Функции обертки

Используется как синтаксический сахар. Обертываем функцию своим названием.

Для нормальной кодировки в HTML: <meta charset=”UTF-8”><meta>

Копировать массив: a3 = […array]

Тип BigInt

b в конце числа

o.f.call(…….) чтобы добавить параметры

o.f.apply(…..) добавить параметры массивом

this

в JS this может указывать и на метод, и на функцию

Имеет более широкую нагрузку: глобальный контекст или другое значение.

Стрелочные функции

Для удобства, сахар, пишутся в одной строке f = (a,b) => (a+b)

Явное и неявное преобразование типов в js

Приводит в соответствии с документацией ЭтмоСкрипт

cl('5'+5) 🡪 55

cl(5 + '5') 🡪 55

cl(5 + Number(‘5’)) 🡪 10

cl(5+'5'\*1) 🡪 10

cl(!!5) 🡪 true (преобразование к Boolean)

== и != преобразуют типы

=== и !== не преобразуют типы

Создание новых объектных типов JS

Объектный тип определяется функцией, которая породила этот объект

var s = new String("hello world");

cl(s.constructor);

cl(s.\_\_proto\_\_);

cl(String.prototype);

cl(s.\_\_proto\_\_ === String.prototype);

Для функции str нужно что-то указать в proto.

У ф-ии есть свойство prototype(там лежит объект-прототип, который она кладет как ссылку новый объект) и она изначально его туда закладывает