



JS

JavaScript

лекція 11

Огляд нововведень в ECMAScript 6

Трішки історії

- В грудні 2009 затверджено версію №5, де додано режим "strict mode"
- В червні 2011 затверджено версію 5.1
- І в червні 2015 шоста редакція яка відома як ES6 або ECMAScript 2015
- Додано новий функціонал для написання складних додатків, включно класи та модулі
- Наразі ведеться робота над версією №7
- Підтримка браузерами <https://kangax.github.io/compat-table/es6/>

Підтримка класів

- class
- constructor
- extends замість прототипів
- super
- методи та властивості
- Особливості
 - Не можна викликати без new
 - Методи мають доступ до super
 - всі методи працюють в строгому режимі
 - методи не можна перелічити через for..in
 - класи не впливають

Строкові шаблони

- Використовуються замість конкатенація
- Підтримують кілька рядків

```
let str = `Здравствуйτε ${first}. Мы в ${new Date().getFullYear()} году`;
```

Оператор const та let

- let доступний в середині блоку, особливо важливо для циклів for і while
 - доступна лише після оголошення
 - не можна переоголосити
 - доступна лише в середині блоку
- const - використовується для оголошення незмінних котрі не можна змінювати
 - все як в let

Скорочений формат створення об'єктів

- `"var a = {obj1, obj2}"` замість `"var a = {'obj1': obj1, 'obj2': obj2}");`

Модулі

- По суті модуль це інструкція котра викликається неявно, за допомогою створення файлу та виконання його за допомогою інтерпретатора ES
- Один файл - один модуль
- Модуль має окрему область видимості
- `export`
 - властивість `[[Exports]]` у модулів
 - `export var variable;`
 - `export const CONSTANT = 0;`
 - `export let scopedVariable = 20;`
 - `export function func(){};`
 - `export class Foo {};`
- `Import`
 - `import <любое имя> from '<путь к модулю>';`
 - `import * as sub from './sub.js';`

Arrow-функції

- $(x, y) \Rightarrow \{ \}$ замість `function (x,y) { }`

Генератори

- Відрізняються тим що можуть перервати виконання ф-ї, повернути значення і далі продовжити виконання

```
'use strict';  
function* generateSequence() {  
  yield 1;  
  yield 2;  
  return 3;  
}  
let generator = generateSequence();  
let one = generator.next();  
alert(JSON.stringify(one)); // {value: 1, done: false}
```

Об'єкти WeakSet, WeakMap

```
var promise = new Promise(function(resolve, reject) {  
  // Эта функция будет вызвана автоматически  
  
  // В ней можно делать любые асинхронные операции,  
  // А когда они завершатся — нужно вызвать одно из:  
  // resolve(результат) при успешном выполнении  
  // reject(ошибка) при ошибке  
})
```

Механізм Promise

```
var promise = new Promise(function(resolve, reject) {  
  // Эта функция будет вызвана автоматически  
  
  // В ней можно делать любые асинхронные операции,  
  // А когда они завершатся — нужно вызвать одно из:  
  // resolve(результат) при успешном выполнении  
  // reject(ошибка) при ошибке  
})
```


Значення по замовчуванню для аргументів функції

```
function multiply(a, b = 1) {  
    return a*b;  
}  
multiply(5); // 5
```

Структури даних Map та Set

- Структури даних Map і Set, спрощують роботу зі специфічними типами колекцій. Map дозволяє визначати колекції наборів в форматі ключ / значення, при тому, що в якості ключа і значення можуть виступати будь-які вирази JavaScript. За аналогією Set дозволяє задати безліч будь-яких виразів JavaScript;

Нові методи для строк, масивів, математичних операцій

- `Array.from()` 22.1.2.1
- `Array.prototype.keys` 22.1.3.13
- `Array.prototype.entries` 22.1.3.4
- `Array.prototype.copyWithin()` 22.1.3.3
- `Number.isSafeInteger()` 20.1.2.5
- `Math.fround()` 20.2.2.17
- `String.prototype.startsWith()` 21.1.3.18
- `String.prototype.endsWith()` 21.1.3.6
- `Math.imul` 20.2.2.19

Babel

- <https://babeljs.io/>
- Початок роботи
 - Створити каталог для файлів
 - Виконати команди
 - `npm init -y`
 - `npm install babel-cli babel-core babel-preset-es2015 --save-dev`
 - Додати скрипти в `package.json`
 - `"build": "babel src -d dist --presets es2015",`
 - `"watch": "babel src -d dist --presets es2015 -w"`
 - Створити файл з кодом ES6
 - Виконати
 - `npm run build`
 - `npm run watch`

Завдання