an hour of

node.js

install node.js: http://nodejs.org
download the workshop: http://tiny.cc/node
say hi to your neighbors

why node?

Reason 1

it’s fast!

“CPU utilization was hovering around 1%

… **developers were bored**”

–Walmart on Black Friday

Reason 2

it’s fast! (to write)

"it took **fewer** developers **half the time** to reach feature parity with the Java app"

–Paypal

“twice as fast using one core
…than when Java was using five”

–Paypal

is node faster than java?

a clarification

no.

async APIs are slower
…but total program performance may improve!

63,063 modules!

Act 1

common interview questions in js

follow along!

open the act1 folder
run the examples with `node examples/<filename>`
or just use the node REPL: `node`

reverse.js

```
var list = ['a', 'b', 3, 4, 5];

console.log(list.reverse());
// => [ 5, 4, 3, 'b', 'a' ]
```

palindrome.js

```
function palindrome (word) {
  return word.split('').reverse().join('') == word;
}

console.log(palindrome('racecar')); // => true
console.log(palindrome('cat')); // => false
```

fizzbuzz.js

```
for(var i=1, ii=100; i<=ii; ++i) {
  var three = i%3 === 0
    , five = i%5 === 0
    , both = three && five;

  if(both)
    console.log('FizzBuzz');
  else if(three)
    console.log('Fizz');
  else if(five)
    console.log('Buzz');
  else
    console.log(i);
}
```

your turn!

open exercises/missingno.js

to test: `node check.js`

here is cheatsheet.js:

- variables: `**var** i, b='meow', c=['array']`
- functions: `**function** cat() { **return** 'meow'; }`
- loops: `**for**(;;){ }` `**while**(false){ }`
- output: `console.log('taco');`
- arrays: `[1, 2, 3].length` `[1, 2, 3].sort()`

count.js

```
var i = 0
  , count;

count = function () {
  return ++i;
};

console.log(count()); // => 1
console.log(count()); // => 2
console.log(count()); // => 3
```

counter.js

```
var makeCounter;

makeCounter = function () {
  var i = 0;

  return function () {
    return ++i;
  };
};

var a = makeCounter()
  , b = makeCounter();

console.log(a()); // => 1
console.log(b()); // => 1
console.log(a()); // => 2
```

your turn!

beginner: exercises/sleepsort.js

advanced: exercises/inception.js

to test: `node check.js`