Codifying .caller, .arguments

Claude Pache (as *es-legacy-function-reflection*)
Jack Works (as *proposal-strict-built-in-functions*)
presented by Mark S. Miller, Agoric

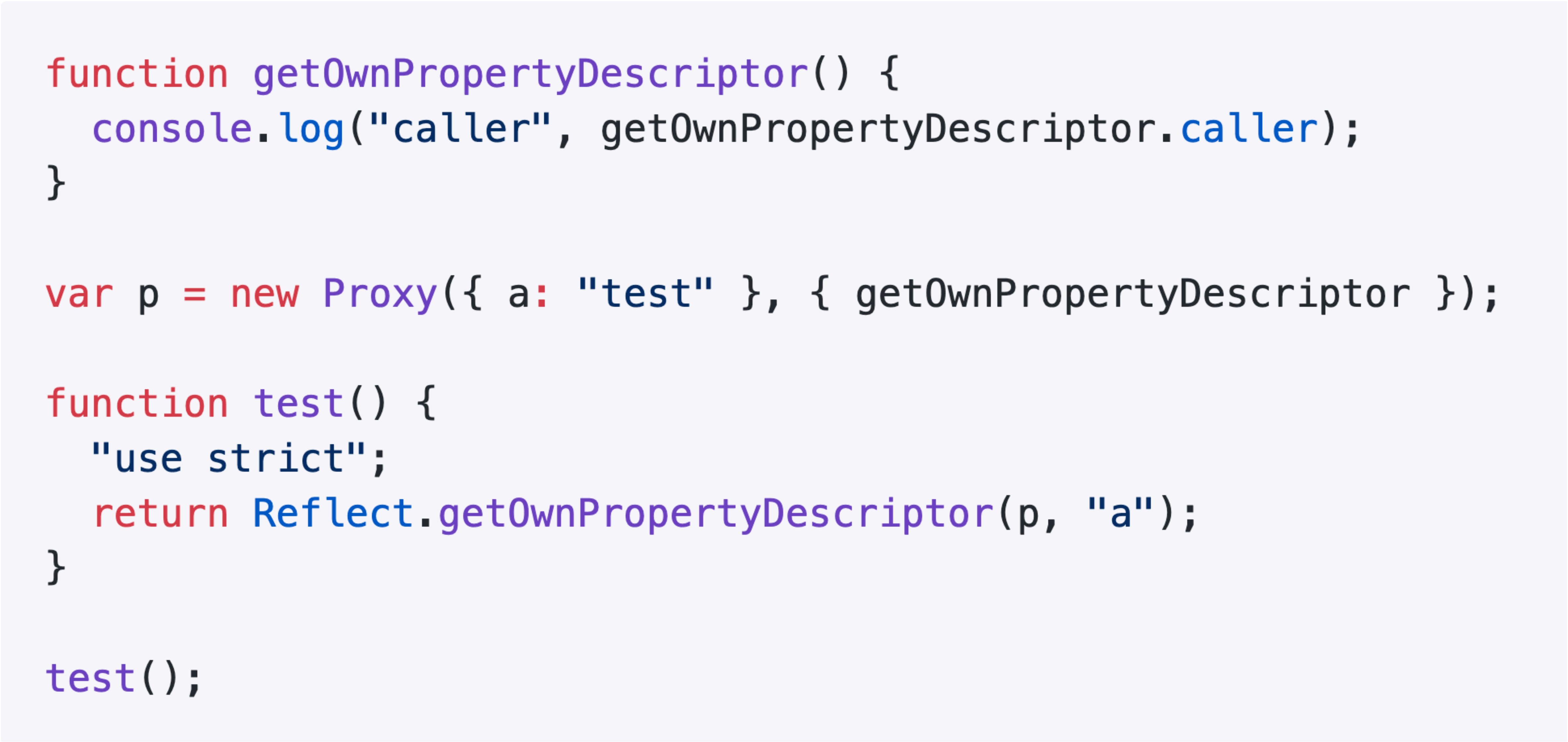
tc39 February 2020, Oahu Hawaii

```
function getOwnPropertyDescriptor() {
  console.log("caller", getOwnPropertyDescriptor.caller);
}

var p = new Proxy({ a: "test" }, { getOwnPropertyDescriptor });

function test() {
  "use strict";
  return Reflect.getOwnPropertyDescriptor(p, "a");
}

test();
```



```javascript
function getOwnPropertyDescriptor() {
  console.log("caller", getOwnPropertyDescriptor.caller);
}

var p = new Proxy({ a: "test" }, { getOwnPropertyDescriptor });

function test() {
  "use strict";
  return Reflect.getOwnPropertyDescriptor(p, "a");
}

test();
```

Firefox	`TypeError: access to strict mode caller function censored`
Chrome	`caller null`
Safari	`caller null`
Edge	`caller function getOwnPropertyDescriptor() { [native code] }`

```
defineOwnProperties(Function.prototype, {
  caller: {
    get() {

    },
    set: undefined,
    enumerable: false,
    configurable: true
  },
  arguments: {...}
});
```

```
defineOwnProperties(Function.prototype, {
  caller: {
    get() {
      // Normative optional: allowed
      throw new TypeError('...');






    },
    set: undefined,
    enumerable: false,
    configurable: true
  },
  arguments: {...}
});
```

```
defineOwnProperties(Function.prototype, {
  caller: {
    get() {
      if (!sloppyTarget(this, currentRealm)) {
        throw new TypeError('...');
      }
      const result = magicCaller(this);




      return result;
    },
    set: undefined,
    enumerable: false,
    configurable: true
  },
  arguments: {...}
});
```

```
defineOwnProperties(Function.prototype, {
 caller: {
  get() {
   if (!sloppyTarget(this, currentRealm)) {
    throw new TypeError('...');
   }
   const result = magicCaller(this);
   if (!sloppyCaller(result, currentRealm)) {
    return null;
   }
   return result;
  },
  set: undefined,
  enumerable: false,
  configurable: true
 },
 arguments: {...}
});
```

`sloppyTarget`

✔ = returns null when not in the stack frame
💥 = always throws a TypeError

type of target	Firefox 71	Chrome 79	Safari 13	Edge 18	Proposed
non-function	💥	N/A	N/A	N/A	💥
builtin	💥	💥	💥	💥	💥
strict	💥	💥	💥	💥	💥
generator/async	💥	💥	💥	💥	💥
non-constructor	💥	💥	💥	💥	💥
sloppy	✔	✔	✔	✔	✔
cross-realm	✔	N/A	N/A	N/A	💥

sloppyCaller

✔ = returns the purported caller
⛔ = returns null
💥 = throws a TypeError

type of purported caller	Firefox 71	Chrome 79	Safari 13	Edge 18	Proposed
builtin	⛔	⛔	⛔	✔ 👎	⛔
strict	💥	⛔	💥	💥	⛔
generator/async	✔	✔	💥	✔	⛔
non-constructor	✔	✔	✔	✔	✔
sloppy	✔	✔	✔	✔	✔
cross-realm	✔	✔	✔	✔	⛔

sloppyCaller

✔ = returns the purported caller

⛔ = returns null

💥 = throws a TypeError

type of purported caller	Firefox 71	Chrome 79	Safari 13	Edge 18	Proposed
builtin	⛔	⛔	⛔	✔ 👎	⛔
strict	💥	⛔	💥	💥	⛔
generator/async	✔	✔	💥	✔	⛔
non-constructor	✔	✔	✔	✔	✔
sloppy	✔	✔	✔	✔	✔
cross-realm	✔	✔	✔	✔	⛔

set: undefined,

⛔ = assignment fails silently
💥 = a TypeError is thrown

operation	Chrome, Safari, Edge 79, 13, 18	Firefox 71	Proposed
sloppyTarget.caller = 42;	⛔	⛔	⛔
other.caller = 42;	💥	💥	⛔

"use strict";
sloppyTarget.caller = 42;
	💥	⛔	💥

"use strict";
other.caller = 42;
	💥	💥	💥

Don’t break the web

- Cross-browser behavior

Protect

- No new leakage
- Avoid cross-realm leakages

Builtin vs strict functions

- No observable diff. Indistinguishable

Deprecate

- TypeError > null > just working
- Normative optional-ish
- Fixable, removable

Minimize API

Minimize Implementation Burden

`sloppyCaller`

Questions?

✔ = returns the purported caller
⛔ = returns null
💥 = throws a TypeError

type of purported caller	Firefox 71	Chrome 79	Safari 13	Edge 18	Proposed
builtin	⛔	⛔	⛔	✔ 👎	⛔
strict	💥	⛔	💥	💥	⛔
generator/async	✔	✔	💥	✔	⛔
non-constructor	✔	✔	✔	✔	✔
sloppy	✔	✔	✔	✔	✔
cross-realm	✔	✔	✔	✔	⛔