)`.

## Задания к лабораторной работе № 4

**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.** Создать пользовательский объект `Студент`. Свойства: имя, фамилия, физика (оценка), математика, информатика. Методы: вывести свое имя и фамилию в окно, рассчитать средний балл и вывести на страницу. Создать 3 экземпляра объекта `Студент`. Вывести информацию о всех студентах.

Добавить какое-нибудь свойство объекту `Студент`. Задать значение этого свойства для всех экземпляров. Вывести информацию о всех студентах на страницу.

Вместо объекта `Студент` можно придумать свой пример. Это приветствуется.

**Задание 3.** Использовать объекты `Math`, `Array` и `String` из лабораторной работы № 3.

`Delete`. Удалить из массива элемент с номером 3, вывести исходный массив и полученный.

`In`. Проверить, имеется ли у массива 2-й элемент. Проверить наличие любого свойства у созданных ранее пользовательских объектов `Gruppa` и `Студент`.

`Instanceof`. Проверить, являются ли созданные объекты объектами `Array`, `String`, `Object`.

`Typeof`. Создать функцию, которая просто возвращает значение 5. Узнать тип всех созданных ранее объектов, включая функцию. Узнать тип всех свойств пользовательских объектов `Gruppa` и `Студент`.