**1.Литерал объекта.**

Литерал объекта — список, заключенный в фигурные  
скобки. Внутри фигурных скобок через запятую указываются пары  
значений: название свойства и значение свойства. Название свойства  
и значение разделяются двоеточиями. Если литерал объекта присвоить в качестве значения переменной, то эта переменная может быть  
отождествлена с объектом.

**2.Конструктор обьекта**

Конструктор объектов  
Само собой разумеется, что с помощью описания литералов мы можем создавать объекты практически в неограниченном количестве.  
Другими словами, никто не запрещает нам создавать объекты в индивидуальном порядке, описывая для каждого объекта его литерал.  
Но если объектов много, а набор свойств и методов у них одинаков,  
то процедура создания объектов будет достаточно утомительной. Поэтому при создании однотипных объектов (то есть объектов, имеющих одинаковые наборы свойств) имеет смысл использовать некое подобие конвейера. Более конкретно, существует возможность описать  
специальную функцию, которая позволит создавать объекты с одинаковым набором свойств. Такая функция называется конструктором объектов. Функция-конструктор по большому счету определяет  
тип объекта, набор его свойств и методов.

**3.Для чего необходимо this в конструкторе**

Свойства и методы создаваемого объекта задаются в теле функции конструктора с помощью операторов присваивания, имена переменных-свойств записываются с ключевым словом this (этот).

Ключевое свойство this представляет ссылку на текущий, то есть определяемый конструктором объект. Все операторы присвоения с this, расположенные в теле функции-конструктора, определяют свойства объекта.

1. **Свойство prototype для чего необходимо**

С помощью свойства prototype можно добавлять новые свойства и методы к конструкторам объектов. Добавленные к конструктору свойства и методы будут также добавлены ко всем его экземплярам.

**5.оператор delete**

помощью оператора delete у объекта удаляются свойства и методы.  
Оператор используется в следующем формате:  
delete объект.свойство  
После оператора delete указывается ссылка на свойство или метод.  
В результате выполнения соответствующей команды свойство или  
метод, указанные после оператора delete, удаляются из объекта.  
i   
С помощью оператора delete можно удалить не только свойство или  
метод объекта, но и сам объект. Команда удаления объекта выглядит как delete объект. После ее выполнения объект, на который ссылается переменная, указанная после оператора delete, будет удален.

**6.Что будет, если удалить элемент из массива**

При удалении элемента массива удаляется и его индекс, но оставшиеся элементы сохраняют свои прежние индексы, а длина массива не изменяется.

**7.Для чего использ оператор 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.

**8.оператор инстэнс фор**  
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)

**9.как определить тип обьекта**

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.