# ECMASCRIPT HARMONY

PlovdivConf 2015

## ЙОРДАН ИВАНОВ

https://github.com/ivanovyordan
http://ivanovyordan.com

## КОИ СТЕ ВИЕ?

# ВЪВЕДЕНИЕ

# СЪВМЕСТИМОСТ

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

| Платформа | Съвместимост |
|---|---|
| Babel.js | 76% |
| Microsoft Edge | 72% |
| Mozilla Firefox 37 | 62% |
| Traceur | 60% |
| Google Chrome 42 | 43% |
| Microsoft IE 11 | 15% |

## КАКВО ДА ПРАВИМ ДНЕС?

- https://babeljs.io
- https://github.com/google/traceur-compiler

```
var gulp = require('gulp');
var babel = require('gulp-babel');

gulp.task('default', function () {
  return gulp.src('src/app.js')
    .pipe(babel())
    .pipe(gulp.dest('dist'));
});
```

## КАКВО НОВО?

- let и const
- Object.observe()
- Аргументи
- Arrows
- Promises
- Класове
- Шаблони
- Модули

## LET И CONST

```
function let_const() {
  let foo;

  if(true) {
    // OK
    const foo = 'bar';

    // Error
    foo = 'baz';
  }

  // Error
  let foo = 'baz';
}
```

```
function letTest() {
  let x = 31;

  {
    let x = 71;
    console.log(x);  // 71
  }

  console.log(x);  // 31
}
```

## OBJECT.OBSERVE()

```
let obj = {
  foo: 0,
  bar: 1
};

Object.observe(obj, function(changes) {
  console.log(changes);
});

obj.baz = 2;
// [{name: 'baz', object: [object], type: 'add'}]
```

## СТОЙНОСТИ ПО ПОДРАЗБИРАНЕ

```
function setBackgroundColor(element, color = 'rosybrown') {
  element.style.backgroundColor = color;
}

setBackgroundColor(someDiv);               // 'rosybrown'
setBackgroundColor(someDiv, undefined); // 'rosybrown'
setBackgroundColor(someDiv, 'blue');    // 'blue'
```

## ОСТАВАЩИ ПАРАМЕТРИ

```
add = function(category, ...items) {
  console.log(category + ': ' + items.join(', '));
};

add('fruit', 'apple'); // fruit: apple
add('dairy', 'milk', 'cheese'); // dairy: milk, cheese
```

## ARROWS

```
var bob = {
  name: 'Bob',
  friends: [],
  printFriends() {
    this.friends.forEach(friend => {
      console.log(this.name + ' knows ' + friend);
    });
  }
}
```

## PROMISES

```
function timeout(duration = 0) {
  return new Promise((resolve, reject) => {
    setTimeout(resolve, duration);
  })
}

let promise = timeout(1000).then(() => {
  return timeout(2000);
}).then(() => {
  throw new Error('hmm');
}).catch(err => {
  return Promise.all([timeout(100), timeout(200)]);
});
```

## КЛАСОВЕ

```
class Marine extends Unit {
  constructor(name) {
    this.name = name;
  }

  sayHi() {
    return super.sayHello();
  }

  static spawn() {
        return new Marine('Jim Raynor');
  }
}
```

## ШАБЛОНИ

```
const template = `
  <table>
  ${people.map(person => `
    <tr>
      <td>$${person.firstName}</td>
      <td>$${person.lastName}</td>
    </tr>
  `)}
  </table>
`;
```

## МОДУЛИ

```
// lib/math.js
export function sum(x, y) {
  return x + y;
}

export var pi = 3.141593;

// app.js
import * as math from 'lib/math';
alert('2π = ' + math.sum(math.pi, math.pi));
```

## ВЪПРОСИ?

БЛАГОДАРЯ ВИ?