

# JavaScript (ES6) Overview

Mark DuBois – [Mark@WebProfessionals.org](mailto:Mark@WebProfessionals.org)

[@WebProMinute](#)



Most browsers = 98%; mobile = 99%

# Can I Use Today?

Feature name	Current browser	Desktop browsers							Mobile	
		Edge 16	Edge 17	FF 52 ESR	FF 58	FF 60 ESR	CH 66, OP 53	CH 67, OP 54	SF 11	SF 11.1
<b>Syntax</b>										
○ default function parameters	3/7	3/7	3/7	6/7	3/7	3/7	3/7	3/7	3/7	3/7
○ basic functionality	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ explicit undefined defers to the default	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ defaults can refer to previous params	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ arguments object interaction	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ temporal dead zone	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ separate scope	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ new Function() support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ rest parameters	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
○ basic functionality	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ function length property	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ arguments object interaction	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ can't be used in setters	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ new Function() support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ spread (...) operator	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15
○ with arrays, in function calls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with arrays, in array literals	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with sparse arrays, in function calls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with sparse arrays, in array literals	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with strings, in function calls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with strings, in array literals	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with astral plane strings, in function calls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with astral plane strings, in array literals	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with generator instances, in calls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
○ with generator instances, in arrays	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

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



## Template Strings

Remember when...

```
document.write("The value of myVar is  
" + myVar + " and...");
```



# Template Strings

```
10 ▾ const competition = {
11     compName: 'Web Professionals',
12     compType: 'Web Design',
13     supervisor: 'Mark DuBois',
14     city: "Louisville, KY",
15     dates: "June 26 - 28, 2018"
16 };
17 const myMarkup = `
18 <div>
19   <h1>${competition.compName}</h1>
20   <h2>${competition.supervisor}</h2>
21   <p>I participated in the ${competition.compName}
22     ${competition.compType} competition. It was held
23     in ${competition.city} from ${competition.dates}.
24     It is so much easier to write JS these days.</p>
25 </div>
26 `;
```

01\_ES6TemplateStrings.pdf



# Template Strings Exercise

```
18 ▾ function includeAbbr(strings, ...values) {  
19  
20 ▾     const abbreviate = values.map(???? => {  
21         if(????[????]) {  
22             return `23 ▾         } else {  
24             return ?????;  
25         } } );  
26  
27 ▾     return strings.reduce((????, string, i) =>  
28         {  
29             return ???? + string + (????[i] || '');  
30         }, '');  
31     }
```

01a\_ES6TemplateStringsExercises.pdf



# Security

Always trust your visitors...

Never trust their input...

o2\_ES6Security.pdf



## Swapping variable contents

```
20 [visible,grid,hidden] = [grid,hidden,visible];
```

o3\_ES6SwappingVariables.pdf



## New String Methods

- `endsWith( )`
- `includes( )`
- `repeat( )`
- `startsWith( )`





# New String Methods

## New String Methods

String is **Web design and development**

### **startsWith( ) method**

`startsWith('Web')? true`

`startsWith('web')? false`

`startsWith('design')? false`

`startsWith('design', 4)? true`

`endsWith('ment')? true`

`endsWith('and', 14)? true`

`includes('develop')? true`

`'123456'.repeat(2)? 123456123456`



# New String Methods

## New String Methods

### Bad Joke Coming

Remember we can multiply a string by a number to get NaN. If we apply `.repeat`, we can get...

### See Console Log

05NewStringMethods.html



# For Of Loop

## For Of Loop

Array containing ['basket', 'soccer', 'base', 'soft', 'Pooh']

### Older approach

```
for (let i = 0; i < balls.length; i++) {  
  ...  
}
```

Resulting in:

basketball soccerball baseball softball Poohball

You can also use balls.forEach(ball)... But you can't use a break or continue once the loop begins.

Or, you could use for(ball in balls)... but many people don't use for in loops.

We now have a for of loop

```
for (const ball of balls) { ...
```

Which yields

basketball soccerball baseball softball Poohball

And we can "filter" the array contents

```
for (const ball of balls) {  
  if (ball === 'Pooh') {  
    continue;  
  }  
}
```

Which yields

basketball soccerball baseball softball

o5\_ES6ForOfLoop.pdf



# Spread operator

```
14 const fruit = [...'Durian'];  
15 console.table(fruit); // because fruit belongs  
   on a table *grin*
```

o6\_ES6Spread.pdf



# Rest operator

```
14 ▾ function calcBill(salesTax, ...enteredAmounts) {  
15     // sales tax followed by an unknown number of  
    // values the 'rest' of them  
16     //console.log(salesTax, enteredAmounts); //  
    // for debugging  
17  
18     return enteredAmounts.map(amount => (amount *  
    salesTax) + amount );  
19 }
```

o7\_ES6Rest.pdf



# Generators

```
16 ▼ function* someNumbers(){  
17     yield 27310;  
18     yield 50780;  
19     yield 53323;  
20     yield 54051;  
21     yield 62310;  
22 }
```

o8\_ES6Generators.pdf



# JavaScript (ES6) Overview

Mark DuBois – [Mark@WebProfessionals.org](mailto:Mark@WebProfessionals.org)

[@WebProMinute](#)

