



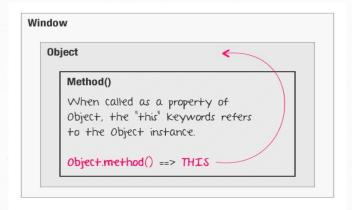
Education **Academy**

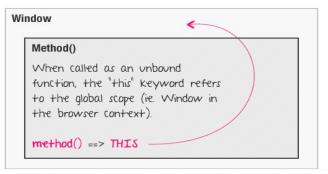
WWW.ITEDUCATE.COM.UA

JavaScript Pro

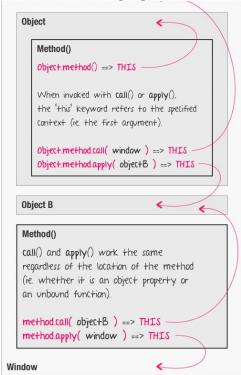
- Контекст выполнения.
- Наследование и extend.
- Замыкания.

Контекст по умолчанию





Call & Apply window



Метод call

Метод call может применяться для вызова функции в контексте нужного объекта

Синтаксис

var result = fun.call(thisArg[, arg1[, arg2[, ...]]])

Аргументы

thisArg

Задает значение this внутри функции. Если thisArg - null или undefined, то это будет глобальный объект. В ином случае, this будет равноObject(thisArg) (то есть thisArg, если thisArg уже объект, или String,Boolean или Number, если thisArg - примитивное значение соответствующего типа). Таким образом, при выполнении функции всегда соблюдается условиеtypeof this == 'object'.

arg1, arg2 ..

Аргументы, с которыми будет вызвана функция.

Метод apply

Метод apply() вызывает функцию с указанным значением this и аргументами, предоставленными в виде массива

Синтаксис

var result = function.apply(thisArg[, argsArray]);

Аргументы

thisArg

Задает значение this внутри функции.

Если thisArg - null или undefined, то это будет глобальный объект.

В ином случае, this будет равно Object(thisArg) (то есть thisArg, если thisArg уже объект, или String, Boolean или Number, если thisArg - примитивное значение соответствующего типа). Таким образом, при выполнении функции всегда соблюдается условие typeof this == 'object'.

argsArray

Массив аргументов, с которыми будет вызвана функция, или null/undefined для вызова без аргументов.



Mетод bind

Metog bind() создаёт новую функцию, которая при вызове устанавливает в качестве контекста выполнения this предоставленное значение. В метод также передаётся набор аргументов, которые будут установлены перед переданными в привязанную функцию аргументами при её вызове.

Синтаксис

var wrapper = func.bind(context[, arg1, arg2...])

Аргументы

func

Произвольная функция

context

Контекст, который привязывается к func

arg1, arg2, ...

Если указаны аргументы arg1, arg2... — они будут прибавлены к каждому вызову новой функции, причем встанут перед теми, которые указаны при вызове.

Результат вызова func.bind(context) аналогичен вызову bind(func, context), описанному выше. То есть, wrapper — это обёртка, фиксирующая контекст и передающая вызовы в func. Также можно указать аргументы, тогда и они будут фиксированы, но об этом чуть позже.



Каррирование

Карринг (currying) или каррирование — термин функционального программирования, который означает создание новой функции путём фиксирования аргументов существующей.

Метод func.bind(context, ...) может создавать обёртку, которая фиксирует не только контекст, но и ряд аргументов функции.

КОНТАКТНЫЕ ДАННЫЕ

ITEA ул. Кутузова, 18/7, оф. 301 Киев 03133

+38 044 590 08 38 info@iteducate.com.u a www.iteducate.com.u https://www.facebook.com/ITeducate

