# Debugging JS with Chrome

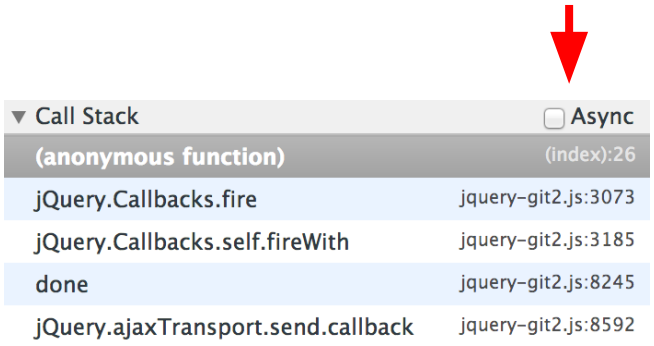
>The little power tools

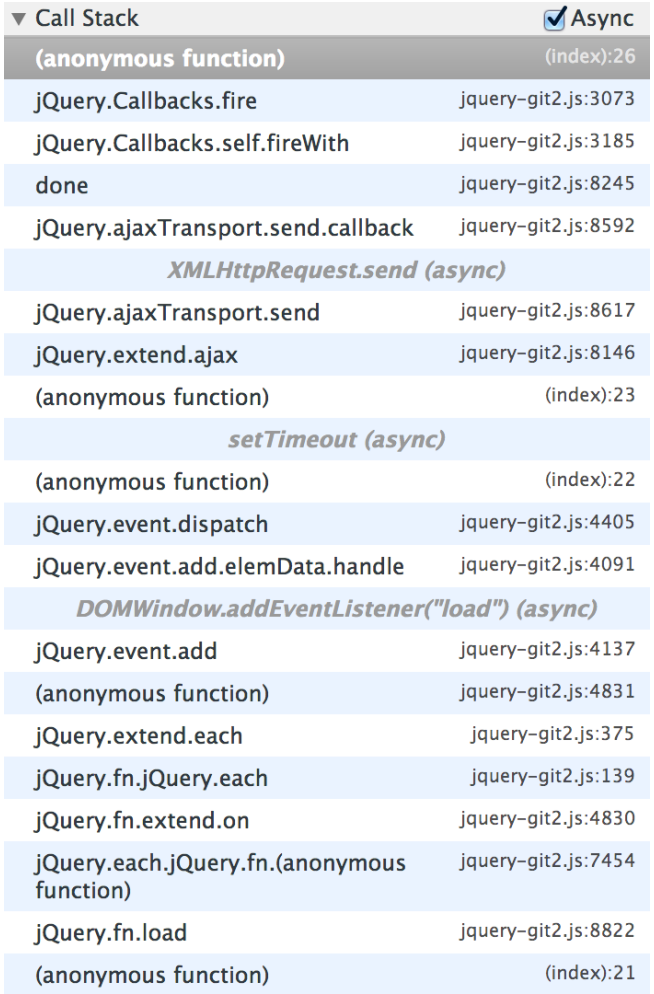
Igor Zalutsky

fitbit

# Async call stacks

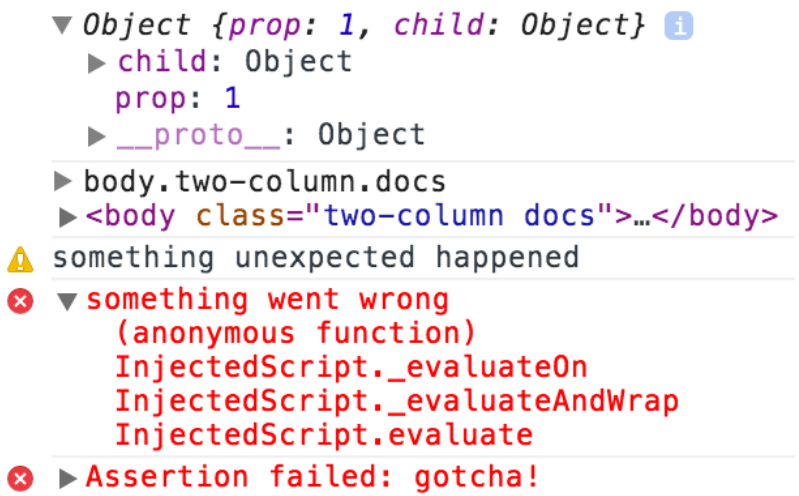
```
setTimeout(function () {
    $.ajax({
        url: 'http://echo.jsontest.com/foo/bar'
    }).done(function(response) {
        debugger;
    });
}, 500);
```



# Console API: the basics

```
var obj = {
  prop: 1, child: {
    prop: 2
    }
};
console.log(obj);
console.log('%O\n%o', document.body, document.body);
console.warn('something unexpected happened');
console.error('something went wrong');
console.assert("str" instanceof String, 'gotcha!');
```



# Console API: time() and count()

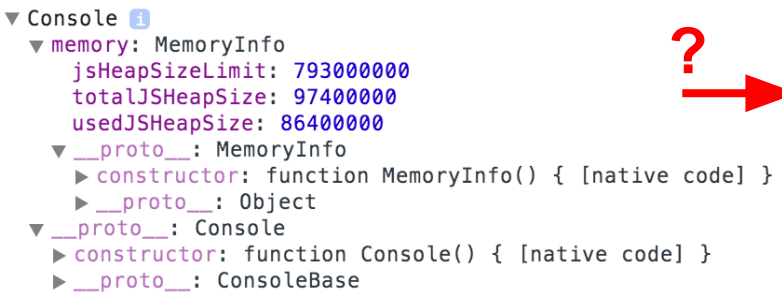
```
function fibonacci(n) {
  return (n > 1) ? fibonacci(n - 2) + fibonacci(n - 1) : n;
}
console.time('fibonacci');
console.log(fibonacci(30));
console.timeEnd('fibonacci');

document.addEventListener('keydown', function(event) {
  console.count(String.fromCharCode(event.keyCode));
});
```

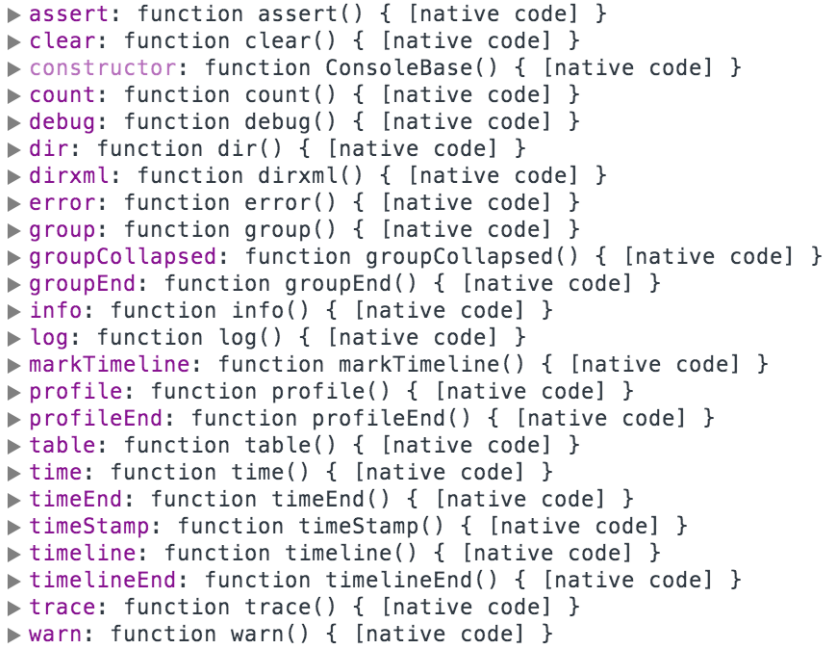
```
832040
fibonacci: 14.798ms
undefined
W: 1
W: 2
W: 3
A: 1
S: 1
D: 1
D: 2
```

# Console from the inside

console.dir(console);




?



# Console API: table()

```
var people = [
    {
        name: 'Hopper', surname: 'Herring'
    }, {
        name: 'Sampson', surname: 'Douglas'
    }, {
        name: 'Carmella', surname: 'Vincent'
    }
];
console.table(people);
console.table(
    document.querySelectorAll('a'),
    ['href', 'text']
);
```

| (index) ▲ | name | surname |
|---|---|---|
| 0 | "Hopper" | "Herring" |
| 1 | "Sampson" | "Douglas" |
| 2 | "Carmella" | "Vincent" |

VM3826:11

| (index) | href | text ▲ |
|---|---|---|
| 132 | "https://developers.g… | " " |
| gc-logo | "https://developers.g… | "" |
| 0 | "https://developers.g… | "" |
| 68 | "https://developers.g… | "0.1" |
| 58 | "https://developers.g… | "1.0" |
| 47 | "https://developers.g… | "1.1" |
| 130 | "http://www.apache.or… | "Apache 2.0 License" |
| 13 | "https://developers.g… | "Authoring and Develo… |
| 96 | "https://developers.g… | "Blog posts" |
| 136 | "https://developers.g… | "Careers" |

# Console API: other methods

| | |
|---|---|
| `clear()` | remove all previous output |
| `debug(), info()` | aliases for log() |
| `group(), groupEnd()` | grouped output |
| `profile(), profileEnd()` | last evaluated expression |
| `timeStamp()` | marking the timeline |
| `trace()` | stack trace |

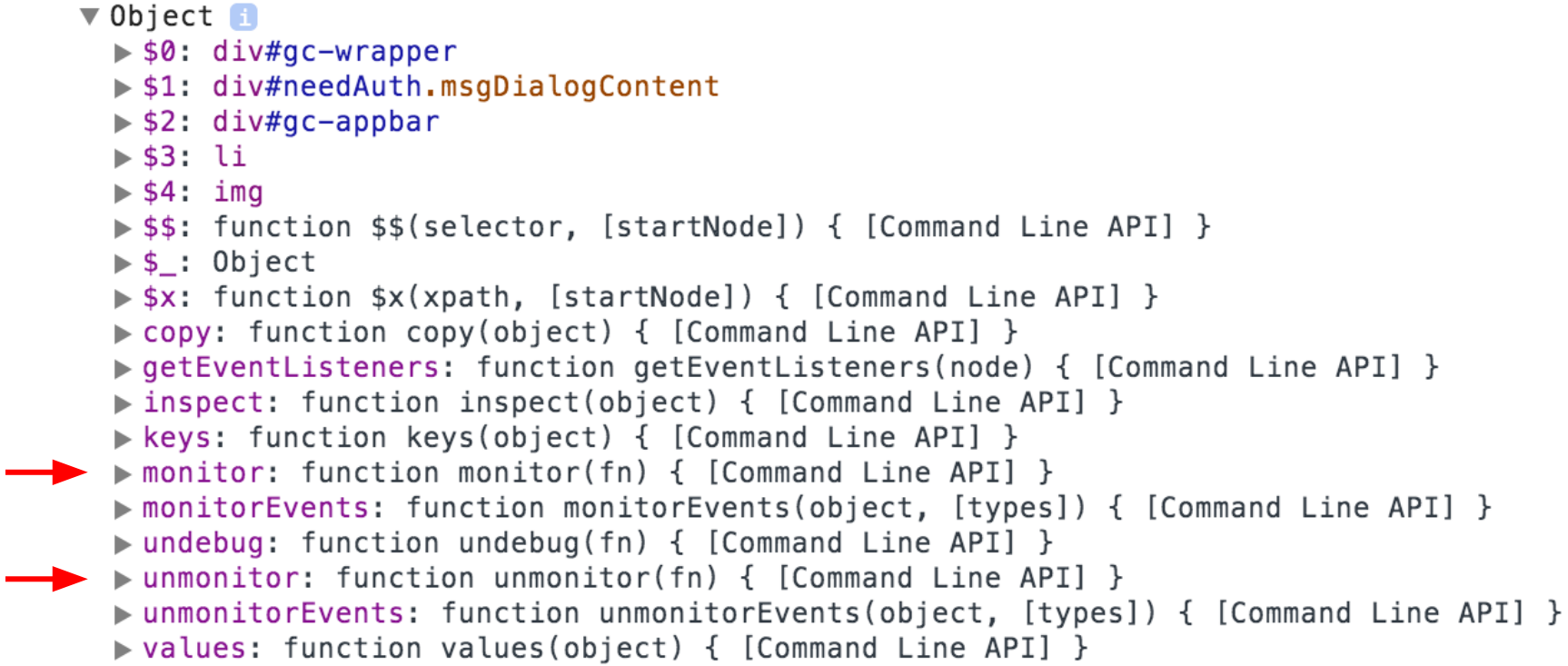
# Digging out Command Line API

```
(function() {
   debugger;
})();
```

```
with (__commandLineAPI || { __proto__: null }) {
(function() {
    debugger;
})();
}
```

```
var members = Object.keys(__commandLineAPI)
  .filter(function(key) {
    return ! (key in console);
}).reduce(function(obj, key) {
   obj[key] = __commandLineAPI[key];
   return obj;
}, {});
console.dir(members);
```

# Command Line API: overview



# Command Line API: shortcuts

| | |
|---|---|
| `$(selector)` | `document.querySelector()` |
| `$$(selector)` | `document.querySelectorAll()` |
| `$_` | last evaluated expression |
| `$0 - $4` | last 5 inspected DOM nodes or heap entries |
| `$x(path)` | XPath query |

# Command Line API: monitoring events

```
> monitorEvents(document, 'mousedown');
  undefined
  mousedown ▶ MouseEvent {dataTransfer: null, toElement: div#gc-wrapper, fromElement: null, y: 116, x: 171…}
  mousedown ▶ MouseEvent {dataTransfer: null, toElement: th, fromElement: null, y: 74, x: 394…}
> monitorEvents(document, 'key');
  undefined
  mousedown ▶ MouseEvent {dataTransfer: null, toElement: div#gc-wrapper, fromElement: null, y: 130, x: 189…}
  keydown ▶ KeyboardEvent {altGraphKey: false, repeat: false, metaKey: false, altKey: false, shiftKey: false…}
  keypress ▶ KeyboardEvent {altGraphKey: false, repeat: false, metaKey: false, altKey: false, shiftKey: false…}
  keyup ▶ KeyboardEvent {altGraphKey: false, repeat: false, metaKey: false, altKey: false, shiftKey: false…}
> unmonitorEvents(document, ['keyup', 'keydown']);
  undefined
  mousedown ▶ MouseEvent {dataTransfer: null, toElement: td, fromElement: null, y: 147, x: 345…}
  keypress ▶ KeyboardEvent {altGraphKey: false, repeat: false, metaKey: false, altKey: false, shiftKey: false…}
>
```

# Command Line API: hidden hooks

```
> function f() {};
  monitor(f);
  f(1); f(2); f(3);
  function f called with arguments: 1
  function f called with arguments: 2
  function f called with arguments: 3
< undefined
> debug(f);
  f(4);
  function f called with arguments: 4
< undefined
> undebug(f);
  unmonitor(f);
```

```
1 with (__commandLineAPI || { __proto__: null }) {
2 function f() {};
3 monitor(f);
4 f(1); f(2); f(3);
5 }
```

# Command Line API: other methods

| | |
|---|---|
| `copy(object)` | copies string representation to clipboard |
| `inspect(object)` | shows object in DOM inspector or profiler |
| `getEventListeners(object)` | returns hash of arrays of listeners |
| `profile(), profileEnd()` | last evaluated expression |
| `keys(object)` | same as Object.keys(object) |
| `values(object)` | returns array of object’s values |

# Thanks!

## Useful links

- Google Chrome Console API docs
- Google Chrome Command Line API docs
- “Lesser-Known JavaScript Debugging Techniques” by Amjad Masad
- “Advanced JavaScript Debugging with console.table()” by Marius Schulz

## Follow me!

 @igorzij

 github.com/zij

 linkedin.com/in/izalutsky