- Sam Baxter
- Rachit Nigam
- Arjun Guha
- Joe Politz
- Shriram Krishnamurthi

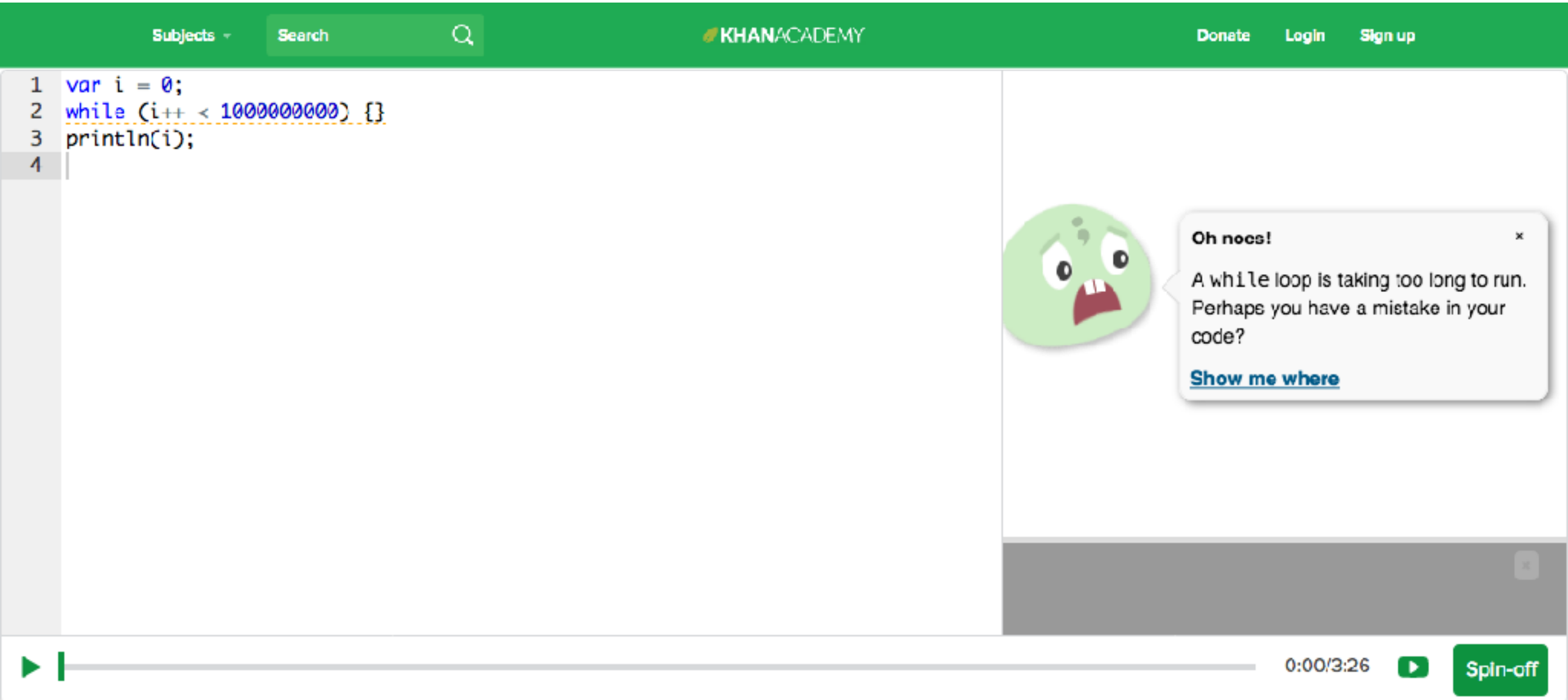
STOPIFY

WRESTLING CONTROL FROM WEB BROWSERS

WEB PROGRAMMING ENVIRONMENTS

WEB PROGRAMMING ENVIRONMENTS

WEB PROGRAMMING ENVIRONMENTS



THE BROWSER RUNTIME

Run

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```

THE BROWSER RUNTIME

Run Stop

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```

THE BROWSER RUNTIME

Run Stop

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```

EVENT QUEUE

Run

THE BROWSER RUNTIME

Run Stop

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```

i = 2

i = 1

i = 0

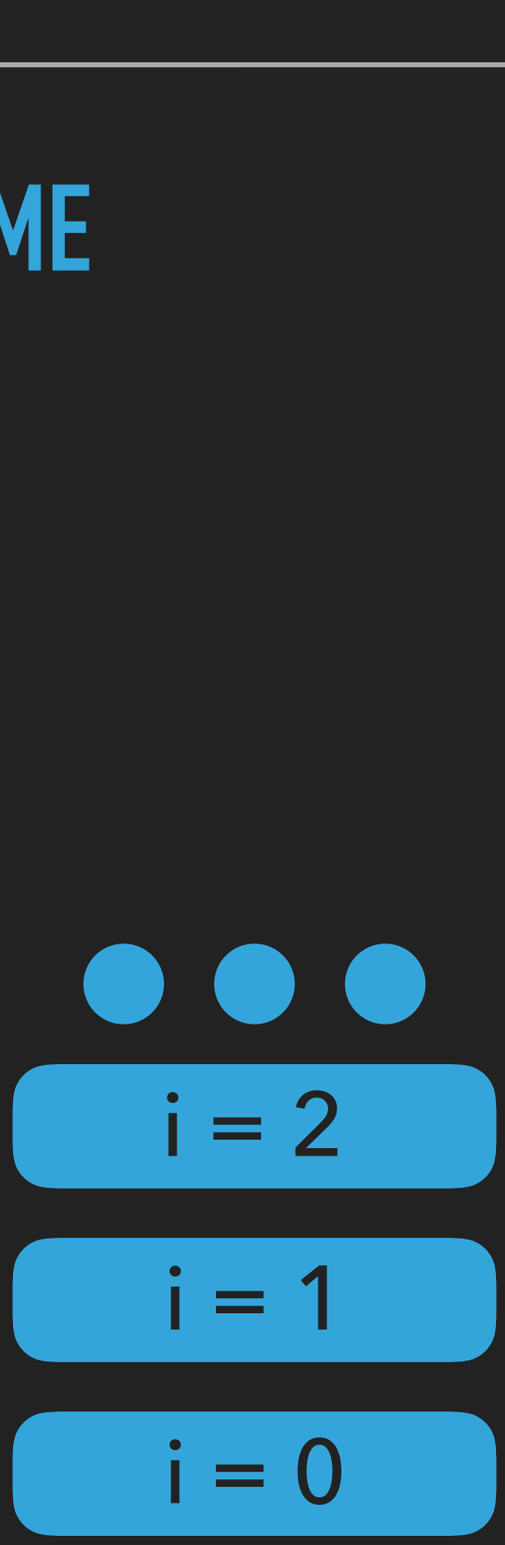
EVENT QUEUE

Run

THE BROWSER RUNTIME

Run Stop

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```



EVENT QUEUE

Stop

Run

THE BROWSER RUNTIME

Run Stop

```
function foo() {

    let i = 0;

    while (true) {

        i++;

    }

}

foo ();
```

i = 147

i = 2

i = 1

i = 0

EVENT QUEUE

Stop

Stop

Run


OCaml


Java


Dart


C++

MAYBE OTHER LANGUAGES AVOID THIS PROBLEM...

ALL INHERIT THE LIMITATIONS OF WEB BROWSERS!

JavaScript → JavaScript

STOPIFY

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Run

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Run

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

SAVING THE STACK AT RUNTIME

Run

Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

SAVING THE STACK AT RUNTIME

Run

Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Stop

Run

SAVING THE STACK AT RUNTIME

Run

Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Stop

Run

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Stop

Run

SAVING THE STACK AT RUNTIME

Run Stop

```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

Stop

SAVING THE STACK AT RUNTIME

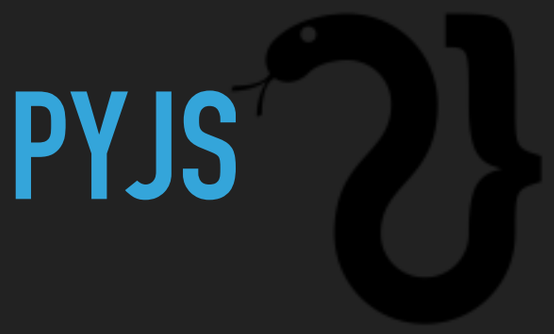
Run Stop

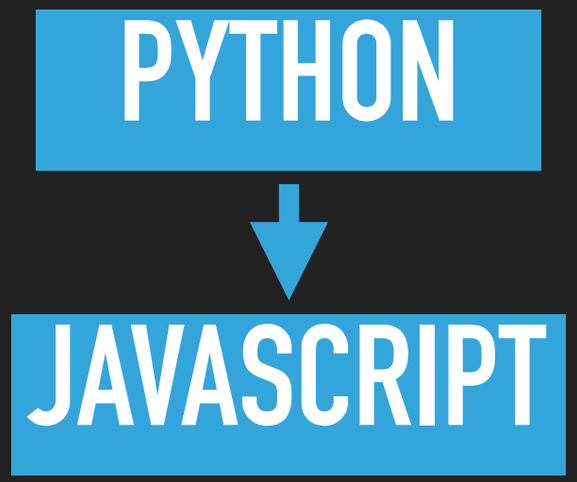
```
function odd(n) {
  if (n == 0) return false;
  else return even(n-1);
}
function even(n) {
  if (n == 0) return true;
  else return odd(n-1);
}
even(100000)
```

EVENT QUEUE

PYTHON

PYJS





PYJS

PYTHON

JAVASCRIPT

JAVASCRIPT+CC

PYJS

PYTHON

↓

JAVASCRIPT

↓

JAVASCRIPT+CC

↓

STOPIFY

PYJS

PYTHON

↓

JAVASCRIPT

↓

JAVASCRIPT+CC

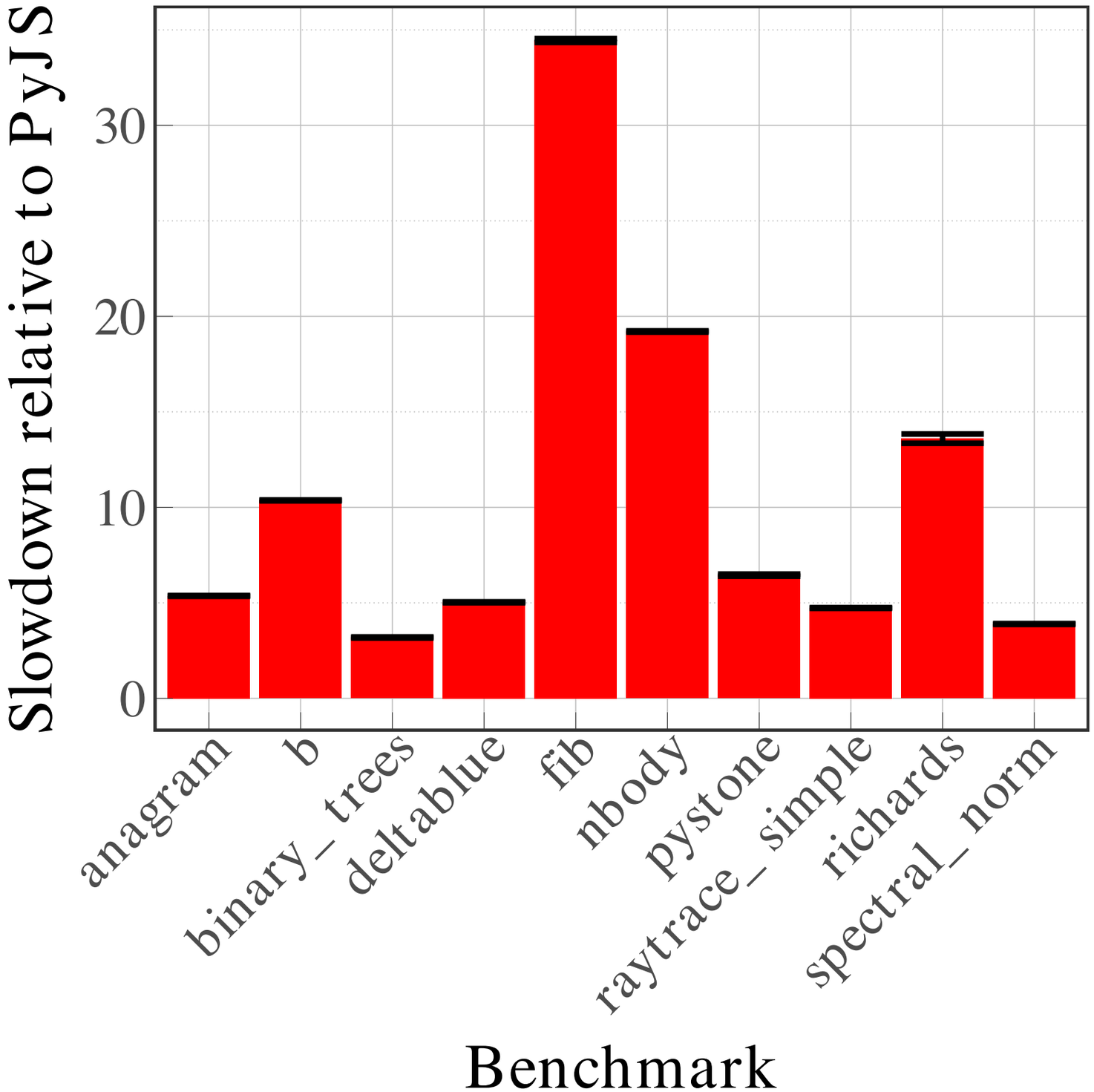
↓

STOPIFY

↓

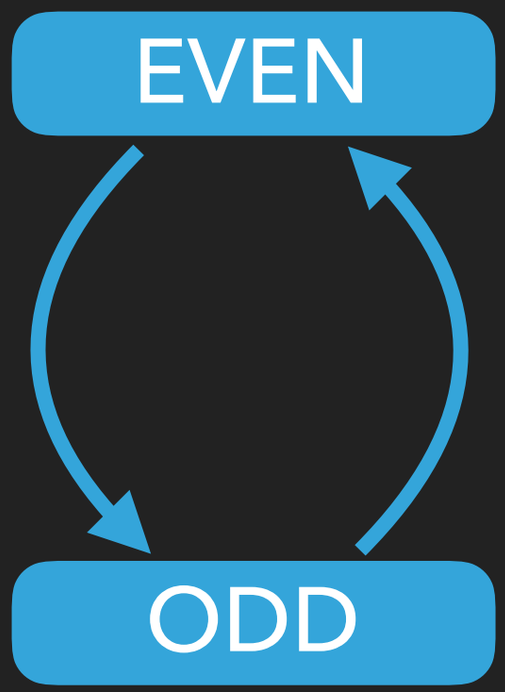
STOPPABLE JAVASCRIPT

UNFORTUNATELY, THIS IS SLOW



JAVASCRIPT CALL GRAPH

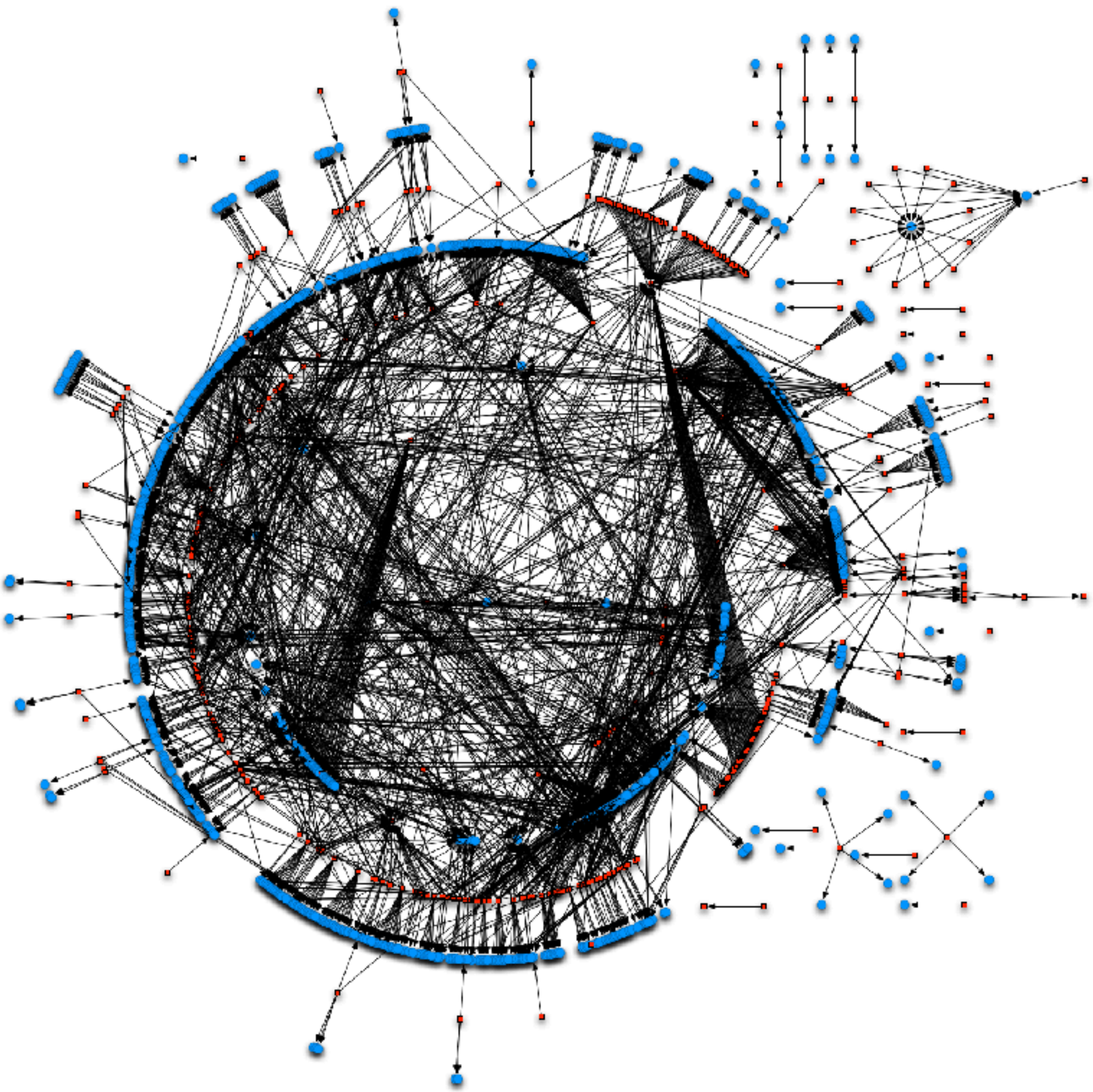
JAVASCRIPT CALL GRAPH



JAVASCRIPT CALL GRAPH

JAVASCRIPT CALL GRAPH

JAVASCRIPT CALL GRAPH



SPOT THE INFINITE LOOP

SPOT THE INFINITE LOOP

```
x + 1;
```

SPOT THE INFINITE LOOP

```
x + 1;
```

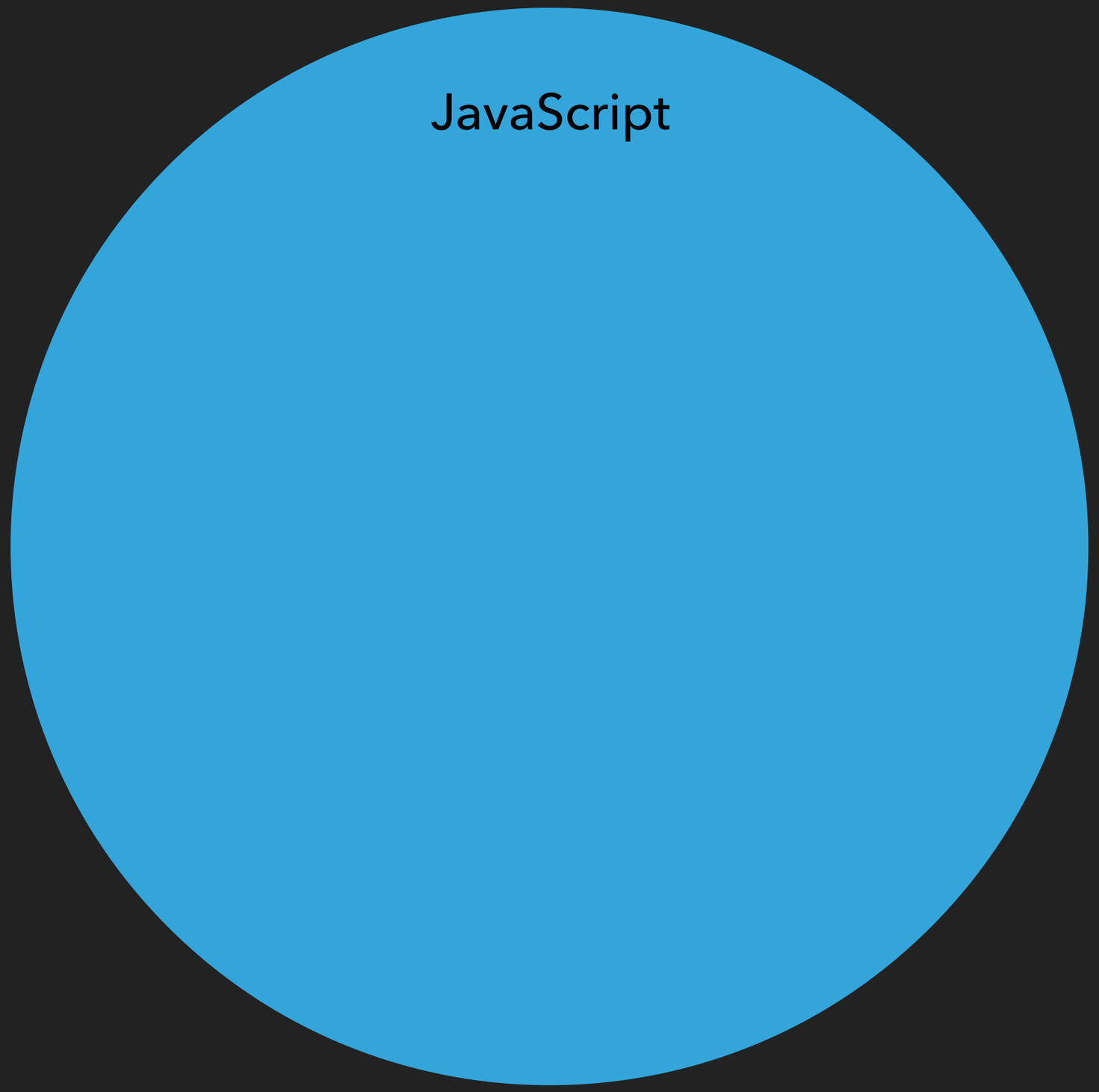
```
const x = {
  toString: function() {
    while (true) {}
  }
};
x + 1;
```

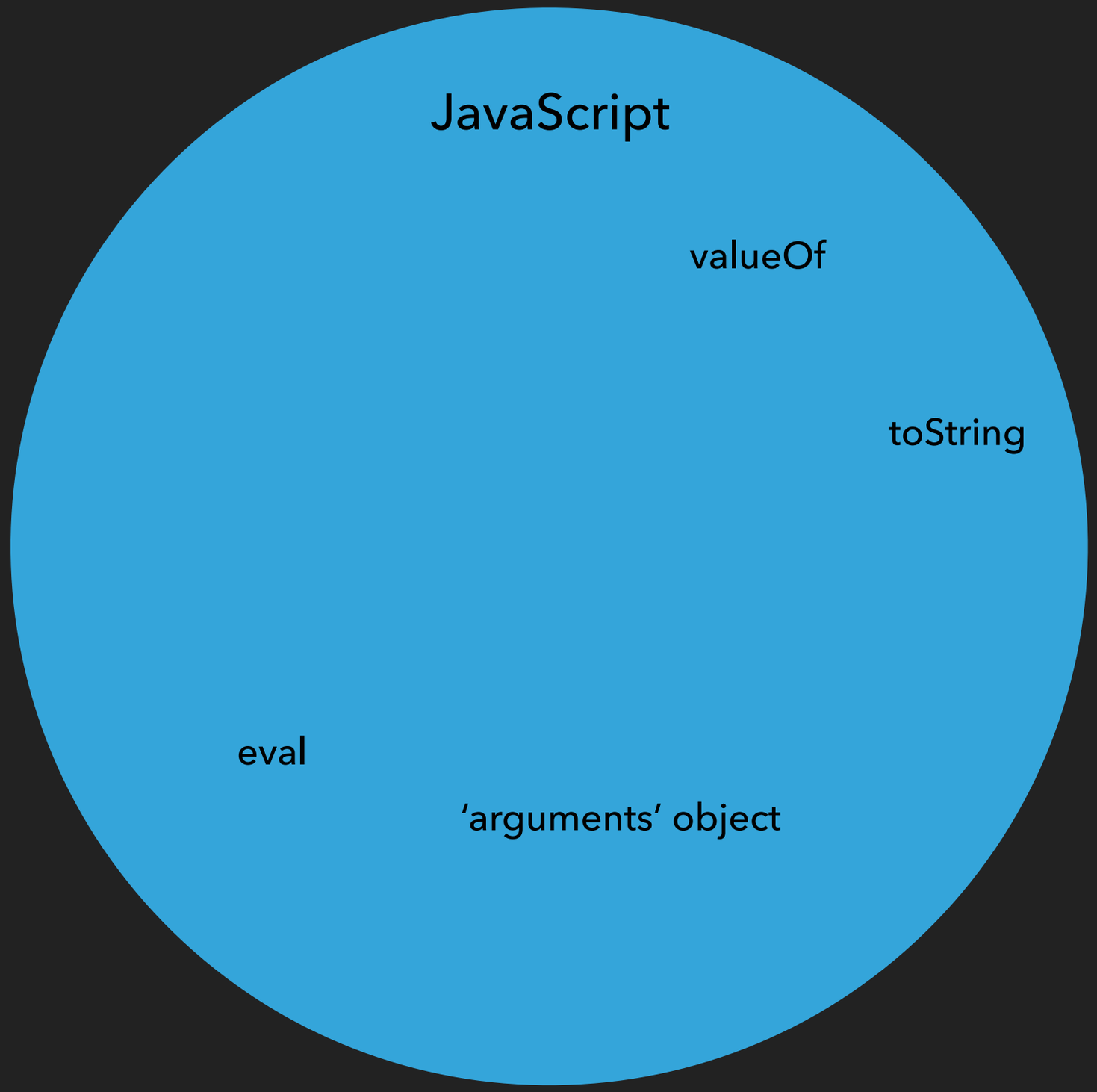
SPOT THE INFINITE LOOP

```
x + 1;
```

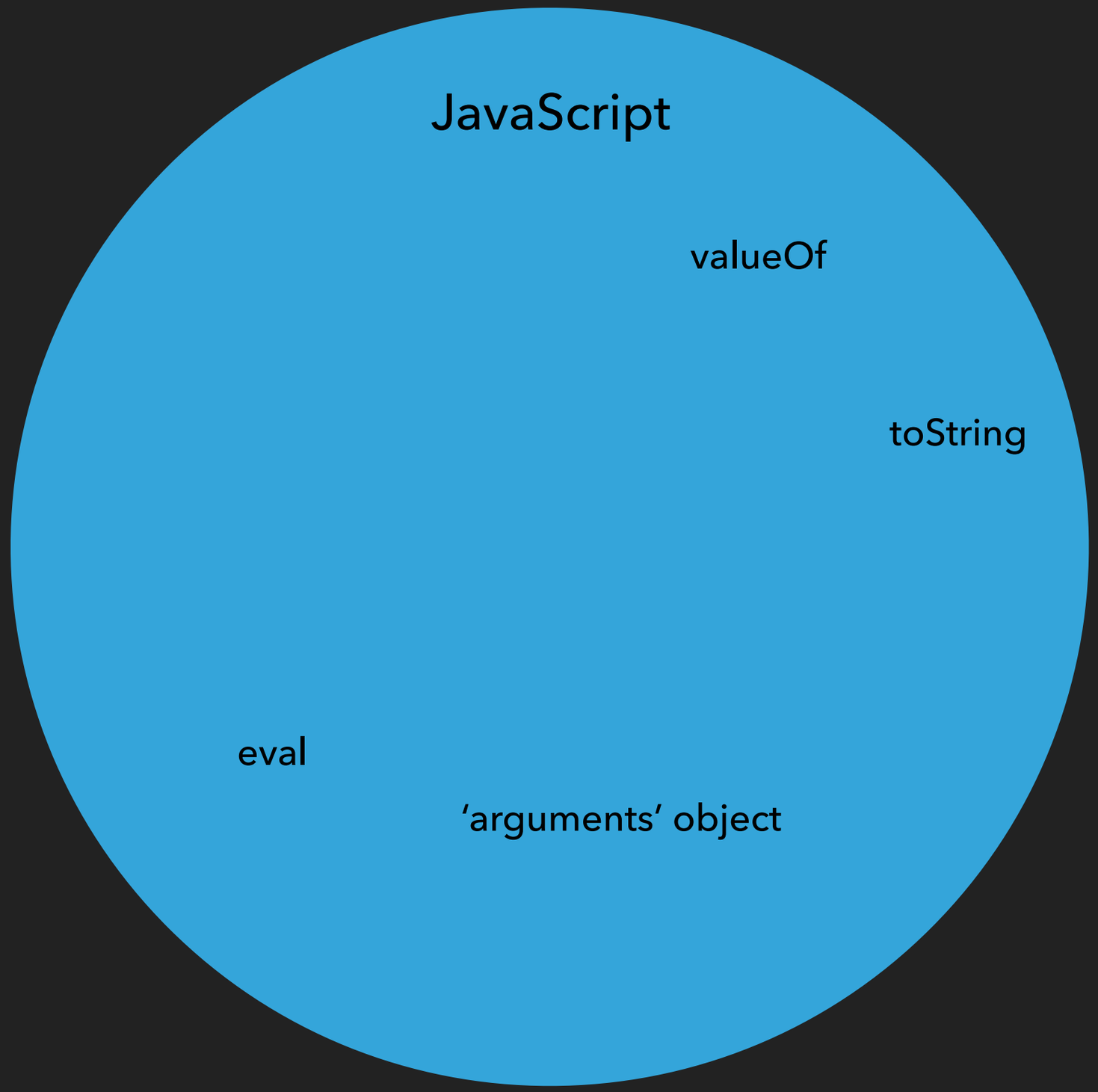
```
const x = {
  toString: function() {
    while (true) {}
  }
};
x + 1;
```

- Implicit Method Calls
  - ‘toString’ and ‘valueOf’
  - Getters and Setters
- Dynamic Code Execution (eval)


JavaScript


JavaScript
valueOf
toString
eval
‘arguments’ object

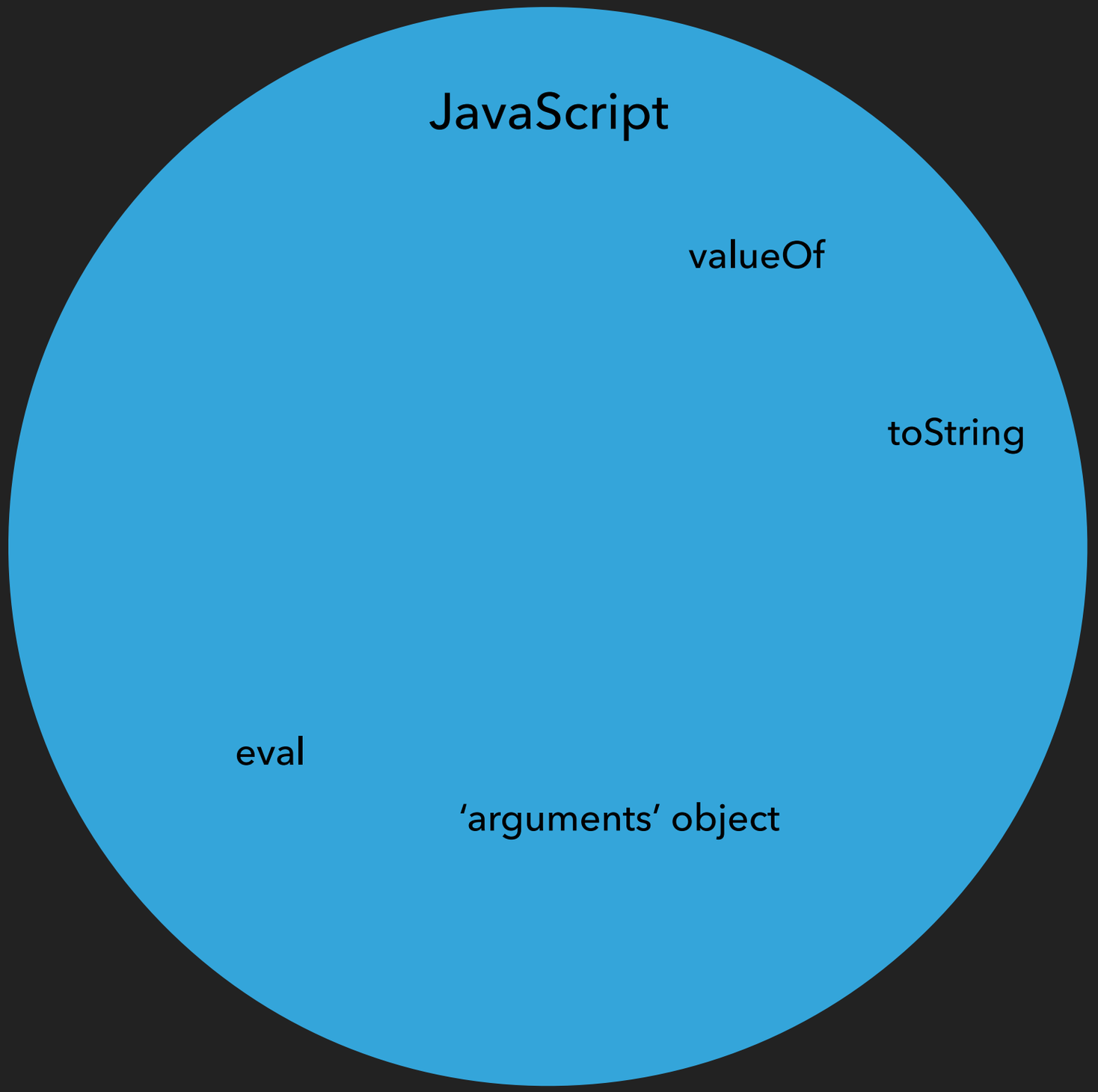
COMPILERS USE RESTRICTED SUB-LANGUAGES



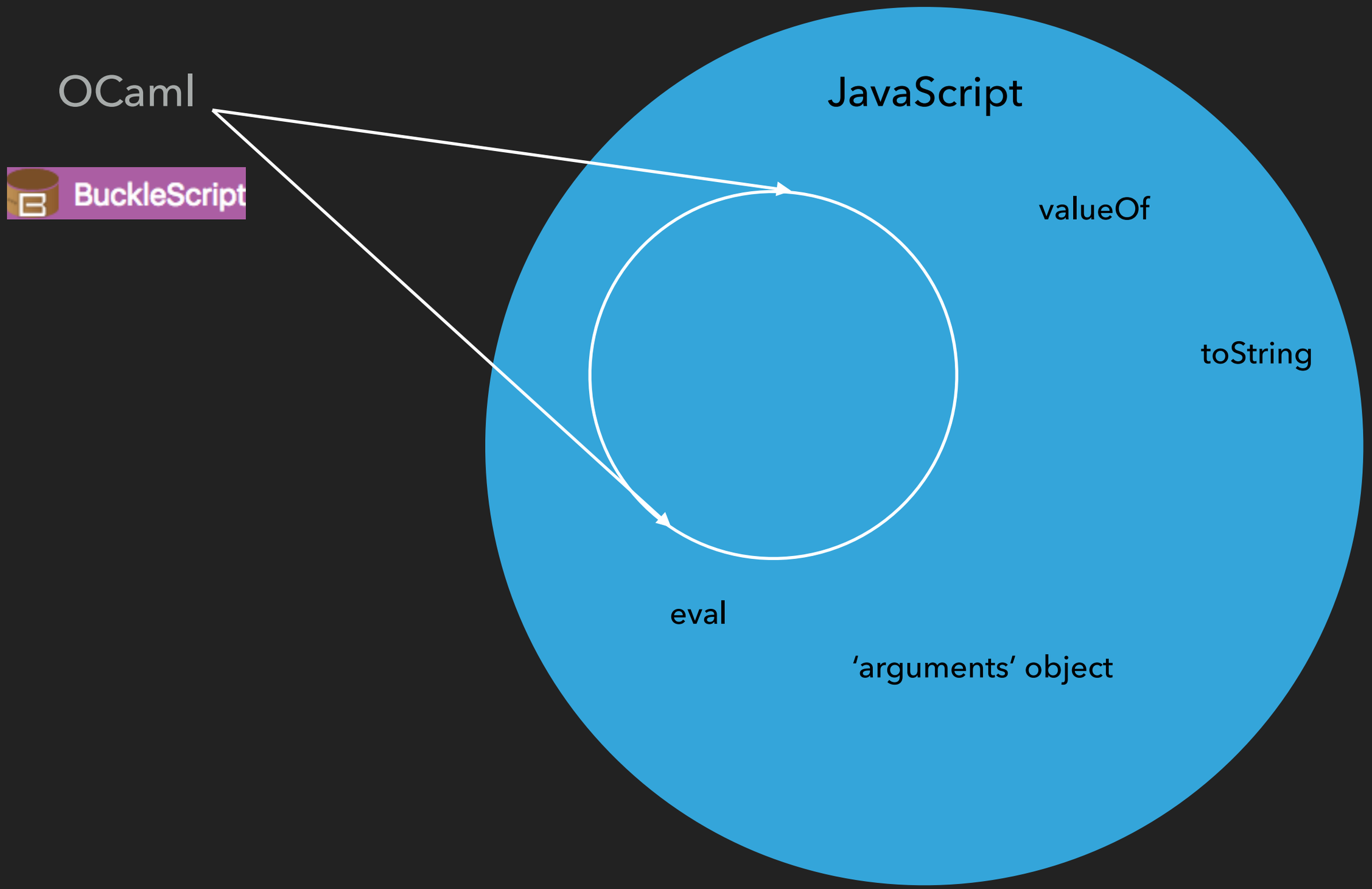
COMPILERS USE RESTRICTED SUB-LANGUAGES

OCaml

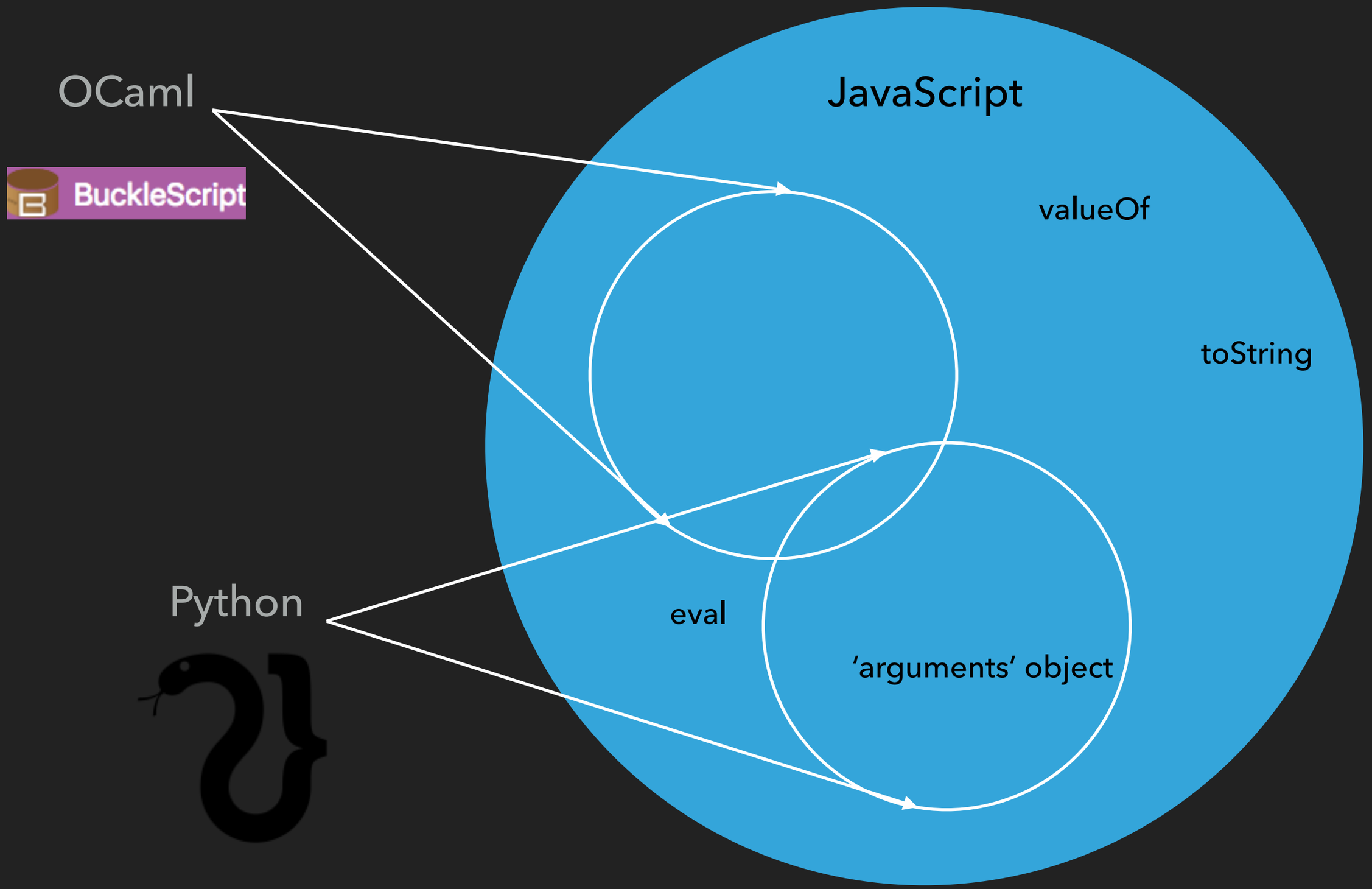
BuckleScript



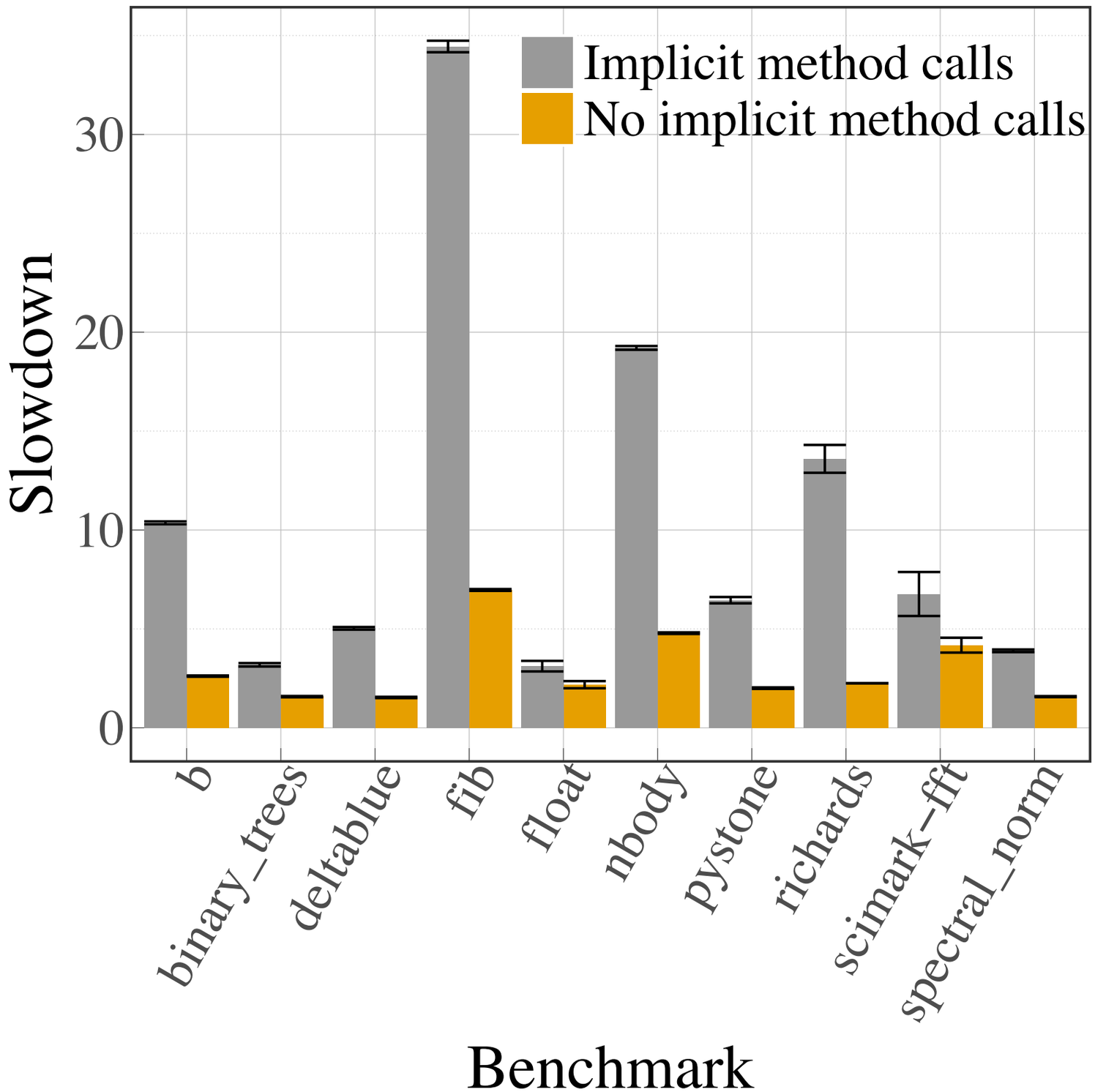
COMPILERS USE RESTRICTED SUB-LANGUAGES



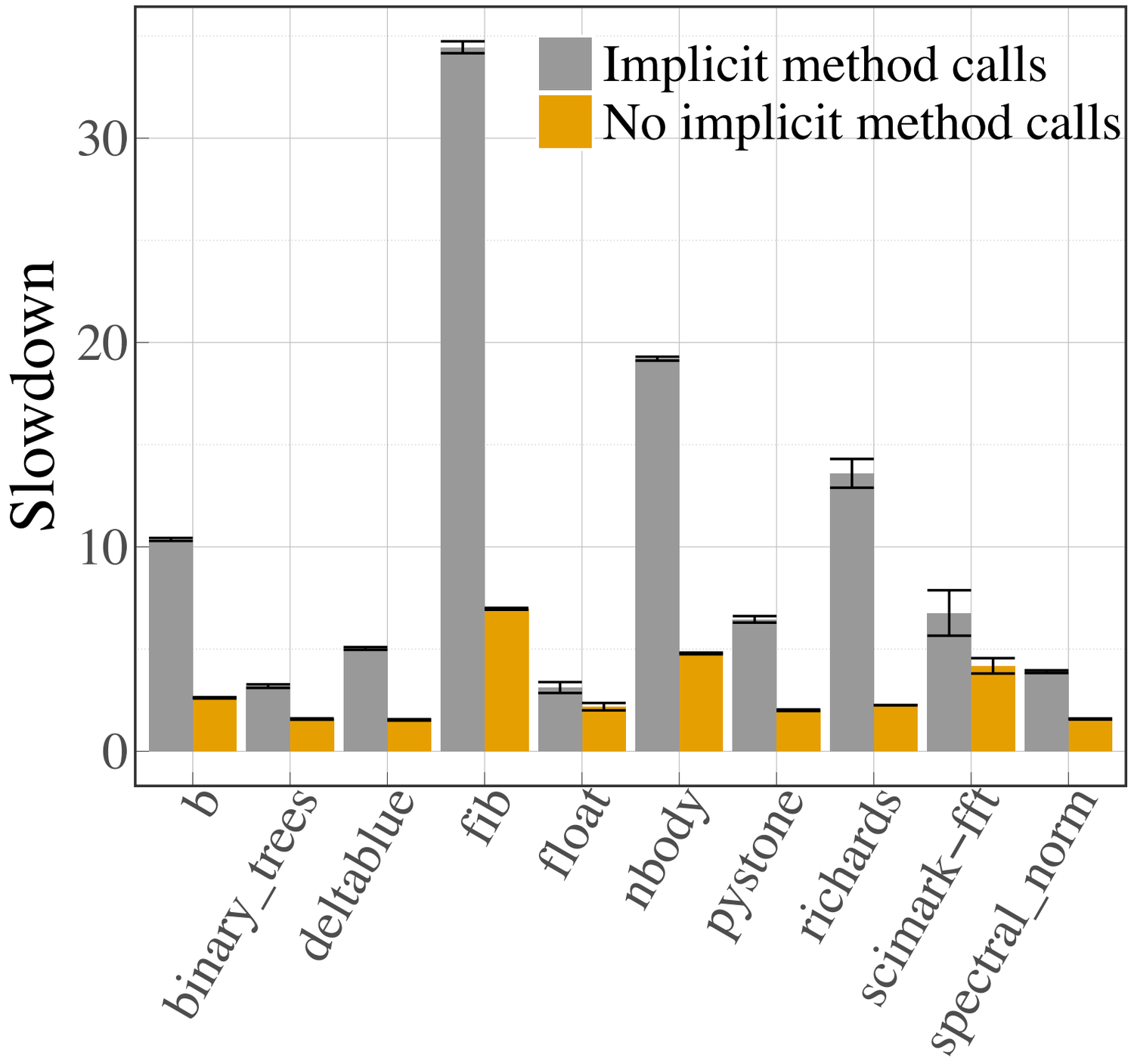
COMPILERS USE RESTRICTED SUB-LANGUAGES



IMPROVING PERFORMANCE WITH SAFE ASSUMPTIONS



IMPROVING PERFORMANCE WITH SAFE ASSUMPTIONS

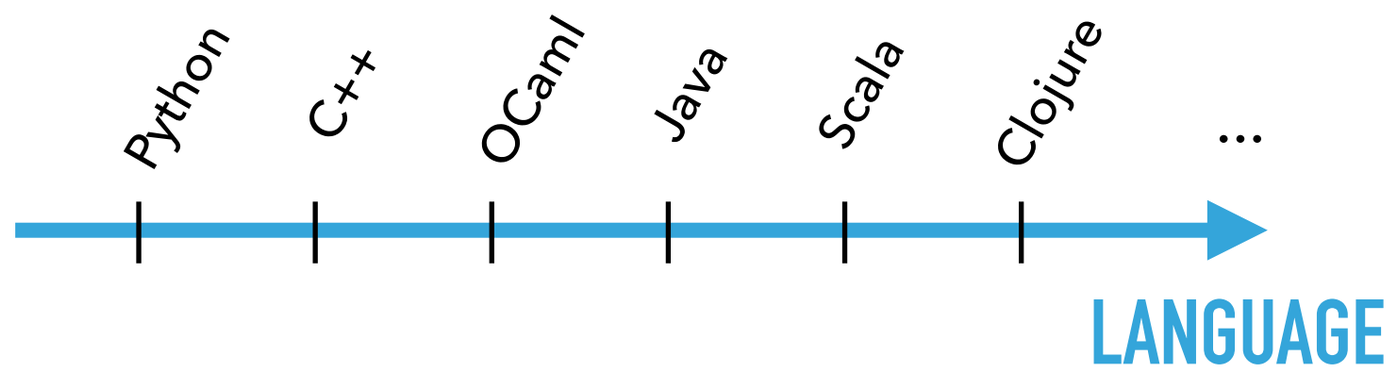


Median Slowdown

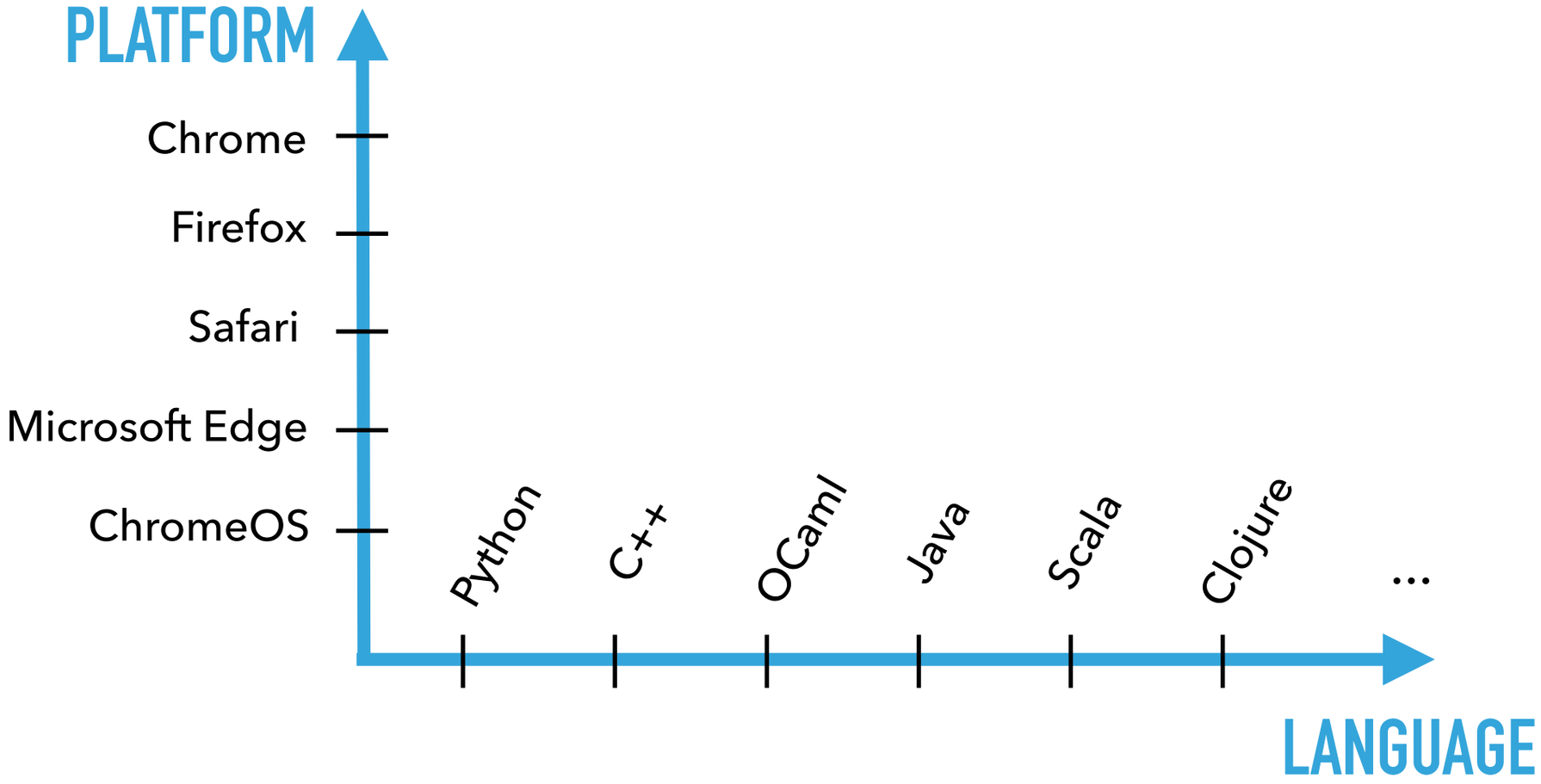
1.7x

OPTIMIZATIONS

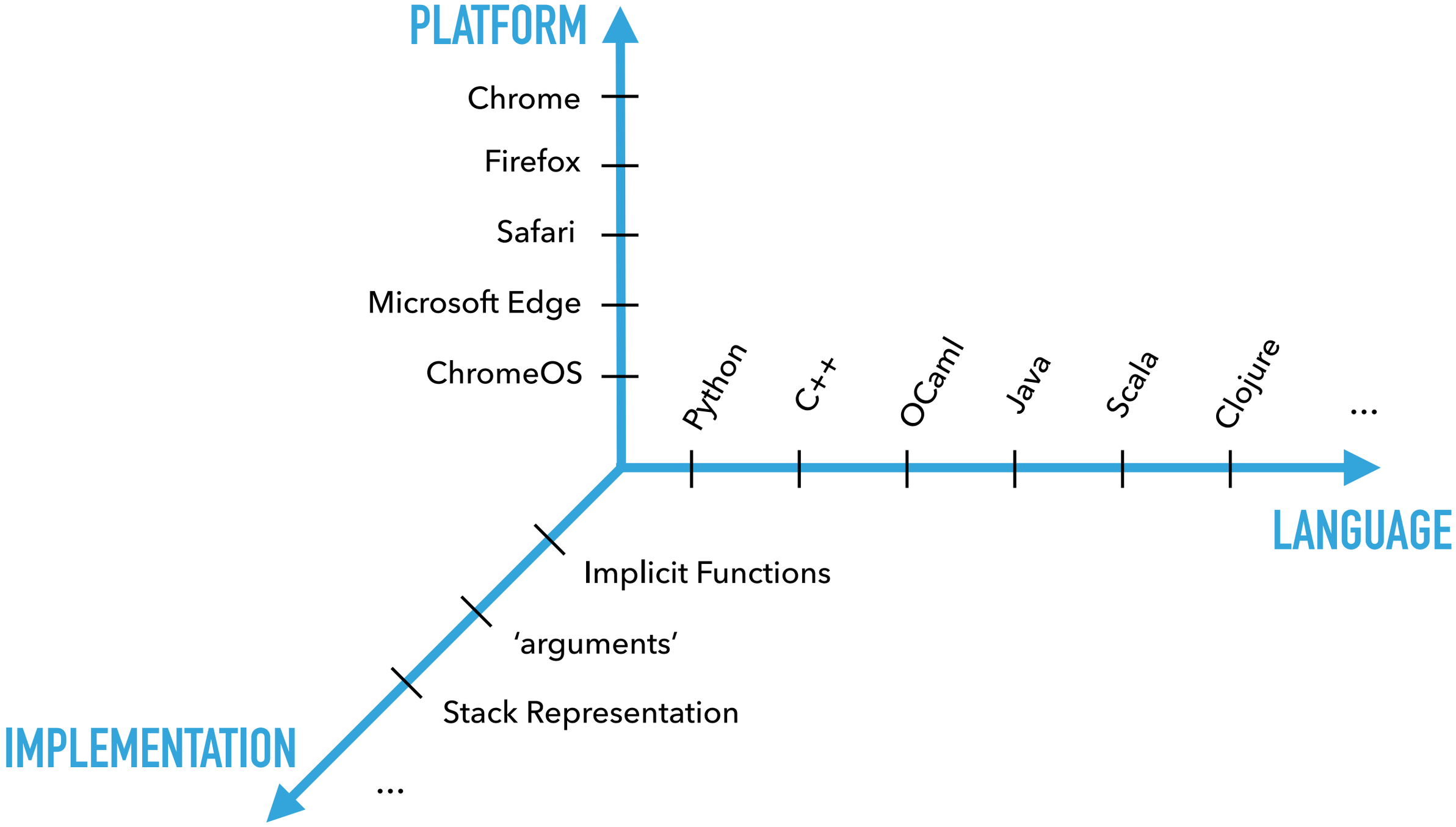
OPTIMIZATIONS



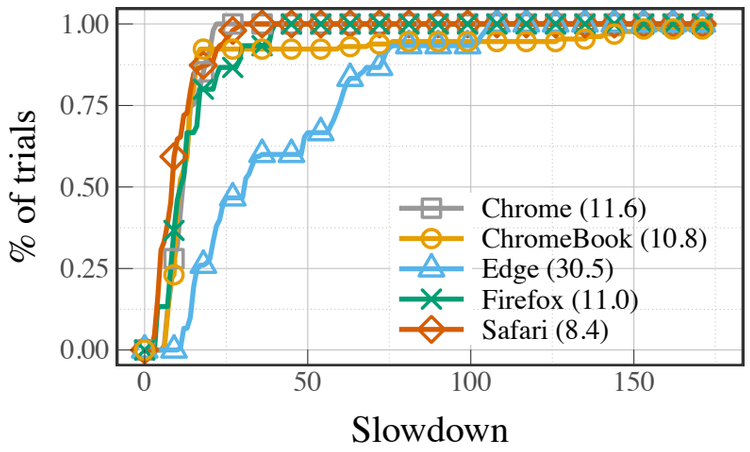
OPTIMIZATIONS



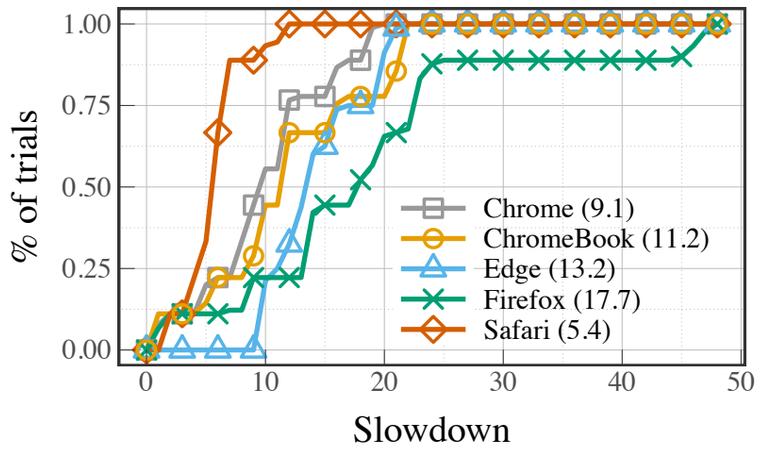
OPTIMIZATIONS



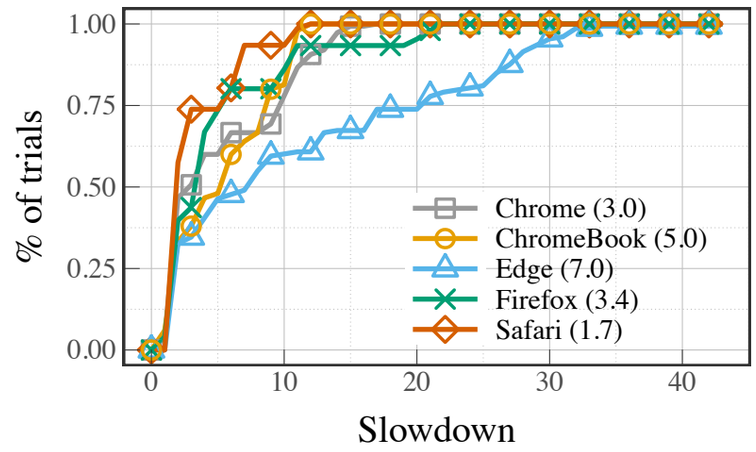
C++ (Emscripten)



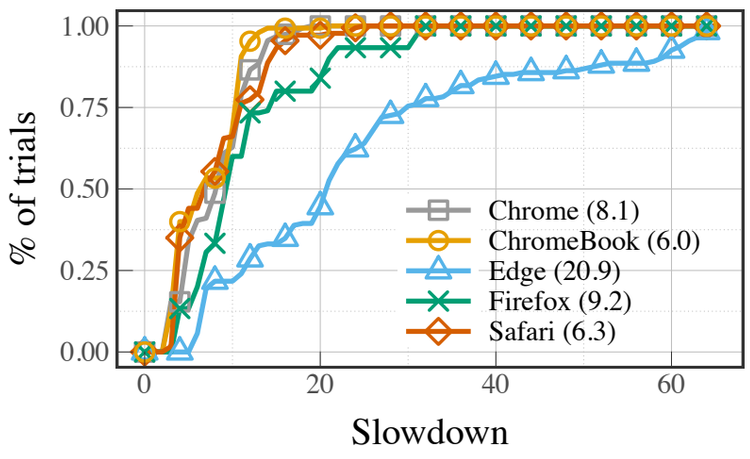
Clojure (ClojureScript)



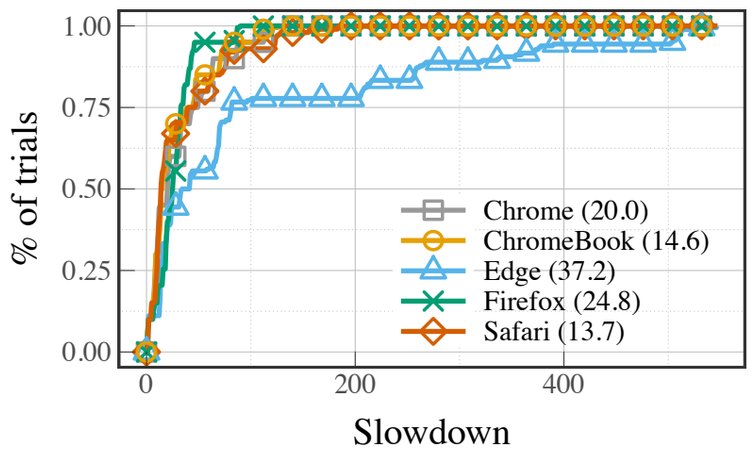
Dart (dart2js)



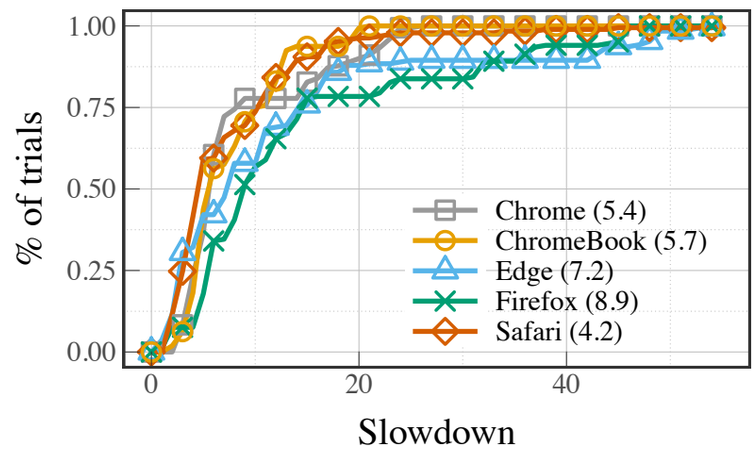
Java (JSweet)



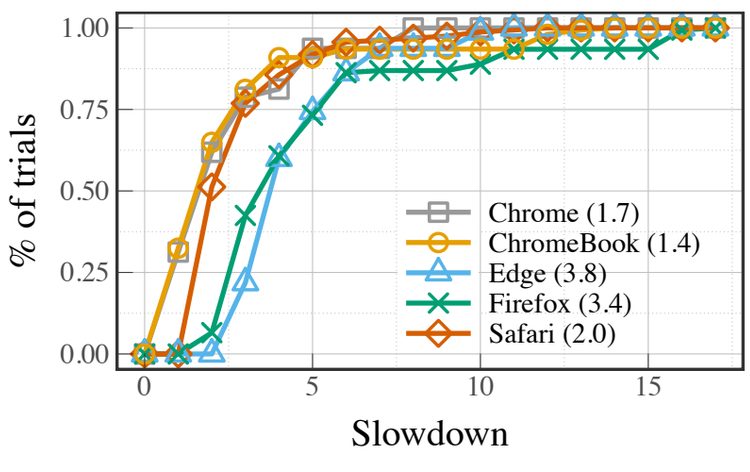
JavaScript



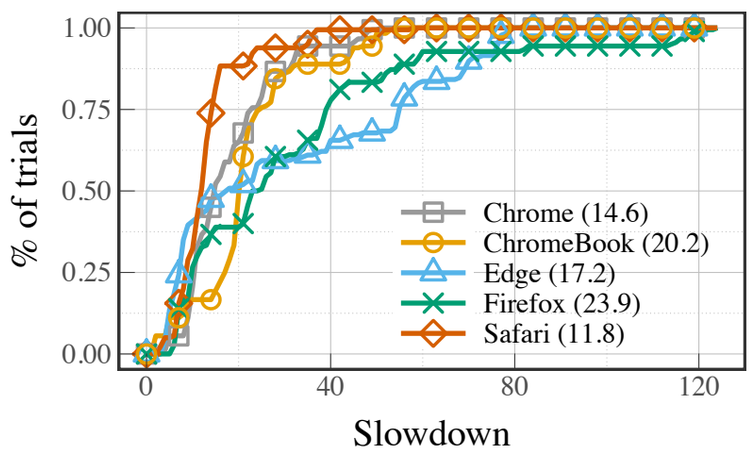
OCaml (BuckleScript)



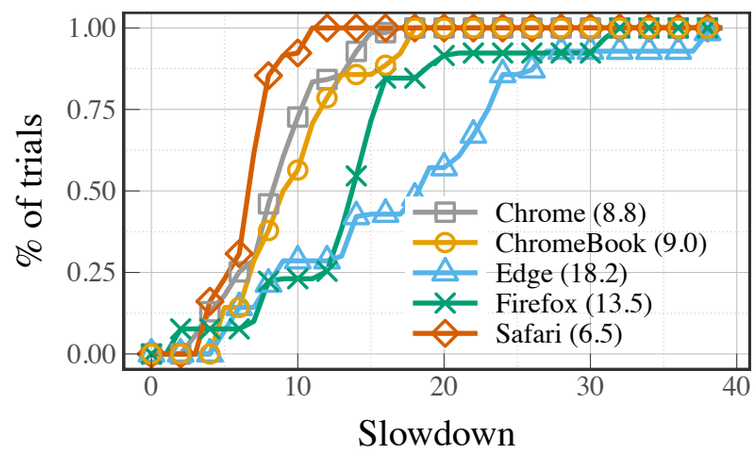
Python (PyJS)

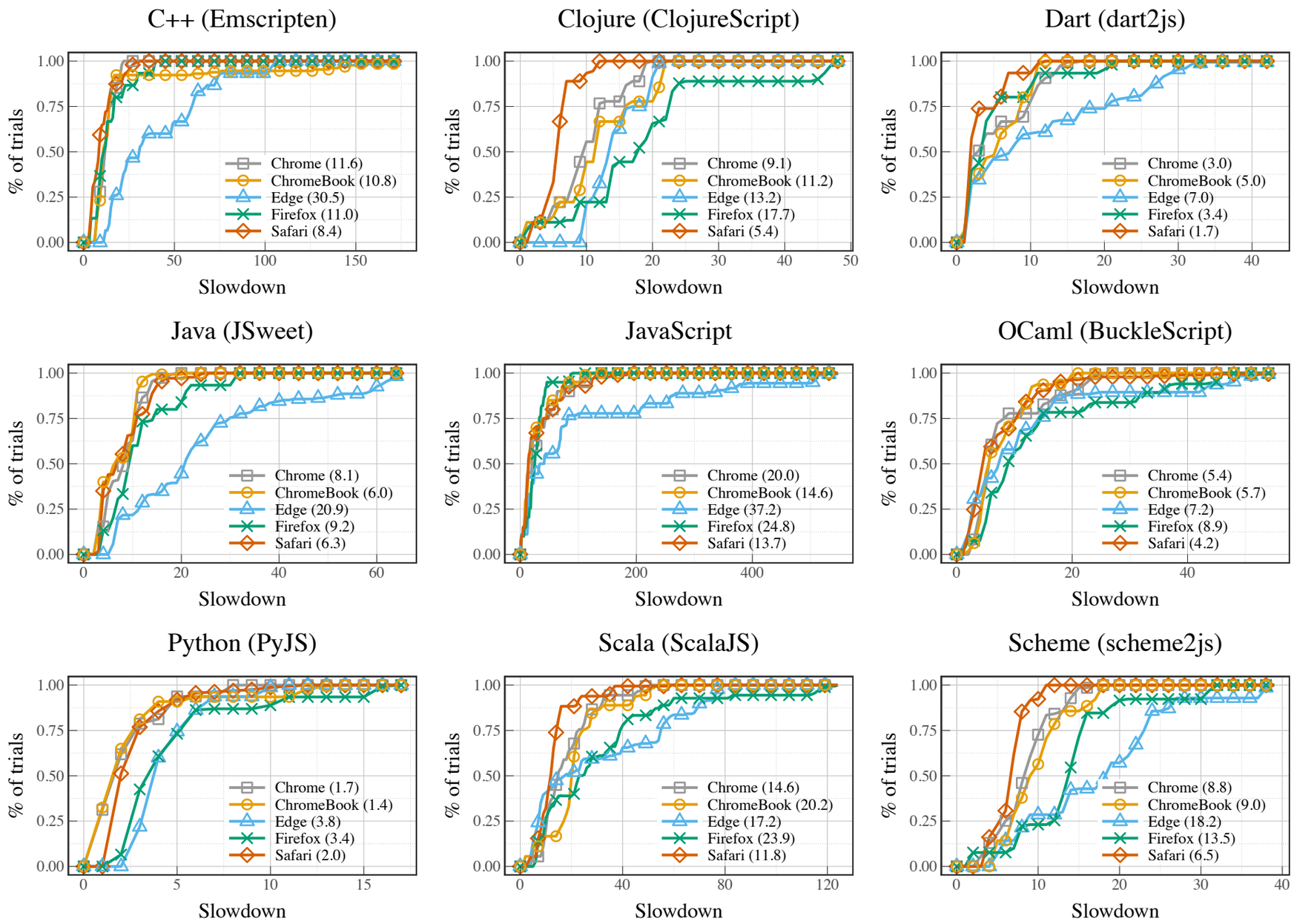


Scala (ScalaJS)



Scheme (scheme2js)




C++ (Emscripten)
% of trials
Slowdown
Chrome (11.6)
ChromeBook (10.8)
Edge (30.5)
Firefox (11.0)
Safari (8.4)
Clojure (ClojureScript)
Chrome (9.1)
ChromeBook (11.2)
Edge (13.2)
Firefox (17.7)
Safari (5.4)
Dart (dart2js)
Chrome (3.0)
ChromeBook (5.0)
Edge (7.0)
Firefox (3.4)
Safari (1.7)
Java (JSweet)
Chrome (8.1)
ChromeBook (6.0)
Edge (20.9)
Firefox (9.2)
Safari (6.3)
JavaScript
Chrome (20.0)
ChromeBook (14.6)
Edge (37.2)
Firefox (24.8)
Safari (13.7)
OCaml (BuckleScript)
Chrome (5.4)
ChromeBook (5.7)
Edge (7.2)
Firefox (8.9)
Safari (4.2)
Python (PyJS)
Chrome (1.7)
ChromeBook (1.4)
Edge (3.8)
Firefox (3.4)
Safari (2.0)
Scala (ScalaJS)
Chrome (14.6)
ChromeBook (20.2)
Edge (17.2)
Firefox (23.9)
Safari (11.8)
Scheme (scheme2js)
Chrome (8.8)
ChromeBook (9.0)
Edge (18.2)
Firefox (13.5)
Safari (6.5)

```
#include <stdio.h>

int factorial(int n) {
  auto result = 1;
  for (int i = 1; i <= n; i++) {
    result *= i;
    printf("intermediate fact(%d): %d\n", i, result);
  }

  return result;
}

int main() {
  auto n = 6;
  printf("Factorial(%d): %d\n", n, factorial(n));
}
```

```
intermediate fact(1): 1
intermediate fact(2): 2
intermediate fact(3): 6
intermediate fact(4): 24
intermediate fact(5): 120
```

- Sam Baxter
- Rachit Nigam
- Arjun Guha
- Joe Politz
- Shriram Krishnamurthi

STOPIFY

http://www.stopify.org

WRESTLING CONTROL FROM
WEB BROWSERS