A BROWSE THROUGH ES2016

A BROWSE THROUGH ES2016

Jez Higgins

A BROWSE THROUGH ES6

- A Sprint Through ES6
- An ES6 Spike

YOU SHOULD LEAVE IF …

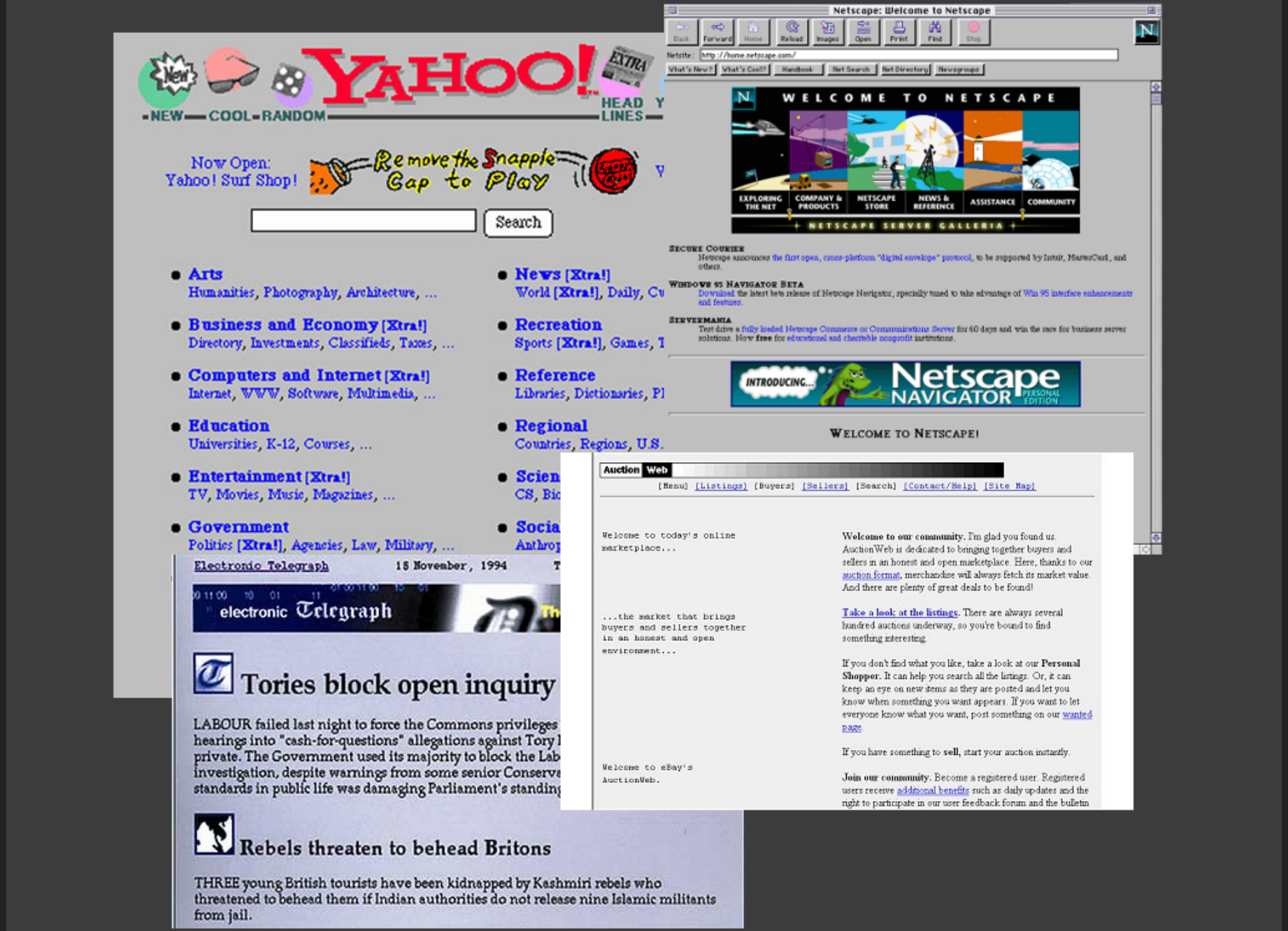
- JavaScript isn’t your thing
- JavaScript is your thing and you’re ES6ed up already

PERHAPS STAY IF …

- JavaScript is your thing

HOW DID WE GET HERE?

HOW DID WE GET HERE?


YAHOO!
NEW
COOL
RANDOM
HEAD LINES
Now Open:
Yahoo! Surf Shop!
Remove the Snapple Cap to Play
Search
Arts
Humanities, Photography, Architecture, ...
Business and Economy [Xtra!]
Directory, Investments, Classifieds, Taxes, ...
Computers and Internet [Xtra!]
Internet, WWW, Software, Multimedia, ...
Education
Universities, K-12, Courses, ...
Entertainment [Xtra!]
TV, Movies, Music, Magazines, ...
Government
Politics [Xtra!], Agencies, Law, Military, ...
News [Xtra!]
World [Xtra!], Daily, Cu
Recreation
Sports [Xtra!], Games, T
Reference
Libraries, Dictionaries, P
Regional
Countries, Regions, U.S.
Scien
CS, Bio
Socia
Anthro
Netscape: Welcome to Netscape
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Netsite: http://home.netscape.com/
What's New?
What's Cool?
Handbook
Net Search
Net Directory
Newsgroups
WELCOME TO NETSCAPE
EXPLORING THE NET
COMPANY & PRODUCTS
NETSCAPE STORE
NEWS & REFERENCE
ASSISTANCE
COMMUNITY
NETSCAPE SERVER GALLERIA
SECURE COURIER
Netscape announces the first open, cross-platform "digital envelope" protocol, to be supported by Intuit, MasterCard, and others.
WINDOWS 95 NAVIGATOR BETA
Download the latest beta release of Netscape Navigator, specially tuned to take advantage of Win 95 interface enhancements and features.
SERVERMANIA
Test drive a fully loaded Netscape Commerce or Communications Server for 60 days and win the race for business server solutions. Now free for educational and charitable nonprofit institutions.
INTRODUCING...
Netscape NAVIGATOR PERSONAL EDITION
WELCOME TO NETSCAPE!
Electronic Telegraph
15 November, 1994
electronic Telegraph
Tories block open inquiry
LABOUR failed last night to force the Commons privileges
hearings into "cash-for-questions" allegations against Tory
private. The Government used its majority to block the Lab
investigation, despite warnings from some senior Conserve
standards in public life was damaging Parliament's standing
Rebels threaten to behead Britons
THREE young British tourists have been kidnapped by Kashmiri rebels who threatened to behead them if Indian authorities do not release nine Islamic militants from jail.
Auction Web
[Menu] [Listings] [Buyers] [Sellers] [Search] [Contact/Help] [Site Map]
Welcome to today's online marketplace...
...the market that brings buyers and sellers together in an honest and open environment...
Welcome to eBay's AuctionWeb.
Welcome to our community. I'm glad you found us. AuctionWeb is dedicated to bringing together buyers and sellers in an honest and open marketplace. Here, thanks to our auction format, merchandise will always fetch its market value. And there are plenty of great deals to be found!
Take a look at the listings. There are always several hundred auctions underway, so you're bound to find something interesting.
If you don't find what you like, take a look at our Personal Shopper. It can help you search all the listings. Or, it can keep an eye on new items as they are posted and let you know when something you want appears. If you want to let everyone know what you want, post something on our wanted page.
If you have something to sell, start your auction instantly.
Join our community. Become a registered user. Registered users receive additional benefits such as daily updates and the right to participate in our user feedback forum and the bulletin

HOW DID WE GET HERE?

HOW DID WE GET HERE?

HOW DID WE GET HERE? BY ACCIDENT

London Supercloud @pikesley · Feb 10

JavaScript, a language designed by somebody who has only ever had other languages described to them but has never actually seen or used one

allan kelly @allankellynet · Mar 2

Controversial: "I don't like JavaScript, its not really a language"
@aahoogendoorn @TopconfAT

1 3

JAVASCRIPT OOPSYS

Weird scoping - Wacky type conversion - Numbers are IEEE 754 double precision floats, but you can do bitwise operations on them as if they were 32 bit integers - Everything's visible - Functions do triple duty - What is *this*? - Semicolon insertion - Prototype inheritance - Wait? Did you just go global? - Equality…

EQUALS, RIGHT

```
If Type(x) is the same as Type(y), then
    If Type(x) is Undefined, return true.
    If Type(x) is Null, return true.
    If Type(x) is Number, then
        If x is NaN, return false.
        If y is NaN, return false.
        If x is the same Number value as y, return true.
        If x is +0 and y is −0, return true.
        If x is −0 and y is +0, return true.
        Return false.
    If Type(x) is String, then return true if x and y are exactly
        the same sequence of characters (same length and same charact
        in corresponding positions).
        Otherwise, return false.
    If Type(x) is Boolean, return true if x and y are both true or
          both false. Otherwise, return false.
```

BUT NOW WE HAVE ES6

- ES6 fixes none of this
- All our crappy JavaScript will continue to run
- Our new JavaScript can be less crappy

BLOCK SCOPE

```
{
  let x = 0;
  // ---
}
```

BLOCK SCOPE

```
{
  let x = 0;
  // ---
}
console.log(x); // ERROR!
```

```
{
  console.log(x); // ERROR
  // ---
  let x = 0;
  // ---
}
```

BLOCK SCOPE

```
{
  let x = 0;
  // ---
}
```

BLOCK SCOPE

```
let x = 0;
// ---
{
  let x = "something else";
  // ---
}
// ---
```

BLOCK SCOPE

```
let x = "Hello";

const y = "World";

x = "Goodbye";
```

BLOCK SCOPE

```
let x = "Hello";

const y = "World";

x = "Goodbye";

y = "Cruel World";  // ERROR
```

FOR LOOPS

```
for (let i = 0; i != 10; ++i) {
  // ---
}
// ---
```

FOR LOOPS

```
const arr = [2, 4, 6, 8, 10];

for (let i in arr) {
  let v = arr[i];
  // ---
}
```

FOR LOOPS

```
const arr = [2, 4, 6, 8, 10];

for (const i in arr) {
  let v = arr[i];
  // ---
}
```

FOR LOOPS

```
const arr = [2, 4, 6, 8, 10];

for (const v of arr) {
  // ---
}
```

FOR LOOPS

```
const arr = [2, 4, 6, 8, 10];

for (const [i, v] of arr.entries()) {
  // ---
}
```

ITERATORS

```
let m = new Map();
m.set("hat", "Red");
m.set("flavour", "Mint");

for (const [k, v] of m) {
  // ---
}
```

ITERATORS

```
const twoTimesTable = {
  [Symbol.iterator]: function() {
    let n = 1;
    return {
      next: function() {
        if (n > 10) return { done: true };
        const p = 2*n;
        ++n;
        return { done: false, value: p };
      }
    }
  }
}

for (const p of twoTimesTable)
  console.log(p);
```

GENERATORS

```
const twoTimesTable = {
  [Symbol.iterator]: function*() {
    for (let n = 1; n <= 10; ++n)
      yield 2*n;
  }
}

for (const p of twoTimesTable)
  console.log(p);
```

GENERATORS

```
const twoTimesTable = {
  *[Symbol.iterator]() {
    for (let n = 1; n <= 10; ++n)
      yield 2*n;
  }
}

for (const p of twoTimesTable)
  console.log(p);
```

GENERATORS

```
function* twoTimesTable() {
  for (let n = 1; n <= 10; ++n)
    yield 2*n;
}

for (const p of twoTimesTable())
  console.log(p);
```

STAND ASIDE LODASH

```
const arr = [1,2,3,4,5, ........., n];

const first_even_number =
    from(arr).filter(n => n%2==0).first();
```

STAND ASIDE LODASH

```
function* randomInterval(min, max) {
    for(;;)
        yield Math.floor(Math.random()*(max-min+1)+min);
}

const first_even_number =
    from(randomInterval(50, 150)).filter(n => n%2==0).first();
```

STAND ASIDE LODASH (@ROBSMALLSHIRE REMIX)

```
function* lucas_sequence() {
    let a = 2, b = 1;
    yield a;
    while (true) {
        yield b;
        [a, b] = [b, a + b];
    }
} // lucas_sequence

const first_six_digit_lucas_number =
    from(lucas_sequence()).filter(n => n.toString().length == 6).firs
```

COROUTINES ALA SMALLSHIRE

```
function* async_search(iterable, async_predicate) {
    for (const item of iterable)
        if (yield* async_predicate(item))
            return item;
    return null;
} // async_search

function* lucas_sequence() {
    let a = 2, b = 1;
    yield a;
    while (true) {
        yield b;
        [a, b] = [b, a + b];
    }
} // lucas_sequence
```

DESTRUCTURING

```
const arr = [2, 4, 6, 8, 10];

for (const [i, v] of arr.entries()) {
  //---
}
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ first, second, third ] = hand;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ first ] = hand;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ first ] = hand;

const [, , third ] = hand;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ first ] = hand;

const { suit: s, card: c } = first;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ first ] = hand;

const { suit, card } = first;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const [ { suit, card } ] = hand;
```

DESTRUCTURING

```
const hand = [ { suit: 'Spades', card: 'Ace' },
               { suit: 'Hearts', card: 'Jack' },
               { suit: 'Clubs',  card: 'Ten' } ];

const { length: len } = hand;
```

DESTRUCTURING

```
const [w,x,y,z] = "abcdef";
```

DESTRUCTURING

```
const [head, ...tail] = "abcdef";
```

DESTRUCTURING

```
const [head, ...tail] = "abcdef";

// head = 'a'
// tail = ['b','c','d','e','f'];
```

DESTRUCTURING

```
const { code, msg = 'No error message available' } = response
```

PARAMETERS

```
function engageWarpSpeed(factor = max()) {
  // ---
}
```

PARAMETERS

```
function engageWarpSpeed(factor = max()) {
  // ---
}

engageWarpSpeed(3);
```

PARAMETERS

```
function engageWarpSpeed(factor = max()) {
  // ---
}

engageWarpSpeed(3);

engageWarpSpeed();
```

PARAMETERS

```
function format(pattern, ...params) {
  // ---
}
```

PARAMETERS

```
function format(pattern, ...params) {
  // ---
}

format("Boring string");

format("%d, %d", 1, 2);
```

PARAMETERS

```
function waggleArm({ extension: 0, rotation: 180, speed: 1 } = {}) {
  // ---
}
```

PARAMETERS

```
function waggleArm({ extension: 0, rotation: 180, speed: 1 } = {}) {
  // ---
}

waggleArm({ extension: 100, rotation: 90, speed: 5 });

waggleArm({ rotation: 270 });

waggleArm({});

waggleArm();
```

PARAMETERS

```
let numbers = [35, 39, 11, 9];

const smallest = Math.min(...numbers);
```

PARAMETERS

```
let numbers = [35, 39, 11, 9];

const smallest = Math.min(...numbers);

const alsoSmallest = Math.min(35, 39, 11, 9);
```

ARROW FUNCTIONS

```
const arr = [1, 2, 3];

const cubes = arr.map(x => x*x*x);
```

ARROW FUNCTIONS

```
const arr = [1, 2, 3];

const cubes = arr.map(x => x*x*x);

cubes.forEach(x => { console.log(x); });
```

ARROW FUNCTIONS

```
function Multiplier(factor) { this.factor = factor; }

Multiplier.prototype.multiplyArray = function(arr) {
  var that = this;
  return arr.map(function (x) {
    return that.factor * x;
  });
};
```

ARROW FUNCTIONS

```
function Multiplier(factor) { this.factor = factor; }

Multiplier.prototype.multiplyArray = function(arr) {
  return arr.map(function (x) {
    return this.factor * x;
  }, this);
};
```

ARROW FUNCTIONS

```
function Multiplier(factor) { this.factor = factor; }

Multiplier.prototype.multiplyArray = function(arr) {
  return arr.map(function (x) {
    return this.factor * x;
  }.bind(this));
};
```

ARROW FUNCTIONS

```
function Multiplier(factor) { this.factor = factor; }

Multiplier.prototype.multiplyArray = function(arr) {
  return arr.map(
    function(that) {
      return function (x) {
        return that.factor * x;
      }
    }(this));
};
```

ARROW FUNCTIONS

```
function Multiplier(factor) { this.factor = factor; }

Multiplier.prototype.multiplyArray = function(arr) {
  return arr.map(x => this.factor * x);
};
```

ARROW FUNCTIONS

```
    () => { . . . . }

   (x) => { . . . . }

    x  => { . . . . }

(x, y) => { . . . . }
```

ARROW

```
x => { return x * x; }

x => x * x
```

PROMISES

```
function handler(request, response) {
  User.get(request.userId,
      user => {
        Notebook.get(user.notebookId,
            notebook => {
              doSomethingAsync(notebook,
                result => response.send(result)
                error => response.send(error);
            },
            error => response.send(error);
      },
      error => response.send(err);
  );
}
```

PROMISES

```
function asyncFunc() {
    return new Promise(
        (resolve, reject) => {
            // ---
            resolve(result);
            // ---
            reject(error);
        });
}

asyncFunc()
  .then(result => { ... })
  .catch(error => { ··· });
```

PROMISES

```
asyncFunc()
  .then(result1 => {
    return anotherAsyncFunction(result1);
  })
  .then(result2 => {
    // ---
  })
  .catch(error => {
    // ---
  });
```

PROMISES

```
function handler(request, response) {
  User.get(request.userId)
    .then(user => Notebook.get(user.notebookId))
    .then(notebook => doSomethingAsync(notebook))
    .then(result => response.send(result))
    .catch(error => response.send(error));
}
```

PROMISES

```
Promise.all([
  asyncFunc1();
  asyncFunc2();
])
.then(([result1, result2] => {
  // ---
})
.catch(error => {
  // ---
});;
```

PROMISES

```
function httpGet(url) {
  return new Promise((resolve, reject) => {
      const request = new XMLHttpRequest();
      request.onload = function() {
        if (this.status === 200)  // Success
          resolve(this.responseText);
        else                      // Something went wrong (404 etc.)
          reject(new Error(this.statusText));
      };
      request.open('GET', url);
      request.send();
    });
}
httpGet('http://www.jezuk.co.uk/')
  .then(body => console.log(body))
  .catch(reason => console.error(`OOPS: ${reason}`));
```

AND THE REST …

ARRAY METHODS

```
Array.of

Array.prototype.map
Array.prototype.filter
Array.prototype.reduce
Array.prototype.includes
Array.prototype.find
Array.prototype.findIndex
```

ARRAY METHODS

```
const data = Array.of(2, 3, 5, 7, 11);
                                        // [2, 3, 5, 7, 11]
const doubled = data.map(n => n*2);
                                        // [4, 6, 10, 14, 22]
const smallish = data.filter(n => n<7);
                                        // [2, 3, 5]
const total = data.reduce((acc, n) => acc+n, 0);
                                        // 28
const biggerThanFive = data.find(n => n>5);
                                        // 7
const idx = data.findIndex(n => n > 5);
                                        // 3
```

OBJECT METHODS

```
Object.keys
Object.values
Object.entries
```

OBJECT METHODS

```
const accu_member = {
  number: '03028',
  name: 'Jez Higgins',
  city: 'Birmingham',
}

for (const key of Object.keys(accu_member))
  console.log(key)

for (const value of Object.values(accu_member))
  console.log(value)

for (const [key, value] of Object.entries(accu_member))
  console.log(`${key} = ${value}`)
```

TEMPLATE LITERALS

```
let event = "accu2017";

let msg = `Hello everyone at ${event}`;
```

CLASS DEFINITIONS

```
class View extends Component {
  constructor(selector) {
    super(selector);
    // ---
  }
  render(surface) {
    // ---
  }
  get visible() {
    // ---
  }
  static allViews() {
    // ---
  }
}
```

MODULES

```
// lib/maths.js
export function mult(x y) {
  return x + y;
}

// app.js
import sum from "lib/maths";

let theAnswer = sum(5, 8);
```

AND THE REST …

- Additional library methods - maths, numbers, strings, arrays, objects
- Binary and octal numeric literals
- Regular expressions
- Subclassable Built-ins
- Unicode
- Tail calls
- Proxies and reflection
- Module loaders

USING ES6 TODAY

- You can use it today
- Transpiling through Babel
- Natively with Node
- You **should** use it today

INTO THE FUTURE

- ES2016 is already here
- ES2017 is almost here

ASYNC/AWAIT

```
async function(request, response) {
  try {
    const user = await User.get(request.userId);
    const notebook = await Notebook.get(user.notebookId);
    response.send(await doSomethingAsync(notebook));
  } catch(err) {
    response.send(err);
  }
}
```

INTO THE FUTURE

- ES2016 is already here
- ES2017 is almost here
- ES2018 is on the move
- And so is ES2019

THANKS

- JavaScript has been profoundly changed by ES6
- It makes our lives easier
- Which will make our programs better

Jez Higgins

@jezhiggins

FURTHER READING

- The ECMAScript Github repo, especially the proposals
- ECMAScript compatibility page tracks the standards
- Accessible yet detailed, Axel Rauschmayer is in a league of his own. His Exploring ES6 and Exploring ES2016 and ES2017 are great.
- Not about ES6 per se, JavaScript Allonge is a lovely book.
- The Iteration with generators and Coroutines code shown in this talk.