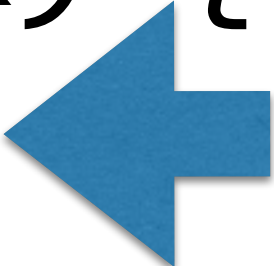


**JS**

# Arrow Functions

```
var nums = [1,2,3];
```

```
nums.map(function(x) {  
    return x * 2;  
});
```

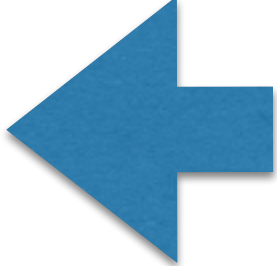


```
nums.map((x) => x * 2);
```

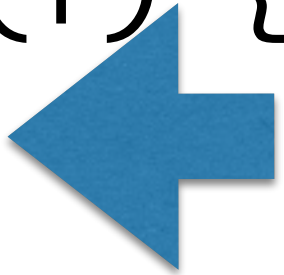
```
var nums = [1,2,3];
```

```
nums.map(function(x) {  
    return x * 2;  
});
```

```
nums.map((x) => x * 2);
```

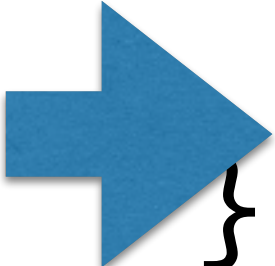


```
var jack = {  
  name: 'jack',  
  friends: ['james', 'steve'],  
  printFriends: function() {  
    this.friends.forEach(function(f) {  
      log(this.name, 'knows', f);  
    })  
  }  
};
```

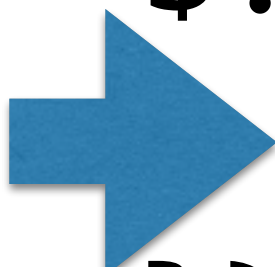


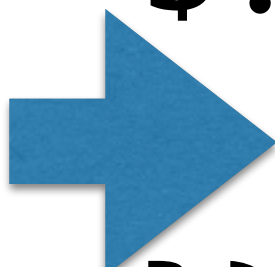
```
// undefined knows james  
// undefined knows steve
```

```
var jack = {  
  name: 'jack',  
  friends: ['james', 'steve'],  
  printFriends: function() {  
    this.friends.forEach((f) => {  
      log(this.name, 'knows', f);  
    })  
  }  
};
```



```
// jack knows james  
// jack knows steve
```

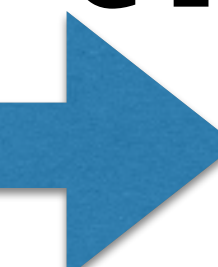
```
var someAjax = {  
  name: 'jack',  
  get: function() {  
    $.getJSON(url, function(d) {  
       log(this.name, d);  
    })  
  }  
};
```

```
var someAjax = {  
  name: 'jack',  
  get: function() {  
    $.getJSON(url, (d) => {  
      log(this.name, d);  
    })  
  }  
};
```



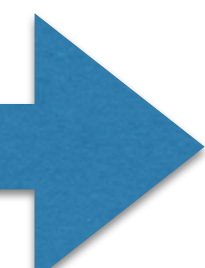
# Classes

```
class Person {  
    constructor(name, age) {  
        this.name = name,  
        this.age = age  
    }  
    about() {  
        log(this.name, this.age);  
    }  
};
```



```
class Person {  
    constructor(name, age) {  
        this.name = name,  
        this.age = age  
    }  
    about() {  
        log(this.name, this.age);  
    }  
};
```

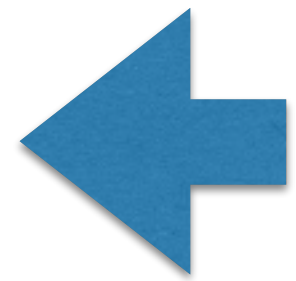
```
class Person {  
    constructor(name, age) {  
        this.name = name,  
        this.age = age  
    }  
    about() {  
        log(this.name, this.age);  
    }  
};
```



```
var jack =  
    new Person('jack', 22);
```

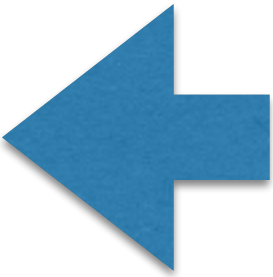
```
jack.about();  
//=> jack 22
```

```
class Son extends Person {  
    constructor(name, age) {  
        super(name, age);  
        this.son = true;  
    }  
};
```




```
var jack = new Son('jack', 22);  
jack.about(); //=> jack 22  
jack.son; //=> true
```

```
class Son extends Person {  
    constructor(name, age) {  
        super(name, age);  
        this.son = true;  
    }  
};
```



```
var jack = new Son('jack', 22);  
jack.about(); //=> jack 22  
jack.son; //=> true
```

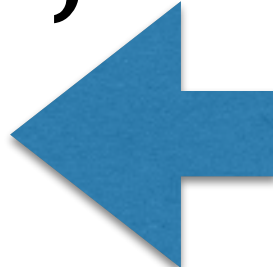
```
class Son extends Person {  
    constructor(name, age) {  
        super(name, age);  
        this.son = true;  
    }  
};
```



```
var jack = new Son('jack', 22);  
jack.about(); //=> jack 22  
jack.son; //=> true
```



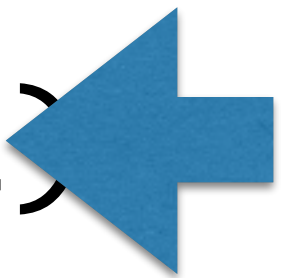
```
class Son extends Person {  
    constructor(name, age) {  
        super(name, age);  
        this.son = true;  
    }  
};
```



```
var jack = new Son('jack', 22);  
jack.about(); //=> jack 22  
jack.son; //=> true
```

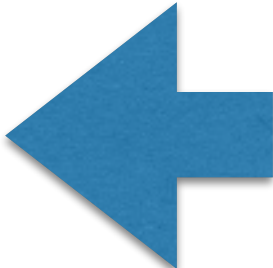
```
class Son extends Person {  
    constructor(name, age) {  
        super(name, age);  
        this.son = true;  
    }  
};
```

```
var jack = new Son('jack', 22);  
jack.about(); //=> jack 22  
jack.son; //=> true
```



# Object Literals

```
var jack = {  
  name: 'jack',  
  age: 22,  
  about() {  
    log(this.name, this.age);  
  }  
};
```



```
var jack = {  
  ['hello_' + (() => 'world')()]: 42  
};
```

```
console.log(jack.hello_world); // 42
```

# Template Strings

`In JavaScript `'\n'` is a line-feed.`

```
var f = `Multiline strings with  
back ticks in ES6.`
```

```
console.log(f);
```

Multiline strings with  
back ticks in ES6



```
var name = 'jack';
```

```
var age = 22;
```

```
`${name} is ${age} years old`
```

# Destructuring

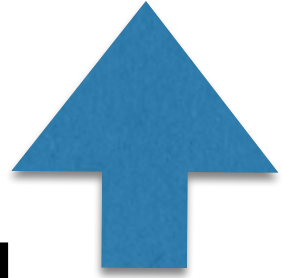
```
var [a, b] = [1, 2];
```

```
a == 1
```

```
b == 2
```

```
var [a, ,b] = [1, 2, 3];
```

a == 1



b == 3

```
var {a, b} = {a: 2, b: 3}
```

```
a == 2
```

```
b == 3
```

```
var {a, b} = {b: 3}
```

```
a == undefined
```



```
b == 3
```

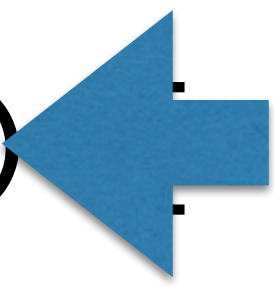
```
function getPerson() {  
    return { name: 'jack',  
            age: 22  
          }  
};
```

```
var {name, age} = getPerson();
```

# Function Arguments



```
function getInfo(print: false)
  if(print) {
    log(this.name, this.age);
  } else {
    return
      `${this.name} ${this.age}`
  }
}
```



```
getInfo();
```

```
function length(...nums) {  
    log(nums.length);  
};
```

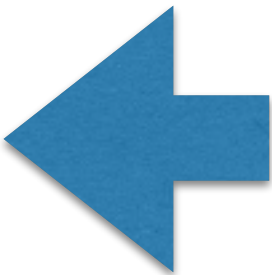


```
length(1,2,3); // 3
```

```
function total(x, y, z) {  
    log(x + y + z);  
};
```

```
total(1, 2, 3)
```

```
total.apply(null, [1, 2, 3]);
```



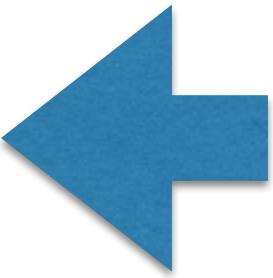
```
total(...[1, 2, 3]);
```

```
function total(x, y, z) {  
    log(x + y + z);  
};
```

```
total(1, 2, 3)
```

```
total.apply(null, [1, 2, 3]);
```

```
total(...[1, 2, 3]);
```




```
function foo({name, age}) {  
    console.log(name, age);  
}
```

```
foo({ name: 'jack', age: 22})
```

```
//=> jack, 22
```

Scope

# Global Scope (window)



Function  
Scope



```
foo = 2;
```

```
var fad = 2;
```

```
function() {  
    bar = 3;
```

```
    var baz = 4;  
}
```

```
foo: 2,  
fad: 2,  
bar: 3
```

```
baz: 4
```



```
function() {  
  if(x) {  
    var foo = 3;  
  }  
  var baz = 4;  
}
```

foo: 3,  
baz: 4



Global Scope (window)

Function Scope

Block Scope



```
foo = 2;
```

```
function() {  
  var baz = 4;  
  if(x) {  
    let y = 2;  
  }  
}
```

foo: 2

baz: 4

y: 2

```
foo = 2;
```

```
function() {  
  var baz = 4;  
  if(x) {  
    var z = 3;  
    let y = 2;  
  }  
}
```

foo: 2

baz: 4,  
z: 3

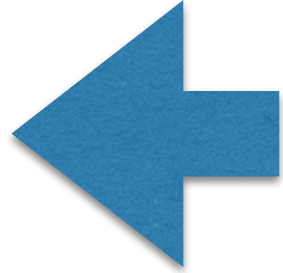
y: 2

# Modules

```
var foo = 2;  
var bar = 3;
```

app.js

```
export {foo, bar};
```



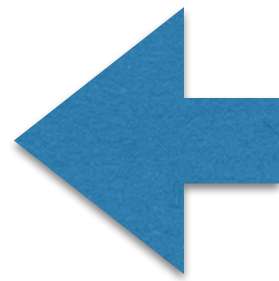
---

```
import {foo} from 'app'
```

```
console.log(foo); // 2
```

foo.js

```
export var foo = 2;
```



app.js

---

```
import {foo} from 'app'
```

```
console.log(foo); // 2
```

foo.js



```
export default function() {  
  return 2;  
};
```

app.js

---

```
import foo from 'app'  
console.log(foo()); // 2
```

foo.js



```
export var foo = 2;  
export var bar = 3;
```

app.js

---

```
module stuff from 'app';
```

```
stuff.foo // 2  
stuff.bar // 3
```

foo.js

# Generators

```
Person.findOne({id: 5}, (per) =>
{
    // person has been got
    Location.findOne(..., (loc) => {
        // location has been got
    });
});
```

```
var per = yield Person.findOne(...);  
var loc = yield Location.findOne(...);  
// async but reads as sync!
```

# ES6 Today

# ECMAScript 6 compatibility table

Also see compatibility tables for [ES5](#), [ES7](#), or [non-standard](#) features

Please note that *some of these tests* represent **existence**, not functionality or full conformance.

Sort by number of features? ☐

Show obsolete browsers? ☐

		23/106	55/101	72/101	4/106	12/106	30/106	65/106	68/106	71/106	80/106	58/106	63/106	69/106	9/106	10/106	
Feature name		Current browser	Traceur	EJS	IE 10	IE 11	FF 24	FF 31	FF 32	FF 33	FF 34	CH 35, OP 22 <sup>[1]</sup>	CH 37, OP 24 <sup>[1]</sup>	Chrome 39 <sup>[1]</sup>	SF 6	SF 7	
<a href="#">proper tail calls (tail call optimisation)</a>	Ⓢ	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	
<a href="#">arrow functions</a>	Ⓢ	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	
<a href="#">const</a>	Ⓢ	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
<a href="#">let</a>	Ⓢ	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	
<a href="#">default function parameters</a>	Ⓢ	No	Yes	Yes	No	No	Yes <sup>[3]</sup>	Yes <sup>[3]</sup>	Yes <sup>[3]</sup>	Yes <sup>[3]</sup>	Yes <sup>[3]</sup>	No	No	No	No	No	
<a href="#">rest parameters</a>	Ⓢ	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	
<a href="#">spread call (...) operator</a>	Ⓢ	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No	
<a href="#">spread array (...) operator</a>	Ⓢ	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	
<a href="#">string spreading</a>	Ⓢ	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No	
<a href="#">class</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	
<a href="#">super</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	
<a href="#">computed properties</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	
<a href="#">shorthand properties</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	Yes	Yes	No	No	No	No	No	
<a href="#">shorthand methods</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	Yes	No	No	
<a href="#">modules</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	
<a href="#">for...of loops</a>	Ⓢ	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	
<a href="#">generators (yield)</a>	Ⓢ	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	
<a href="#">octal literals</a>	Ⓢ	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes <sup>[5]</sup>	Yes <sup>[5]</sup>	Yes <sup>[5]</sup>	No	No	
<a href="#">binary literals</a>	Ⓢ	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes <sup>[5]</sup>	Yes <sup>[5]</sup>	Yes <sup>[5]</sup>	No	No	
<a href="#">template strings</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	
<a href="#">tagged template strings</a>	Ⓢ	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	
<a href="#">RegExp "y" flag</a>	Ⓢ	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	
<a href="#">RegExp "u" flag</a>	Ⓢ	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	
<a href="#">typed arrays</a>	Ⓢ	Yes															
<a href="#">typed arrays (DataView)</a>	Ⓢ	Yes															
<a href="#">Map</a>	Ⓢ	No															
<a href="#">Set</a>	Ⓢ	No															

<http://kangax.github.io/compat-table/es6/>

**Enable Experimental JavaScript** Mac, Windows, Linux, Chrome OS, Android  
Enable web pages to use experimental JavaScript features. [#enable-javascript-harmony](#)  
[Enable](#)

<chrome://flags>

# Traceur

<https://github.com/google/traceur-compiler>



node --harmony

# ES6 module transpiler

<https://github.com/esnext/es6-module-transpiler>

New projects should  
use ES6 Modules