



Базовый курс

Занятие #6. Замыкания

Цель занятия

Закрепление навыков работы с замыканиями

Задание 0. Хороший bind

Написать функцию `myBind(fn, newThis)` работа которой должна быть полностью аналогична работе `fn.bind(newThis)`;

Задание 1. Плохой bind

Написать функцию `myBind2(fn, newThis, n1, n2, ..., nn)` работа которой должна быть полностью аналогична работе `fn.bind(newThis, n1, n2, ..., nn)`. Количество “дополнительных аргументов” `n1, n2... nn` может быть любым.

Напи

Задание 2. Злой bind

Создайте глобальный объект. Давайте назовем его `R`.
его содержимое - два ключа - один `“_”` (подчеркивание) в котором лежит.... ну пусть слово “котик”, второй - `bind` - функция.

функция `R.bind` выполняет “частичное фиксирование параметров функции”. Т.е. если при вызове `R.bind` стоит какое-то значение - то оно будет “зафиксировано” - т.е. как бы мы не извращались с итоговой функцией - у нас все равно в этом параметре придет заданное значение. Если же мы передаем вместо параметра `R._` - то этот параметр становится “свободным” и работает так же как в `bind`. В остальном работа функции аналогична функции `bind` (ВНИМАНИЕ НА ПРИМЕРЫ)

Пример:

```
function g(a, b, c) {  
  console.log(this, a, b, c);  
}
```

```
}  
g(1, 2, 3) → Window, 1, 2, 3;  
  
var p1 = R.bind(g, 1, 2, 3, R._);  
  
p1() → 1, 2, 3, undefined  
p1(4) → 1, 2, 3, 4  
  
var p2 = R.bind(g, 1, R._, 2, 3);  
  
p2() → 1, undefined, 2, 3  
p2(4) → 1, 4, 2, 3  
  
var p3 = R.bind(g, 1, R._, R._);  
p3() → 1, undefined, undefined, undefined  
p3(4) → 1, 4, undefined, undefined  
p3(4,5) → 1, 4, 5, undefined  
p3(4,5,6) → 1, 4, 5, 6
```