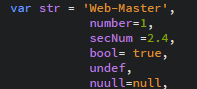
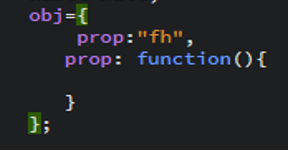
JavaScript-Обектно-Ориентированый язык с прототипной формой наследывания. Клиенский язык поддерживаеться всеми браузерами.

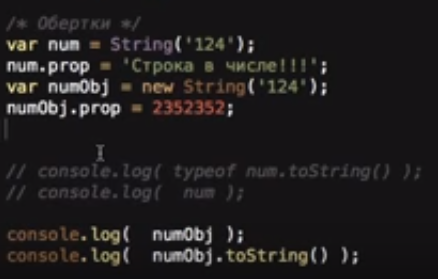
Вывод переменной в браузер .

Основные типы перепменных.

-Примитивные типы.

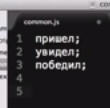
Объектный тип.

Обертки позволяют простым типам вести себя как объекты.

Вывод в консоль браузера.

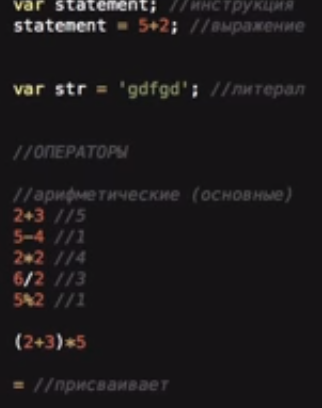
**String=”sjgfbsjvcm s”-Литерал**

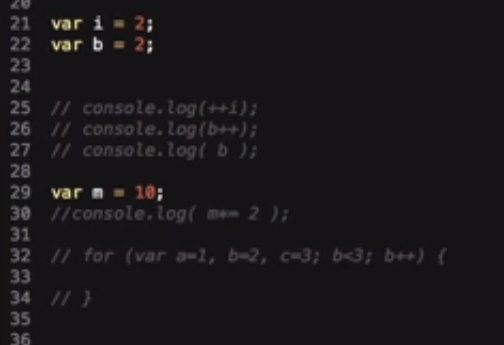
Statements-опредиленные команды(инструкции).

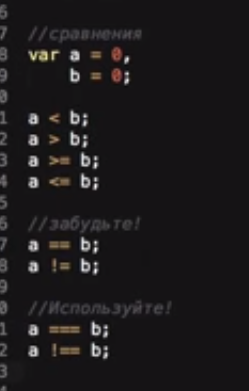


Выражение всегда возвращает каке нибудь значение.

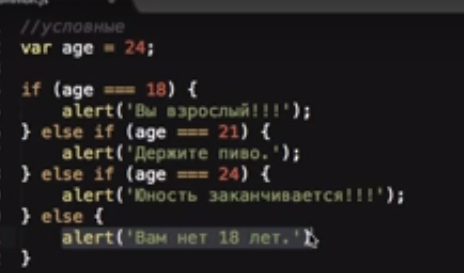
Операторы:



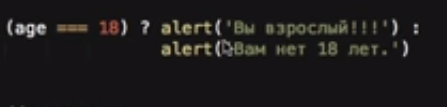


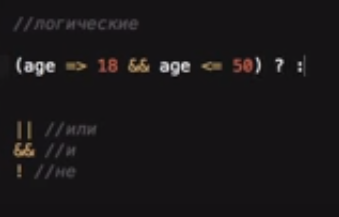
Не различает типы данных.

Строгое равенство и не раваенство.

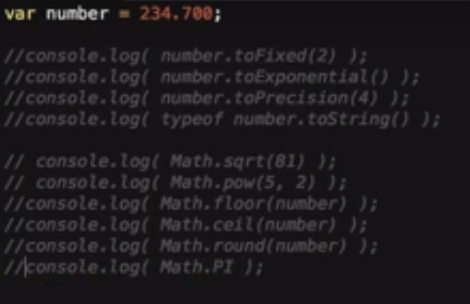
Унарный оператор

Тернарный оператор





Работа с числами

парсит в строку

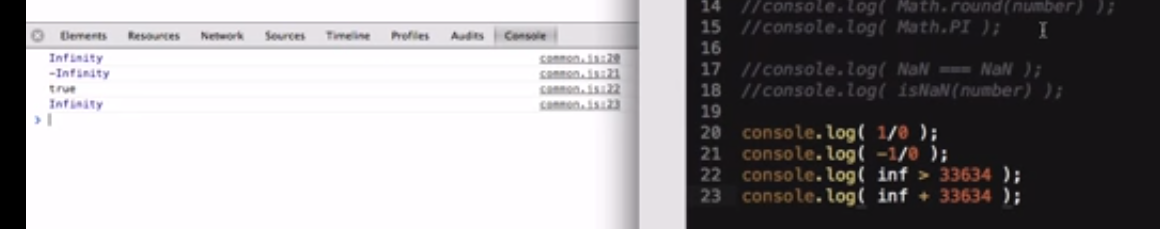
К-во чисел после зап.

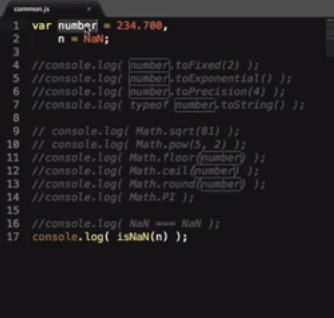
Число в эксп виде

Округление с опр кол цыфр до/после точки

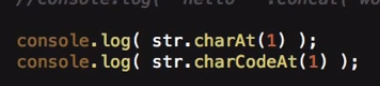
Работа с числами под средством библиотеки Math(степень,корень.)

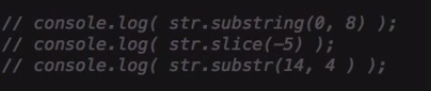
И т д

  
Бесконечность больше любого числа,все операции с ней приводят к получению бесконечности!

Метод проверки являеться ли переменная NaN.

NaN не равно NaN.!!

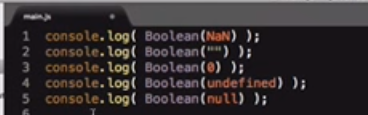


Получить символ из строки под соотвтствующим индексомПолучить код символа из табл ASCII  
мтоды разбития строки на подстроки

разделение строки на массив

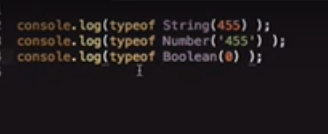
по символу выводим индекс

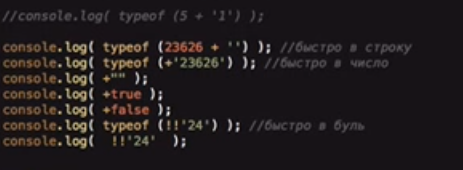
Все типы по умолчатию true кроме

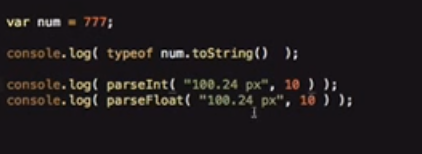


Когда у нас будет undefined:не существующий элемент,свойство,не передать параметры в функцию.

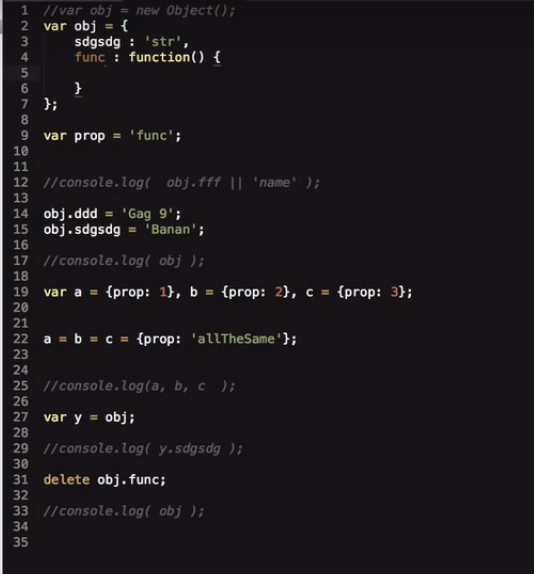
Разница между null and undefined-и то и то обозначает пустоту undefined то что присваевает JS null мы присваеваем сами.

Явное преобразование данных в JS

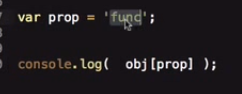


Может указываться система счисления.

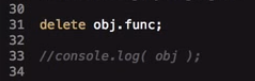
Метод это функция которая является свойством объекта

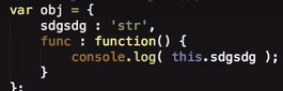
присваевание одного значения нескольким объектам.

Инициализация объектов

Есть точечная анотация а есть скобочная последняя используеться в случае когда нам нужно «вытащить значение переменных» 

Добав-ление новой переменнойп изменение значения старой

удаление свойства объекта

доступ к объекту или его свойству в методе(в функции)

В JS изначально все переменные глобальные!!

Обращение к функцие. 

Глобальная переменная-инициализация 

Способ создания массива (способ литерала)

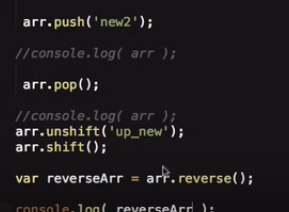
Конструктор: 

Удаление элементов 

Но лучше использывать первое число с какого элемента начинаем удалять второе –сколько элементов удалим. И нет так называемых «Дырок» в массиве.

Можо задать длинну масива самостоятельно 

конкатонация масива

добавление элемента в конец

удаляет последний элемент

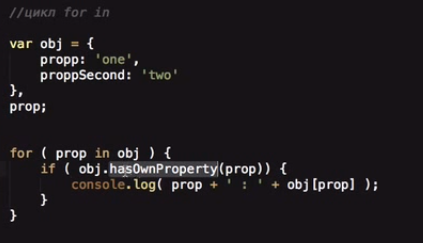
доб в начало

удаляет первый элем

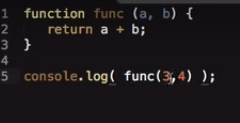
переварачиает массив

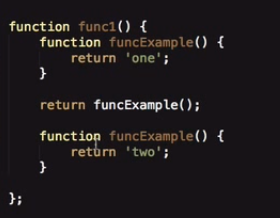
 делает подмассив с какого по какой элемент

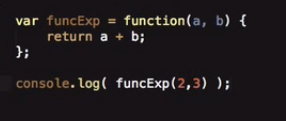
 Сортировка

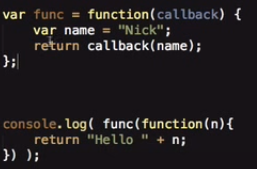
Проверка на принадлежность к данному объекту

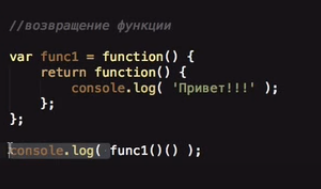
цыкл используют для прохлждения непосредственно по свойству объекта.  
Пример объявления функции

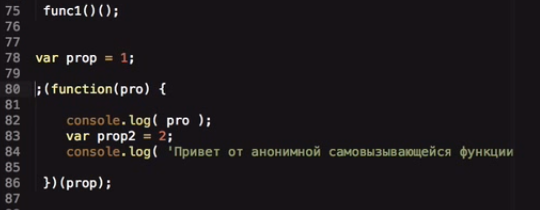
FunctionDeclaration

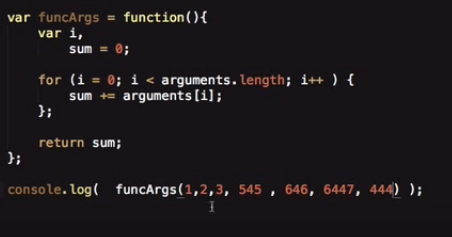
В данном случае.Функция будет перезаписана а только потом выведена.Одна из главных особенностей этой функции.

FunctionExpretion

Функции обратного вызова.

Вазвращение функции

 Анонимная самовызывающайся функция. Позволяен Скрыть наш код от внешней видимости.

 Свойство «arguments» имеет общие свойства с массивом но таковым не являеться. С помощью arguments мы можем передавать в функцию не ограниченое каличество аргументов.