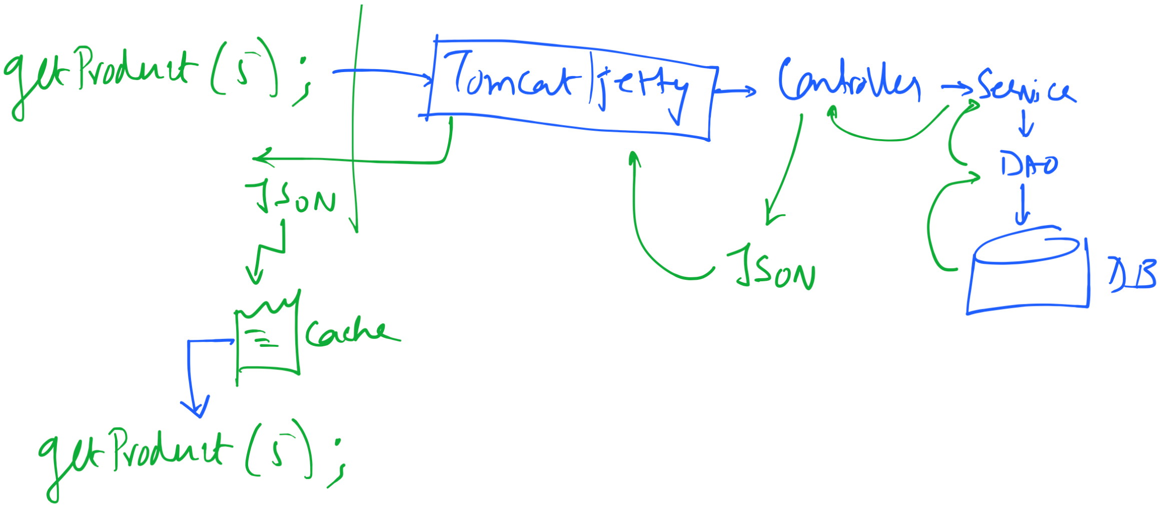
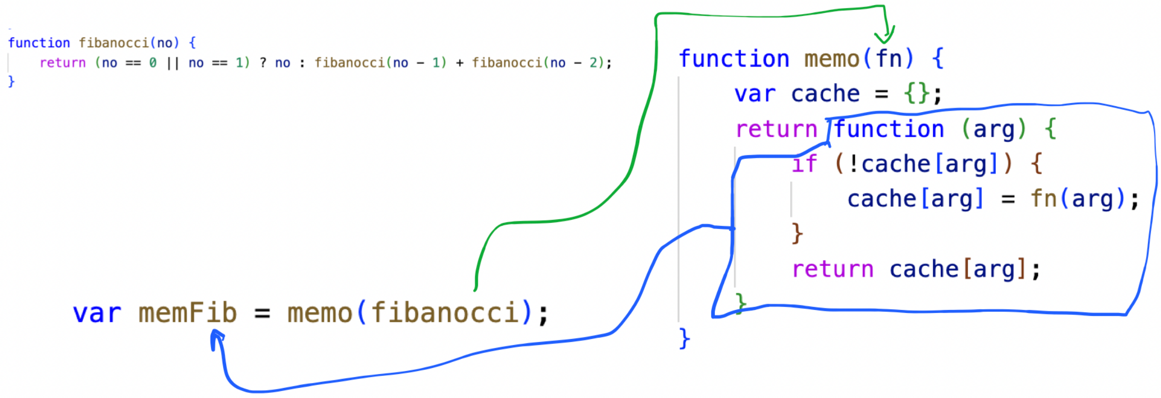
Memoization Pattern

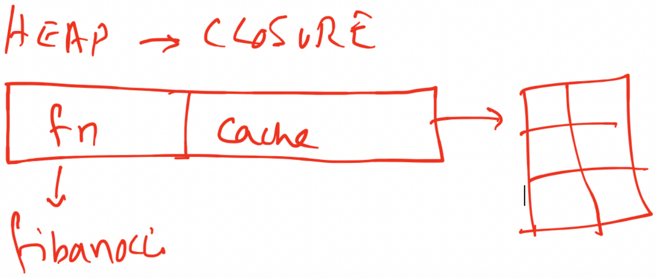


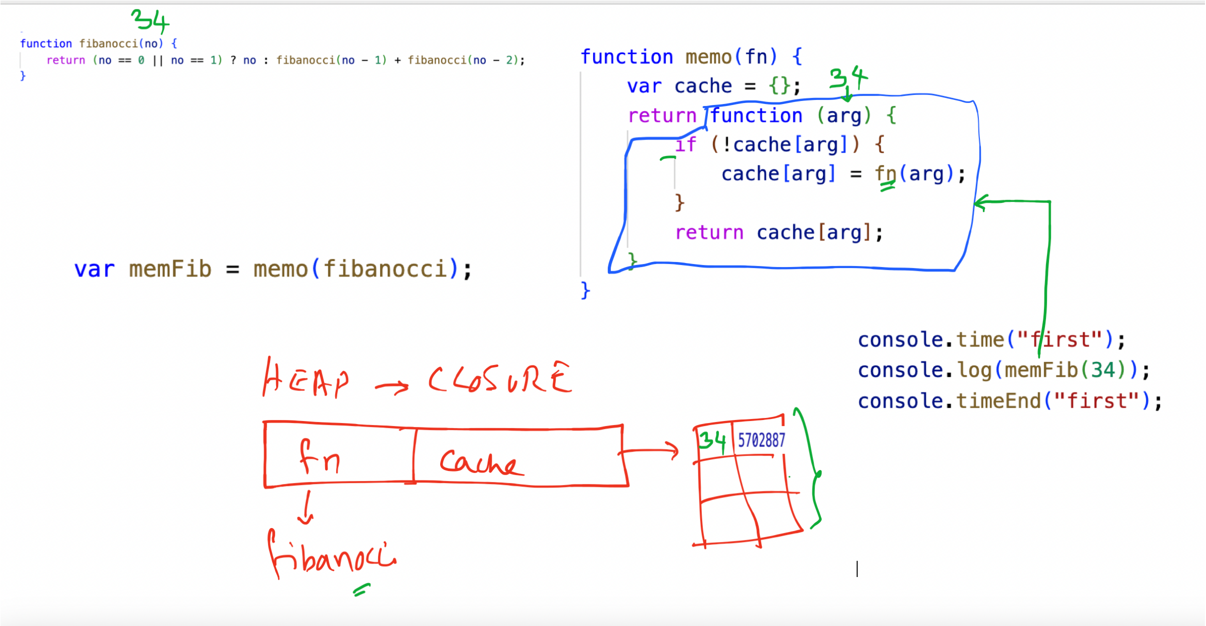


```
function fibanocci(no) {
    return (no == 0 || no == 1) ? no : fibanocci(no - 1) + fibanocci(no - 2);
}
```

```
function memo(fn) {
    var cache = {};
    return function (arg) {
        if (!cache[arg]) {
            cache[arg] = fn(arg);
        }
        return cache[arg];
    }
}
```

```
var memFib = memo(fibanocci);
```



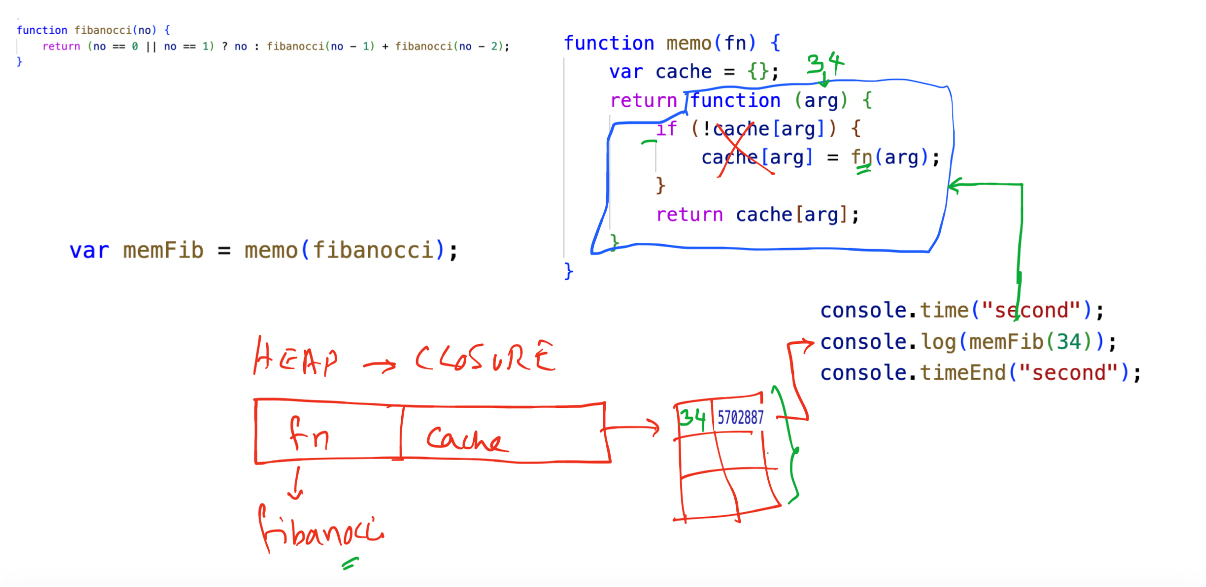


```
function fibanocci(no) {
    return (no == 0 || no == 1) ? no : fibanocci(no - 1) + fibanocci(no - 2);
}
```

```
var memFib = memo(fibanocci);
```

```
function memo(fn) {
    var cache = {};
    return function (arg) {
        if (!cache[arg]) {
            cache[arg] = fn(arg);
        }
        return cache[arg];
    }
}
```

```
console.time("first");
console.log(memFib(34));
console.timeEnd("first");
```



```
function fibanocci(no) {
    return (no == 0 || no == 1) ? no : fibanocci(no - 1) + fibanocci(no - 2);
}
```

```
var memFib = memo(fibanocci);
```

```
function memo(fn) {
    var cache = {};
    return function (arg) {
        if (!cache[arg]) {
            cache[arg] = fn(arg);
        }
        return cache[arg];
    }
}
```

```
console.time("second");
console.log(memFib(34));
console.timeEnd("second");
```