

DECORATORS AND CALL FORWARDING

WHAT IS IT?

Wrapper on top of any function

```
1 function calculate (number = 10000000) {  
2     console.time('calculate');  
3     let result = 0;  
4     for (let index = 0; index < number; index += 1) {  
5         result = result += index;  
6     }  
7     console.timeEnd('calculate');  
8     return result;  
9 }
```

```
1 function memo (func) {  
2     const cache = {};  
3  
4     return function memoiedFunc (arg) {  
5         if (cache[arg]) {  
6             return cache[arg];  
7         }  
8  
9         // result  
10        cache[arg] = func(arg);  
11  
12        return cache[arg];  
13    };  
14 }
```

```
1  function memo (func) {
2      const cache = {};
3
4      return function memoiedFunc (arg) {
5          const date = Date.now();
6          if (cache[arg]) {
7              console.log('EXISTING in memoiedFunc = ' + (Date.now() - date));
8              return cache[arg];
9          }
10
11         // result
12         cache[arg] = func(arg);
13         console.log('NEW in memoiedFunc = ' + (Date.now() - date));
14
15         return cache[arg];
16     }
17 }
```

```
1  function memo (func) {
2      const cache = {};
3
4      return function memoiedFunc (...args) {
5          if (cache[args]) {
6              return cache[args];
7          }
8
9          // result
10         cache[args] = func.apply(this, args);
11
12         return cache[args[0]];
13     };
14 }
15
```

```
6         return cache[args];
7     }
8
9     // result
10    cache[args] = func.apply(this, args);
11
12    return cache[args[0]];
13 };
14 }
15
16 rand = memo(Math.random.bind(Math));
17 rand(25, 35);
18 rand(12, 11);
19 rand(12, 6);
20 rand(12, 11); // warm
```

DETAILS

Good decorators implementation

@decorator proposal