**Массивы**

Массив – упорядоченная коллекция значений, которые называются элементами массивов.

Каждый элемент характеризуется числовой позицией в массиве, которая называется индексом.

Особенности:

1. Гетерогенные: элементы могут иметь разные типы.
2. Динамические: меняют размер по мере надобности.
3. Разреженные: могут отсутствовать элементы с некоторыми индексами.
4. Специальные: числовые индексы работают как имена свойств.

**Создание массива:**

Способ 1: использование «литерала массива»

Let a = []

Let b = [1, 2 + 3];

Let c = [1, “Alex”, [3]];

Способ 2: функция Array() (вызывается с new или без):

1. Без аргументов.
2. Одно число (целое неотрицательное) – массив указанной длины, без элементов.
3. Один нечисловой аргумент – массив этого элемента.
4. Более одного аргумента – массив, который состоит из указанных элементов.

Let a = new Array();

**Метод Array.from()** создает массив. Первый аргумент – интегрируемый объект или объект, подобный массиву.

Второй опциональный аргумент – функция преобразования (маппинга) исходных элементов.

Let a1 = Array.from(‘foo’); // [‘f’, ‘o’, ‘o’]

**Метод Array.off()** создает массив из произвольного числа аргументов, вне зависимости от их числа и типа.

Любой массив – это объект.

Любой объект – коллекция свойств и значений.

Любое свойство имеет строковое имя.

Универсальное обращение к свойству: obj[name], где name автоматически приводится к типу string.

Если имя свойства – это правильный идентификатор, вместо obj[name] можно записать obj.name.

Length будет больше максимального индекса массива (и почти всегда ровно на 1 больше).

Свойства length можно записывать. Уменьшение length приводит к тому, что исчезают элементы и индексы.

**Обход массива**

Обычно обход массива выполняется при помощи for.

Каждый массив обладает набором методов, изначально определенных в объекте Array.prototype.

Многие из этих методов допускают косвенное применение к объектам, подобным массивам. Но некоторые методы меняют тот массив, у которого вызываются, а не все объекты, подобные массивам, допускают изменение (пример – строки).

Reverse()

На месте обращает порядок следования элементов.

Sort()

Concat()

Slice()

Splice()

indexOf()

lastIndexOf()

find()

findIndex()

includes()

every()

some()

filter() – массив элементов, для который f() = true

map() - возвращает массив с результатами вызова f()

forEach() – выполняет f() для каждого элемента

**Редукция элементов**

reduce(callback, initialValue?)

reduceRight(callback, initialValue?)

Методы формируют из массива скалярное значение при помощи аккумулятора. Аккумулятор – функция с четырьмя аргументами: предыдущее значение аккумулятора, текущий элемент, текущий индекс, исходный массив.

**Объекты, подобные массивам**

Это объекты, которые дают доступ к своим свойствам при помощи индекса и имеют свойство length.