

JavaScript

Table des matières

| | |
|---|----|
| 1. Bases | 3 |
| a. typeof / comparaison | 3 |
| b. Extend | 3 |
| c. Sélection multiple | 3 |
| d. Event sur « parent » | 3 |
| e. Array | 4 |
| f. Regex | 6 |
| g. Custom Events | 7 |
| h. Callbacks / Promises | 8 |
| Callbacks | 8 |
| Promises | 8 |
| 2. Patterns | 9 |
| a. IIFE | 9 |
| b. Reveal pattern | 9 |
| « Singleton » | 9 |
| « Constructor » | 11 |
| c. Objet | 11 |
| d. defineProperty | 12 |
| e. Constructor | 12 |
| Avoir une fonction (avec paramètres) retournant un objet sans avoir à faire « new » | 13 |
| Fonction « Static » | 13 |
| f. « Façon TypeScript » | 13 |
| g. heritage | 14 |
| h. Ajout d'un objet à « window » | 14 |
| i. Fonction | 14 |
| Créer un contexte | 15 |
| j. Chaining | 15 |
| k. Factory | 15 |
| l. Singleton | 16 |
| m. Mediator | 16 |
| n. Command | 17 |
| 3. Formats, loaders, bundlers | 17 |
| a. UMD | 18 |

| | |
|--|----|
| Browser | 18 |
| Node | 18 |
| b. AMD..... | 18 |
| Browser | 19 |
| Node | 19 |
| c. CommonJS | 20 |
| Browser | 20 |
| Node | 21 |
| d. Es6..... | 22 |
| Browser | 23 |
| Node | 25 |
| e. TypeScript | 26 |
| Browser | 26 |
| 4. Tests..... | 28 |
| a. Jasmine "browser"..... | 28 |
| b. Mocha "browser" | 28 |
| Génération d'un bundle avec Webpack | 28 |
| c. Karma | 29 |
| d. Scripts « package.json » | 30 |
| e. Tests e2e..... | 30 |
| 5. Intégration continue | 31 |
| a. Editor config | 31 |
| b. Serveur de développement..... | 31 |
| c. Automation..... | 32 |
| d. Linter | 32 |
| e. Bundler | 33 |
| f. CI server..... | 34 |
| 6. Grunt..... | 35 |
| a. Installation en global de grunt..... | 35 |
| b. Concat | 35 |
| c. « grunt-browserify »..... | 36 |
| d. Serveur local..... | 36 |
| e. Minimiser les fichiers js avec « Uglify » | 38 |
| f. Pour copier des fichiers | 38 |
| g. Plus | 39 |
| 7. JS code Sniffer | 39 |

1. Bases

a. typeof / comparaison

```
console.log(10 == '10'); // true
console.log(10 === '10'); // false

console.log(null == undefined); // true
console.log(null === undefined); // false

console.log(false == 0); // true
console.log(false === 0); // false
console.log(true == 1); // true
console.log(true === 1); // false
```

b. Extend

```
var extend = function (source, target) {
  if (typeof source == 'object') {
    for (var property in source) {
      if (!target.hasOwnProperty(property)) {
        target[property] = source[property];
      }
    }
  }
  return target;
}
```

c. Sélection multiple

On sépare d'une virgule chaque sélection

```
var elements = document.querySelectorAll('.item-1, .item-2');
```

d. Event sur « parent »

Exemple le code de la fonction n'est exécutée (affichage d'une boîte de dialogue) que si on clique sur un bouton.

```
<div class="container">
  <p class="bg-info">Lorem ipsum ...</p>
  <button class="btn btn-default">Ok</button>
  <button class="btn btn-danger">Cancel</button>
</div>

<script>
  var container = document.querySelector('.container');
  container.addEventListener("click", doSomething, false);

  function doSomething(e) {
    var target = e.target;
    if (target.nodeName == 'BUTTON') {

      // do...
      alert();
    }
  }
</script>
```

e. Array

Création

```
var arr = []; // tableau vide
// ou
var arr = ['a', 'b', 'c'];
```

Ajout / insertion

- push

```
arr.push('d');
arr.push('e', 'f'); // plusieurs valeurs
```

- concat

```
var arr = ['a', 'b', 'c'];
var arr2 = ['x', 'y', 'z'];
arr = arr.concat(arr2);
console.log(arr); // ["a", "b", "c", "x", "y", "z"]
```

- slice

Extrait les éléments à l'index, on peut indiquer en second paramètre l'index de fin

```
var a1 = ['a', 'b', 'c', 'd', 'e']; // retirer c et d
var a2 = ['1', '2', '3']; // insérer à la place a2

var a1 = a1.slice(0, 2); // ['a', 'b']
// concaténation de tableau
a1 = a1.concat(['1', '2', '3']); // ['a', 'b', '1', '2', '3']
// chaining
a1 = a1.slice(0, 2).concat(a2);
```

- Unshift

Documentation

Ajout en début de tableau les arguments

- Splice

Documentation

Ajout des éléments à l'index

Accès

Par index

```
var arr = ['a', 'b', 'c'];

console.log(arr[1]); // b
arr[0] = 'x';
```

Obtenir l'index d'un élément

```
var arr = ['a', 'b', 'c', 'a'];

console.log(arr.indexOf('c')); // 2
console.log(arr.indexOf('z')); // -1
```

Suppression

- pop

Supprime et renvoie le dernier élément

```
console.log(arr.pop());
```

- Shift

Inverse de pop. Supprime et renvoie le premier élément.

```
console.log(arr.shift());
```

- splice

```
var arr = ['a', 'b', 'c'];

arr.splice(0, 2); // index , nombre d'éléments
console.log(arr); // ["c"]
```

On peut « transférer » les éléments

```
var arr = ['a', 'b', 'c'];

var arr2 = arr.splice(0, 2); // index , nombre d'éléments
console.log(arr2); // ["a", "b"]
console.log(arr); // ["c"]
```

Tri

```
var arr = [2, 8, 5, 1];
console.log(arr.sort()); // [1, 2, 5, 8]

var arr = ['c', 'a', 'b'];
console.log(arr.sort()); // ["a", "b", "c"]
```

Avec fonction : évalue si la valeur retournée est inférieure à la valeur précédente

```
var arr = [2, 8, 5, 1];
arr.sort(function(val1, val2) {
  return val1 - val2;
});
console.log(arr); // [1, 2, 5, 8]
```

To string

```
var arr = ['a', 'b', 'c'];
console.log(arr.toString()); // "a,b,c"
```

f. Regex

test

```
var text = 'Le chat est blanc. Quel beau chat!';
var pattern = /chat/i;

console.log(pattern.test(text)); // true
```

Drapeau i pour ignorer la casse

Ou avec RegExp

```
var text = 'Le chat est blanc. Quel beau chat!';
var regex = new RegExp('chat', 'i');

console.log(regex.test(text)); // true
```

Exec

```
var text = 'Le chat est blanc. Quel beau chat!';
var regex = new RegExp('chat', 'g');
var result = regex.exec(text);
console.log(result.index); // 3

result = regex.exec(text);
console.log(result.index); // 29
```

Drapeau global, sinon ne renvoie que la première instance

Avec une boucle

```
var text = 'Le chat est blanc. Quel beau chat!';
var regex = new RegExp('chat', 'g');
var result = regex.exec(text);

while(result) {
    console.log(result.index);
    result = regex.exec(text);
}
```

Match

```
var text = 'Le chat est blanc. Quel beau chat!';
var regex = new RegExp('chat', 'g');

var result = text.match(regex);
console.log(result); // ["chat", "chat"]
```

g. Custom Events

```
function detectIE() {
    var ua = window.navigator.userAgent;
    // IE 10
    // ua = 'Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)';

    // IE 11
    // ua = 'Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko';

    // Edge 12 (Spartan)
    // ua = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/39.0.2171.71 Safari/537.36 Edge/12.0';

    // Edge 13
    // ua = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/46.0.2486.0 Safari/537.36 Edge/13.10586';

    var msie = ua.indexOf('MSIE ');
    if (msie > 0) {
        // IE 10 or older
        return parseInt(ua.substring(msie + 5, ua.indexOf('.', msie)), 10);
    }

    var trident = ua.indexOf('Trident/');
    if (trident > 0) {
        // IE 11
        var rv = ua.indexOf('rv:');
        return parseInt(ua.substring(rv + 3, ua.indexOf('.', rv)), 10);
    }

    var edge = ua.indexOf('Edge/');
    if (edge > 0) {
        // Edge (IE 12+)
        return parseInt(ua.substring(edge + 5, ua.indexOf('.', edge)), 10);
    }
    // other browser
    return false;
}

var event,
    appName = navigator.appName;

if(detectIE()){
    console.log(detectIE());
    event = document.createEvent('Event');
    event.initEvent('myevent', true, true);
}
else{
    event = new Event('myevent');
}

//Ecoutez l'événement.
document.addEventListener('myevent', function (e) {
    console.log(e); // Event {isTrusted: false, type: "myevent", target: document,
    currentTarget: document, eventPhase: 2...}
}, false);

//fire
document.dispatchEvent(event);
```

h. Callbacks / Promises

Callbacks

Simple

```
function p1 (resolve) {
  resolve("p1");
}
function c1 (r1) {
  console.log(r1);
  return "from c1";
}
// appel
p1(c1);
```

Multiple

```
function p2(callback) {
  callback("p2");
}

function p3(callback) {
  callback("p3");
}

function runAll() {
  p1(function(r1) {
    console.log(r1);

    p2(function(r2) {
      console.log(r2);

      p3(function(r3) {
        console.log(r3);
      });
    });
  });
}
// appel
runAll();
```

Promises

Simple

```
var promise1 = new Promise(p1);
promise1.then(c1);
```

Multiple

```
function r1All(r1) {
  console.log(r1); // ["p1", "p2", "p3"]
  return "from r1 All"; // result return
}
function r2All(r2) {
  console.log(r2); // "from r1"
}
var promise1 = new Promise(p1);
var promise2 = new Promise(p2);
var promise3 = new Promise(p3);
Promise.all([promise1, promise2, promise3])
  .then(r1All).then(r2All); // ...
```


2. Patterns

a. IIFE

Fonction anonyme immédiatement invoquée, scope isolé. Exemple avec passage de paramètre (jQuery)

```
(function ($) {  
  
})(jQuery);
```

b. Reveal pattern

« Singleton »

Une seule instance

```
var demo = function () {  
    var message = "hello";  
  
    function init (options){  
    }  
    function doSomething(){  
    }  
    return {  
        init : init,  
        doSomething : doSomething  
    }  
}();
```

Utilisation

```
demo.init({ ... });  
demo.doSomething();
```

Subtilité

```
var revealSingleton = (function () {  
    console.log('reveal'); // exécuté immédiatement  
  
    function doSomething () {  
        console.log('reveal -> do something');  
    }  
  
    return {  
        doSomething : doSomething  
    }  
})();
```

Exécuté immédiatement

Exécuté seulement si on appelle la fonction avec revealSingleton.doSomething()

« Injection »

```

<div class="container">
  <button class="btn btn-default">Ok</button>
  <button class="btn btn-danger">Cancel</button>
</div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
<script>

  var Item = (function(){
    var doSomething = function () {

    }
    return {
      doSomething : doSomething
    }
  })();

  (function ($, Item) {

    $('.btn').on('click', function(){
      Item.doSomething();
    })

  })(jQuery, Item);

</script>

```



« Augmentation »


```

var Item = (function (my) {
  my.doSomething = function () {
    alert();
  }

  return my;
})(Item);

Item.doSomething();

```



« Loose augmentation »


```

var Item = (function (my) {
  my.doSomething = function () {
    alert();
  }

  return my;
})(Item || {});

Item.doSomething();

```



« Constructor »

On peut créer des instances avec « new »

```
var Demo = function(options) {
    var message = "hello";

    function doSomething(){
    }
    return{
        doSomething:doSomething
    }
};
```

Utilisation

```
var demo = new Demo({ ... });
demo.doSomething();
```

c. Objet

```
var item = {
    name : 'item 1',
    doSomething : function () {

    }
};
//pour créer une nouvelle instance
var new_item = Object.create(item);
new_item.name = 'item 2';

console.log(item.name);
console.log(new_item.name);
```

Accès aux propriétés

```
item.name = 'a';
// ou
item['name'] = 'b';
```

On utilisera « `__proto__` » pour ajouter des membres supplémentaires et non pas « prototype »

```
item.__proto__.myFunc = function () {
};
```

Ou direct

```
item.myFunc = function () {
};
```

d. defineProperty

Exemple une propriété « readonly »

```
var item = {
  name : ''
};

Object.defineProperty(item, 'my_readonly_prop', {
  value : 'abcde',
  writable : false,
  enumerable : true,
  configurable : true
});

console.log(item.my_readonly_prop);
```

La valeur pourrait être une fonction

On obtient les « clés » (seulement si la propriété a enumerable à true)

```
console.log(Object.keys(item)); // ["name", "my_readonly_prop"]
```

Si enumerable de « my_readonly_prop » à false , ne renvoie que « name »

Obtenir les informations sur la propriété

```
console.log(Object.getOwnPropertyDescriptor(item, 'my_readonly_prop')); //
Object {value: "abcde", writable: false, enumerable: true, configurable:
true}
```

Héritage avec

```
var mySubItem = Object.create(item);
```

e. Constructor

```
function Item(name) {
  this.name = name;
  // method
  this.doSomething = function () {
  }
}
// prototype
Item.prototype.doWork = function () {
  return this.name;
};
// instances
var item1 = new Item('item1');
var item2 = new Item('item2');

item1.doWork();
item1.doSomething();
```

Note : un prototype est partagé par l'ensemble des objets. Ainsi si on crée un objet avec un prototype et que l'on modifie une propriété pour un objet. Tous les objets auront cette modification. Exemple

```
Item.prototype.prop = { id: '' };
Item1.prop.id = '1';
console.log(item2.prop); // item2 aura sa propriété modifiée
```

Avoir une fonction (avec paramètres) retournant un objet sans avoir à faire « new »

```
var item = function (parameter){
  var result = {
    doSomething : function(){
      console.log('do something with ' + parameter);
    }
  }
  return result;
}

var result = item('Hello');
result.doSomething();
```

Fonction « Static »

```
function Item (name){
  this.name = name;
}
Item.myStaticFunction = function (){

};
// utilisation
Item.myStaticFunction();
```

f. « Façon TypeScript »

```
var Item = (function() {

  var privateMember = 'private!';

  // constructor
  function Item(name) {
    this.name = name;
    // method
    this.doSomething = function() {
      console.log('do something ' + this.name);
    }
  }
  // ou prototype
  Item.prototype.doWork = function() {
    console.log('do ...' + this.name);
  };

  return Item; // pointe le constructeur
})();

// instances
var item1 = new Item('item1');
var item2 = new Item('item2');
// etc.
item1.doWork();
item1.doSomething();
```

g. heritage

Exemples sur developpez.com (en 3 parties).

Exemple avec « super »

```
var HistoryBase = (function() {
  var subscribers = [];

  function HistoryBase() { }
  HistoryBase.prototype.subscribe = function (subscriber) {
    subscribers.push(subscriber);
  };
  return HistoryBase;
})();
```

```
var HashHistory = (function(_super) {

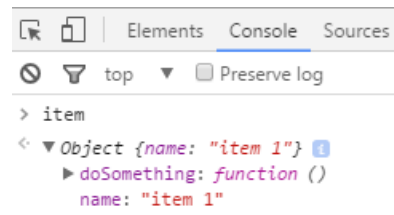
  function HashHistory() {
    _super.call(this, arguments);
  }
  HashHistory.prototype = Object.create(HistoryBase.prototype);
  HashHistory.prototype.constructor = HashHistory;

  return HashHistory;

})(HistoryBase);
```

h. Ajout d'un objet à « window »

```
var item = {
  name : 'item 1',
  doSomething : function () {
  }
};
window.item = item;
```



i. Fonction

Tous les arguments ne sont pas requis. De même on peut passer plus d'arguments que prévu. Accès aux arguments de fonction

```
function func(a,b,c) {
  for(var i = 0; i < arguments.length; i++){
    console.log(arguments[i]); // affiche 10 puis 20
  }
  console.log(c); // undefined
}
func(10,20);
```

Apply

Applique les éléments d'un tableau aux paramètres d'une fonction

```
function fn(p1,p2) {
}
fn.apply(this, ["a", "b"]);
```

Call

```
function fn(p1,p2) {
}
fn.call(this, "a", "b");
```

« Array slice call »

```
function doSomething(){
  var args = Array.prototype.slice.call(arguments, 1);
  console.log(args); // [10]
};
doSomething('abc', 10);
```

Créer un contexte

```
function doSomething(a,b){
}
var item = {
  name : 'abc'
};
doSomething.apply(item, [10, 20]);
doSomething.call(item, 10, 20);
```

Le contexte de la fonction « doSomething » sera l'item. Ce qui permet d'avoir accès aux membres d'item avec « this » depuis la fonction.

j. Chaining

On peut enchaîner les fonctions en retournant l'élément.

```
var item = {
  methodOne : function(){
    return this;
  },
  methodTwo : function (){
    return this;
  }
};
item.methodOne().methodTwo();
```

k. Factory

Retourne un type d'élément selon un critère (argument passé).

```
function Chat(){
  this.cri = function(){
    return 'miaou';
  }
}
function Poule(){
  this.cri = function(){
    return 'cot cot';
  }
}
var myFactory = function (){
  var getAnimal = function (name) {
    if(name == 'chat'){ return new Chat(); }
    else if(name == 'poule'){ return new Poule(); }
  }
  return{
    getAnimal : getAnimal
  }
}
```

l. Singleton

```

var Item = (function(){
  var item;
  function createItem(){
    //this. prop ...
    // this methods ...
  }
  return{
    getInstance : function (){
      if(!item){
        item = new createItem();
      }
      return item;
    }
  }
})();

var instance = Item.getInstance();

```

m. Mediator

```

var mediator = (function(){
  var container = {};

  var subscribe = function(channel, context, func){
    if (!mediator.container[channel]) {
      mediator.container[channel] = []
    }
    mediator.container[channel].push({
      context: context,
      func: func
    });
  };

  var publish = function(channel){
    if (!container[channel]) {
      return false
    }

    var args = Array.prototype.slice.call(arguments, 1);
    for(var i = 0; i < mediator.container[channel].length; i++)
    {
      var sub = mediator.container[channel][i];
      sub.func.apply(sub.context, args)
    }
  }
  return{
    subscribe:subscribe,
    publish:publish
  };
})();

```


n. Command

```

var item = {
  methodOne: function (a) {
    return 'method One :' + a;
  },
  methodTwo: function (a) {
    return 'method Two :' + a;
  }
};

item.execute = function (name) {
  var args = [].slice.call(arguments, 1); // ["A"] puis ["B"]...

  if(item[name]){
    return item[name].apply(item, args);
  }
  return false;
  // return item[name] && item[name].apply(item, args);
};
var result_a = item.execute("methodOne", "A");
var result_b = item.execute("methodTwo", "B");
var result_c = item.execute("methodNotFound", "C");

console.log(result_a); // method One :A
console.log(result_b); // method Two :B
console.log(result_c); // false

```

3. Formats, loaders, bundlers

Formats

- UMD (Universal Module Definition)
- AMD (Asynchronous Module Definition)
- **CommonJS**
- **Es6**

Loaders

- **RequireJS** (AMD)
- **SystemJS** (AMD, CommonJS, UMD, System.register)

Bundlers

- Browserify
- **Webpack**
- **Rollup**

a. UMD

Documentation

```
(function (global, factory) {
  if (typeof define === 'function' && define.amd) {
    define([], factory); // AMD
  } else if (typeof exports === 'object' && typeof module !== 'undefined') {
    module.exports = factory(); // CommonJS
  } else {
    global.lib = factory();
  }
})(this, function () {

  return {
    version : "1.0"
  };
});
```

Remplacer par le nom de la librairie

Instance retournée

Browser

Il suffit de référencer le fichier dans la page d'index

```
<body>
  <script src="src/lib.js"></script>
  <script>
    console.log(lib.version);
  </script>
</body>
```

Note : il serait également possible d'utiliser le format AMD

Node

On peut faire directement un « require »

```
var lib = require('./src/lib');
console.log(lib.version);
```

b. AMD

Exemple

« item.js »

```
define(function () {
  function doSomething(name){
    console.log('Hello ' + name);
  }
  return {
    doSomething: doSomething
  }
});
```

« app.js »

```
define(['./item'], function (item) {
  item.doSomething('marie');
});
```

Browser

Besoin de « [require.js](#) »

Installation avec un **gestionnaire de packages** :

Bower

```
npm i requirejs -S
```

NPM

```
npm i requirejs -S
```

Autre possibilité : téléchargement depuis le [site](#)

Référencer « require.js » dans la page d'index en indiquant le « main » avec l'attribut « data-main »

```
<script data-main="src/app" src="lib/require.js"></script>
```

Node

Avec un bundler : webpack

Installation

```
npm i webpack -D
```

« **webpack.config.js** »

```
module.exports = {
  entry: {
    app: './src/lib.js',
    other: ['./src/other.js'],
    item: ['./src/item.js']
  },
  output: {
    path: './dist',
    filename: '[name].js',
    libraryTarget: 'commonjs2'
  }
};
```

Génération des fichiers avec webpack

```
"scripts": {
  "build": "webpack",
  "start": "node server.js"
},
```

```
npm run build
```

```

▼ dist
  app.js
  item.js
  other.js
▶ node_modules
▼ src
  app.js
  item.js
  other.js

```

On génère plusieurs
fichiers avec webpack

c. CommonJS

Format utilisé par Node.js

Export de fonction

```
exports.fn = function (name) {
  console.log("hi " + name);
};
// ou
function fn (name) {
  console.log("hi " + name);
}
exports.fn = fn;
```

Import et utilisation de la fonction

```
var fn = require('./item').fn;
fn("marie");
```

Export d'objet

```
module.exports = {
  fn: function (name) {
    console.log("hello " + name);
  }
};
```

Le code d'import et d'utilisation est identique à celui pour la fonction.

Browser

- Avec un loader : SystemJS

Installation

```
npm i systemjs -S
```

Dans la page d'index

```
<script src="node_modules/systemjs/dist/system.js"></script>
<script>
  System.config({
    meta: {
      format: 'cjs'
    }
  });
  System.import('src/app.js');
</script>
```

- Avec un bundler : webpack

Installation

```
npm i webpack -D
```

« **webpack.config.js** »

```
module.exports = {  
  entry: './src/lib.js',  
  output: {  
    path: './dist',  
    filename: 'bundle.js'  
  }  
};
```

Génération du bundle

Commande ou script NPM

```
"scripts": {  
  "build": "webpack",  
  "start": "node server.js"  
},
```

```
npm run build
```

Référencer le bundle dans la page d'index

```
<script src="dist/bundle.js"></script>
```

Node

Utilisable directement. C'est le format par défaut de Node.js

d. Es6

Export de fonction

```
export function fn(name) {
  console.log("hello " + name);
}
// ou
function fn(name) {
  console.log("hello " + name);
}
export {fn};
```

On peut indiquer plusieurs éléments exportés d'un même fichier (séparés par une virgule)

Import et utilisation de fonction

```
import {fn} from './item';
fn("marie");
```

On peut indiquer plusieurs éléments importés d'un même fichier

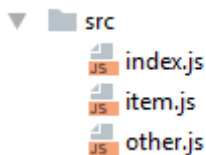
Export de variable/ instance d'objet

```
export var myvar = "abc";
```

Import de variable

```
import {myvar} from './item';
```

Créer un index



« index.js »

Exemple dans « item.js » on exporte plusieurs éléments

```
function fn(name) {
  console.log("hello " + name);
}
var myvar = "abc";
export {fn, myvar};
```

On exporte « tout » des différents fichiers du répertoire

```
export * from './item';
export * from './other';
// ...
```

Il suffira alors de pointer l'import sur le répertoire pour avoir accès à tous les éléments exportés de celui-ci

```
import {fn, doSomething} from './src';

fn("marie");
doSomething();
```

On utilise une fonction de « item.js » et une autre de « other.js »

Browser

Besoin de babel

```
npm i babel-cli babel-preset-latest -D
```

- Avec un loader : SystemJS

Soit générer le code (au format commonjs) avec babel, puis utiliser SystemJS

Installation

```
npm i systemjs -S
```

Générer le code avec babel

Commande ou script NPM

```
"scripts": {
  "build": "babel --presets latest -d dist/ src/"
},
```

Dans la page d'index

```
<script src="node_modules/systemjs/dist/system.js"></script>
<script>
  System.config({
    meta: {
      format: 'cjs'
    }
  });
  System.import('dist/app.js');
</script>
```

On pointe le fichier généré avec babel

Soit utiliser SystemJS pour charger le code es6 avec JSPM qui est le plus simple

Installation

Avec JSPM. Installer [JSPM](#) si ce n'est pas fait

```
npm i jspm -g
```

Puis

```
jspm init
```

Dans la page d'index

```
<script src="jspm_packages/system.js"></script>
<script src="config.js"></script>
<script>
  System.import('src/app.js');
</script>
```

- Avec un bundler : webpack

Installation

```
npm i webpack -D
```

Besoin de babel-loader

```
npm i babel-loader -D
```

« webpack.config.js »

```
module.exports = {
  entry: './src/lib.js',
  output: {
    path: './dist',
    filename: 'bundle.js'
  },
  module: {
    loaders: [{
      test: /\.js$/,
      exclude: /node_modules/,
      loader: 'babel-loader',
      query: {
        presets: ['latest']
      }
    }]
  }
};
```

Générer le bundle

Commande ou script NPM

```
"scripts": {
  "build": "webpack",
  "start": "node server.js"
},
```

Référencer le bundle dans la page d'index

```
<script src="dist/bundle.js"></script>
```


Node

Installation

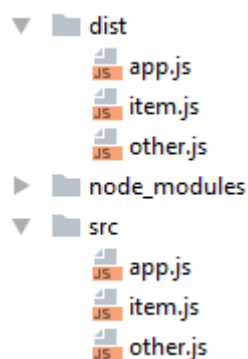
babel

```
npm i babel-cli babel-preset-latest -D
```

Génération des fichiers (format commonjs)

Commande ou script NPM

```
"scripts": {  
  "build": "babel --presets latest -d dist/ src/",  
  "start": "node server.js"  
},
```



Es6 avec Node

Ajouter un fichier « .babelrc »

```
{  
  presets: ['latest']  
}
```

Utiliser la commande « babel-node » pour lancer le serveur

```
"scripts": {  
  "start": "babel-node server.js"  
},
```

e. TypeScript

Browser

Avec un loader : SystemJS

Installation

TypeScript

```
npm i typescript -S
```

SystemJS

```
npm i systemjs -S
```

En dépendance, SystemJS
en ayant besoin

Ecriture de code TypeScript. Exemple

« item.ts »

```
export class Item {
  constructor(public name:string) {}

  doSomething():string {
    return this.name;
  }
}
```

« app.ts »

```
import {Item} from './item';

var item = new Item('Node');
console.log(item.doSomething());
```

Dans la page d'index

```
<script src="node_modules/systemjs/dist/system.js"></script>
<script src="node_modules/typescript/lib/typescript.js"></script>
<script>
  System.config({
    transpiler: 'typescript',
    packages: {
      src: {
        defaultExtension: 'ts'
      }
    }
  });
  System
    .import('src/app')
    .then(null, console.error.bind(console));
</script>
```

Avec un bundler: webpack

Installation

TypeScript

```
npm i typescript -D
```

Webpack

```
npm i webpack -D
```

« awesome-typescript-loader »

```
npm i awesome-typescript-loader -D
```

Si on veut générer la page d'index (avec le bundle référencé)

```
npm i html-webpack-plugin -D
```

« webpack.config.js »

```
var HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {
  entry: './src/app.ts',
  output: {
    filename: 'bundle.js',
    path: __dirname + '/dist'
  },
  devtool: 'source-map',
  resolve: {
    extensions: ['', '.ts', '.js']
  },
  module: {
    loaders: [
      { test: /\.ts$/, loader: 'awesome-typescript-loader' }
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html'
    })
  ]
};
```

Une page « index.html » est générée dans le répertoire « dist » avec le bundle référencé et se servant du template de départ « index.html » de « SRC »

Générer le bundle / page d'index

Commande ou script NPM

```
"scripts": {
  "build": "webpack"
},
```

Référencer le bundle dans la page d'index (si pas d'utilisation de « html-webpack-plugin »)

```
<script src="dist/bundle.js"></script>
```

Sinon ouvrir la page « index.html » de « dist »

4. Tests

- [Mocha](#)

Assertions :

- [should.js](#) « ... should.be ... » (BDD style)
 - `npm install should --save-dev`
- [expect.js](#) « expect.to.be... »
 - `npm install expect.js --save-dev`
- [chai](#) « expect », « assert »
 - `npm install chai --save-dev`
- [autres](#)
 - [Jasmine](#)
 - [JUnit](#)

a. Jasmine "browser"

Download [Jasmine standalone release](#)

b. Mocha "browser"

Installation globale (ou locale au projet)

```
npm i mocha -g
```

Création des fichiers dans le répertoire test par exemple

```
mocha init ./test
```

Puis pour Jasmine ou Mocha

Installer les packages **babel** si utilisation d'es6

```
"devDependencies": {
  "babel-cli": "^6.18.0",
  "babel-core": "^6.20.0",
  "babel-loader": "^6.2.9",
  "babel-preset-latest": "^6.16.0",
  "babel-register": "^6.18.0"
}
```

« .babelrc »

```
{
  "presets": ["latest"]
}
```

Génération d'un bundle avec [Webpack](#)

Install webpack global (or local)

```
npm i webpack -g
```

« webpack.config.js »

```
module.exports = {
  entry: './test/index.js',
  output: {
    path: './test',
    filename: 'specs.js'
  },
  module: {
    loaders: [{
      test: /\.js$/,
      exclude: /node_modules/,
      loader: 'babel-loader'
      //libraryTarget: 'umd'
    }]
  }
}
```

Génération du bundle

```
webpack
```

Référencer ce bundle dans la page « index.html » (spec runner)

Ouvrir dans le navigateur cette page.

c. Karma

Installation globale (ou locale au projet)

```
npm i karma -g
```

Générer « karma.conf.js » avec la commande

```
karma init
```

Choix :

- Framework de test (mocha, jasmine, etc.)
- Launchers (chrome, phantomjs, etc.)
- Files (le bundle webpack ou un index.js avec tous les tests « requis »)

Générer un bundle avec Webpack

Puis lancer les tests avec la commande

```
karma start
```

d. Scripts « package.json »

Avec Mocha (reporter : aspect de rendu des tests, compiler avec babel-register pour es6)

```
"test": "mocha --reporter progress --compilers js:babel-register \"test/spec/*Spec.js\"",
"test:watch": "npm run test -- --watch",
```

Avec Karma

```
"pretest": "webpack",
"test": "karma start",
```

e. Tests e2e

- Nightwatch
- CasperJS

Exemple avec selenium WebDriver (et chrome driver)

```
npm i selenium-webdriver chromedriver -D
```

Ecriture de « tests » (« ./test/e2e/index.js »)

```
var assert = require('assert'),
    test = require('selenium-webdriver/testing'),
    webdriver = require('selenium-webdriver'),
    chrome = require('selenium-webdriver/chrome'),
    By = webdriver.By,
    until = webdriver.until,
    path = require('chromedriver').path;

var service = new chrome.ServiceBuilder(path).build();
chrome.setDefaultService(service);

var baseUrl = "http://localhost/";
/*
doc: http://seleniumhq.github.io/selenium/docs/api/javascript/index.html
*/
test.describe('Hash', function () {
  var driver = new webdriver.Builder()
    .withCapabilities(webdriver.Capabilities.chrome())
    .build();

  this.timeout(15000);
  test.it('Should navigate to home on init', function () {
    driver.get(baseUrl + "spa_router/test/e2e/hash/");
    var pageTitle = driver.findElement(By.tagName('h1'));
    until.elementTextIs(pageTitle, "Home");
  });
  test.it('Allow to navigate', function () {
    driver.findElement(By.linkText("Posts")).click();
    var pageTitle = driver.findElement(By.tagName('h1'));
    until.elementTextIs(pageTitle, "Posts");
    driver.quit();
  });
});
```

Scripts pour lancer les tests (avec Chrome ici)

```
"e2e": "mocha \"test/e2e/index.js\""
```

5. Intégration continue

a. Editor config

Documentation

ajout d'un fichier « .editorconfig » à la racine du projet

```
# editorconfig.org
root = true

[*]
indent_style = space
indent_size = 2
end_of_line = lf
charset = utf-8
trim_trailing_whitespace = true
insert_final_newline = true

[*.md]
trim_trailing_whitespace = false
```

Selon l'éditeur, installer ou non un [plugin](#)

b. Serveur de développement

- http-server
- live-server
- express
- budo
- webpack
- browsersync

Exemple avec Express

```
npm i express -D
```

« server.js »

```
var express = require("express"),
    app = express(),
    path = require('path');

var port = 3000;

app.use(express.static("public"));

app.get('*', function (req, res) {
  res.sendFile(path.join(__dirname, 'index.html'));
});

var server = app.listen(port, function () {
  console.log('api listening on port ', server.address().port);
});
```

c. Automation

- Grunt
- Gulp
- **NPM Scripts**

d. Linter

- jsHint
- JsLint
- **ESLint** ([documentation](#), [formats](#))
- tslint

```
npm i eslint eslint-plugin-import eslint-watch -D
```

Exemple de fichier « eslintrc.json »

```
{
  "root": true,
  "extends": [
    "eslint:recommended",
    "plugin:import/errors",
    "plugin:import/warnings"
  ],
  "parserOptions": {
    "ecmaVersion": 7,
    "sourceType": "module"
  },
  "env": {
    "browser": true,
    "node": true,
    "mocha": true
  },
  "rules": {
    "no-console": 1
  },
  "globals": {
    "define": true,
    "document": true,
    "Promise": true
  }
}
```

Scripts

```
"lint": "esw src --color",
"lint:watch": "npm run lint -- --watch",
```


e. Bundler

Plusieurs possibilités :

- Webpack
- Browserify
- Rollup (« propre »)
- Jspm

Exemple avec rollup

```
"rollup": "^0.21.0",
"rollup-plugin-babel": "^1.0.0",
"uglify-js": "^2.5.0"
```

« /build/build.js »

```
var fs = require('fs');
var zlib = require('zlib');
var rollup = require('rollup');
var uglify = require('uglify-js');
var babel = require('rollup-plugin-babel');
var version = process.env.VERSION || require('./package.json').version;
var banner =
  '/*! \n' +
  ' * AppName v' + version + '\n' +
  ' * (c) ' + new Date().getFullYear() + ' Romagny13\n' +
  ' * Released under the MIT License.\n' +
  ' */';

rollup.rollup({
  entry : './src/appname.js',
  plugins: [
    babel({
      loose: 'all'
    })
  ]
})
.then(function (bundle) {
  return write('dist/appname.js', bundle.generate({
    format: 'umd',
    banner: banner,
    moduleName: 'appname'
  })).code)
})
.then(function () {
  return write(
    'dist/appname.min.js',
    banner + '\n' + uglify.minify('dist/appname.js').code
  )
})
.then(function () {
  return new Promise(function (resolve, reject) {
    fs.readFile('dist/spa.min.js', function (err, buf) {
```

```

    if (err) return reject(err)
    zlib.gzip(buf, function (err, buf) {
      if (err) return reject(err)
      write('dist/appname.min.js.gz', buf).then(resolve)
    })
  })
})
).catch(logError);

function write (dest, code) {
  return new Promise(function (resolve, reject) {
    fs.writeFile(dest, code, function (err) {
      if (err) return reject(err)
      console.log(blue(dest) + ' ' + getSize(code))
      resolve()
    })
  })
};

function getSize (code) {
  return (code.length / 1024).toFixed(2) + 'kb'
};

function logError (e) {
  console.log(e)
};

function blue (str) {
  return '\x1b[1m\x1b[34m' + str + '\x1b[39m\x1b[22m'
};

```

« rollup.config.js » avec format umd

```

import buble from 'rollup-plugin-buble';

export default {
  entry: 'src/spa.js',
  dest: 'dist/spa.js',
  format: 'umd',
  plugins: [ buble() ],
  moduleName: 'spa'
};

```

f. CI server

- [Travis](#)
- Appveyor (windows)
- Jenkins
- Circleci
- Semaphore
- snapci

Exemple avec travis, ajout d'un fichier « .travis.yml »

```
language: node_js
node_js:
  - "6"
```

6. Grunt

Permet d'éviter de devoir faire manuellement les tâches répétitives en les automatisant.

a. Installation en global de grunt

```
npm install -g grunt-cli
```

Installation locale au projet (après avoir navigué jusqu'au projet avec cd ...)

```
npm install grunt --save-dev
```

b. Concat

[Github](#)

```
npm install grunt-contrib-concat --save-dev
```

```
module.exports = function(grunt) {
  grunt.initConfig({
    concat : {
      dist: {
        src: ['src/intro.js', 'src/router.js', 'src/outro.js'],
        dest: 'dist/router.js'
      }
    },
    // autres taches
  });
  // load modules
  grunt.loadNpmTasks('grunt-contrib-concat');
  // define tasks
  grunt.registerTask('default', ['concat', 'uglify', 'browserify']);
};
```

« Intro.js »

```
(function (root, factory) {
  if (typeof define === 'function' && define.amd) {
    // AMD
    define([], factory);
  } else if (typeof module === 'object' && module.exports) {
    module.exports = factory();
  } else {
    // root is window
    root.router = factory();
  }
})(this, function () {
```

Autres fichiers contiennent le code

« outro.js »

```
  return router;
});
```

c. « grunt-browserify »

« grunt-browserify » pour lancer « browserify » (génération du fichier « bundle.js »)

```
npm install grunt-browserify --save-dev
```

Pour lancer

```
grunt
```

... « grunt-contrib-watch » pour mettre à jour automatiquement le fichier « bundle.js »

```
npm install grunt-contrib-watch --save-dev
```

Pour lancer

```
grunt watch
```

A chaque modification et sauvegarde, le bundle sera regénéré.

Pour arrêter « CTRL + C »

« Gruntfile.js »

```
module.exports = function(grunt) {
  // setup config
  grunt.initConfig({
    browserify: {
      app: {
        src: 'src/app.js',
        dest: 'dist/app.bundle.js',
        options: {
          browserifyOptions: {
            debug: true
          }
        }
      }
    },
    watch: {
      app: {
        files: ['src/**/*.js'],
        tasks: ['browserify']
      }
    }
  });

  // load modules
  grunt.loadNpmTasks('grunt-browserify');
  grunt.loadNpmTasks('grunt-contrib-watch');

  // define tasks
  grunt.registerTask('default', ['browserify']);
};
```

d. Serveur local

Si on a besoin d'un serveur local on peut installer « connect » et « livereload » (pour le rafraichissement automatique du browser)

```
npm install grunt-contrib-connect --save-dev
```

```
npm install connect-livereload --save-dev
```

« Gruntfile.js »

```

module.exports = function(grunt) {
  // setup config
  grunt.initConfig({
    browserify: {
      app: {
        src: 'example/app.js',
        dest: 'example/app.bundle.js',
        options: {
          browserifyOptions: {
            debug: true
          }
        }
      }
    },
    watch: {
      app: {
        files: ['example/**/*.js'],
        tasks: ['browserify'],
        options: {
          livereload: true
        }
      }
    },
    connect: {
      app: {
        options: {
          port: 9001,
          base: './',
          middleware: function (connect, options, middlewares) {
            middlewares.unshift(require('connect-livereload')());
            return middlewares;
          }
        }
      }
    }
  });

  // load modules
  grunt.loadNpmTasks('grunt-browserify');
  grunt.loadNpmTasks('grunt-contrib-watch');
  grunt.loadNpmTasks('grunt-contrib-connect');

  // define tasks
  grunt.registerTask('default', ['browserify']);
  grunt.registerTask('serve', ['browserify:app', 'connect:app', 'watch:app']);
};

```

Pour lancer

```
grunt serve
```

On peut se rendre sur <http://localhost:9001/example/>

e. Minimiser les fichiers js avec « Uglify »


[Github](#)

```
npm install grunt-contrib-uglify --save-dev
```

```
module.exports = function(grunt) {
  // setup config
  grunt.initConfig({
    uglify : {
      app:{
        files :{
          'dist/myfile.min.js': ['src/myfile.js']
        }
      }
    },
    // etc.
  });

  // load modules
  grunt.loadNpmTasks('grunt-contrib-uglify');
  // etc.

  // define tasks
  grunt.registerTask('default', ['uglify', 'browserify']);
};
```



f. Pour copier des fichiers

[Github](#)

```
npm install grunt-contrib-copy --save-dev
```

Exemple copier le fichier nom minimisé de « src » dans « dist »

```
module.exports = function(grunt) {
  // setup config
  grunt.initConfig({
    copy: {
      app: {
        files: [
          {expand: true, flatten: true, src: ['src/*'], dest:
            'dist/', filter: 'isFile'}
        ],
      },
    },
  });

  // load modules
  grunt.loadNpmTasks('grunt-contrib-copy');

  // tasks
  grunt.registerTask('default', ['uglify', 'copy', 'browserify']);
};
```

g. Plus

- [Jshint](#)
- [Js code sniffer](#)
- [Cssmin](#)
- [MochaTest](#)

Chercher sur [NPM](#) ou [Gruntjs](#)

7. JS code Sniffer

Pour vérifier que le code respecte les standards d'écritures.

[Site](#)