

---

# Contents

---

<b>Introduction</b> . . . . .	xiii
<b>Part 1. Core JavaScript</b> . . . . .	1
<b>Introduction to Part 1</b> . . . . .	3
<b>Chapter 1. Variables: Declaration, Definition and Type</b> . . . . .	5
1.1. Declarations of functions and variables . . . . .	6
1.1.1. The different declaration keywords . . . . .	6
1.1.2. Lexical scope and definition of a variable according to declaration mode: var, let, const . . . . .	9
1.1.3. Comments (important improvements carried over by ES6) . . . . .	11
1.1.4. General conclusion about the variable declarations in JavaScript. . . . .	11
1.1.5. Naming variables and functions: best practices . . . . .	14
1.2. Variable definition, initialization and typing in JavaScript . . . . .	15
1.2.1. Variables initialization and definition . . . . .	15
1.2.2. Types . . . . .	15
1.2.3. How to use the type “undefined” and the value undefined. . . . .	17
<b>Chapter 2. Controls: Booleans, Branch and Loops</b> . . . . .	19
2.1. Truth values and boolean operators. . . . .	19
2.1.1. Boolean operators: “!” (not), “&&” (and), “  ” (or) . . . . .	19
2.1.2. Relational operators: >, <, >=, <= . . . . .	20
2.1.3. Comparison operators: ==, != (simple) or ===, !== (strict). . . . .	20
2.2. Conditional instructions: branch test, loop test . . . . .	21
2.2.1. Conditional instructions: if ... else, if ... else if ... else . . . . .	21

2.2.2. Ternary conditional operator . . . . .	21
2.2.3. Instruction “switch” . . . . .	22
2.2.4. Classical iteration loop: instruction “for” . . . . .	22
2.2.5. Repeat under condition: instructions “while”, and “do..while” . . . . .	23
2.2.6. Implicit casting of values “undefined” and “null” in boolean context . . . . .	23
2.2.7. Short-cut evaluation: tips for the uncertain definitions . . . . .	24
2.2.8. Exception handling . . . . .	24
<b>Chapter 3. Data: Numbers and Strings . . . . .</b>	<b>27</b>
3.1. Handling numbers . . . . .	28
3.1.1. Literal notation of type “number” variables . . . . .	28
3.1.2. Arithmetic operators . . . . .	29
3.1.3. Math operations using the methods of the object Math. . . . .	30
3.1.4. Evaluation in the “numerical context” versus “boolean context” . . . . .	32
3.2. Handling character strings . . . . .	32
3.2.1. Literal notation of strings . . . . .	32
3.2.2. Backtick syntax, or template syntax, introduced by ES6. . . . .	33
3.2.3. Concatenation operator . . . . .	34
3.2.4. Resolving polymorphism issues with operator + in numerical or string context . . . . .	34
3.2.5. Behavior of the relational and equality operators . . . . .	35
3.2.6. Various facets of string-related issues in a sample application . . . . .	35
3.3. The String.prototype methods . . . . .	37
3.3.1. The need for preprocessing before comparison . . . . .	37
3.3.2. Handling partial comparisons. . . . .	38
3.3.3. Methods for handling strings . . . . .	39
3.3.4. Regular expressions . . . . .	41
3.3.5. Evaluation and uses . . . . .	42
3.3.6. Some examples of useful patterns . . . . .	42
3.3.7. General syntax of a regular expression . . . . .	43
3.3.8. Combining RegExp and String.prototype methods . . . . .	44
<b>Chapter 4. Objects and Prototypes . . . . .</b>	<b>45</b>
4.1. Introduction. . . . .	45
4.2. The objects: concepts versus named entities . . . . .	46
4.3. Object literal notation in JavaScript . . . . .	47
4.3.1. Syntax for “object literal”: . . . . .	47
4.3.2. Important warnings about writing JavaScript object notation . . . . .	48
4.3.3. The object literal first use: to define an object type variable. . . . .	49
4.3.4. The object literal second use: data notation in JSON format . . . . .	49

---

4.3.5. Accessing the individual properties of an object . . . . .	50
4.3.6. Notation syntax evolution with ES6 . . . . .	51
4.4. The builtin methods of Object and Object.prototype . . . . .	51
4.4.1. The methods of Object, Object.prototype, and JSON. . . . .	51
4.4.2. Create an object and specify its properties . . . . .	53
4.4.3. Syntax and usage of the “descriptor” property. . . . .	53
4.4.4. Listing the properties of an object, analyzing a literal . . . . .	54
4.5. Basics of the “prototypal approach” in JavaScript . . . . .	56
4.5.1. JavaScript object's fundamental relation: “has prototype” . . . . .	57
4.5.2. Role of the prototypes and inheritance mechanism . . . . .	58
4.5.3. Object construction: the “literal approach” . . . . .	60
4.5.4. Object construction: the “prototypal approach” . . . . .	61
4.5.5. The pattern “assign/create” . . . . .	62
4.5.6. Object construction: the “classical approach” . . . . .	63
4.6. Comparing “prototypal” and “classical” approaches . . . . .	64
4.6.1. Simulating a class hierarchy in JavaScript . . . . .	65
4.6.2. Summing up what we learned so far. . . . .	68
<b>Chapter 5. Arrays</b> . . . . .	<b>71</b>
5.1. Handling arrays: creation and access to its elements. . . . .	72
5.1.1. Creating an array with the array literal notation . . . . .	72
5.1.2. Checking if a variable is an array . . . . .	72
5.1.3. The length property, the index count . . . . .	73
5.1.4. Accessing individual values in an array: the indices . . . . .	74
5.2. Methods of the object Array and Array.prototype . . . . .	74
5.2.1. The “Mutators” family. . . . .	75
5.2.2. The “Accessors” family . . . . .	77
5.2.3. The “Iteration” family . . . . .	78
5.2.4. Iterating over the elements of an array . . . . .	78
5.2.5. Iteration without a loop, with Array/Array.prototype methods . . . . .	79
5.2.6. Chaining array methods . . . . .	81
5.2.7. Arrays and the arrow function syntax . . . . .	82
5.2.8. The “Iterables” . . . . .	83
5.3. Array of arrays (multidimensional array) . . . . .	83
5.3.1. Frameworks proposing an “augmented Array.prototype” . . . . .	85
<b>Chapter 6. Functions</b> . . . . .	<b>87</b>
6.1. General syntax of a JavaScript function . . . . .	88
6.1.1. Name . . . . .	88
6.1.2. Parameters . . . . .	88
6.1.3. Return. . . . .	89
6.1.4. Function code block and scope . . . . .	89

6.1.5. Creating functions . . . . .	89
6.2. Invoking a function with operator (.) . . . . .	90
6.2.1. The three facets of the “parentheses operator” in a function context . . . . .	91
6.3. Choosing function declaration versus function expression . . . . .	92
6.4. Arguments . . . . .	93
6.4.1. The arguments are passed by value . . . . .	93
6.4.2. The inner object “arguments”. . . . .	94
6.5. Scope: global scope, function scopes and block scopes . . . . .	94
6.5.1. Vocabulary: lexical scope and “namespace”. . . . .	94
6.5.2. Wrapping-up and warnings . . . . .	98
6.6. Function “closures” . . . . .	101
6.6.1. Saving the value of a free variable in a given context . . . . .	102
6.6.2. Creating a list of functions linked to an array of data. . . . .	103
6.6.3. “Currying”: breaking down a function into 1-parameter functions .	106
6.6.4. Compositing functions from an array of functions . . . . .	107
6.7. Immediately invocable functions: IIFE . . . . .	109
6.7.1. Creating a “namespace”, or a named library, with an IIFE . . . . .	109
6.8. The methods of Function.prototype . . . . .	110
6.8.1. Function.prototype.call() and .apply(), and pronoun 'this' . . . . .	112
6.8.2. Function.prototype.bind() . . . . .	112
6.9. Built-in functions . . . . .	113
6.10. Closure and IIFE cheat-sheet. . . . .	114
<b>Chapter 7. From Signs to Patterns . . . . .</b>	<b>117</b>
7.1. Reserved words . . . . .	118
7.2. The pronoun “this” . . . . .	119
7.2.1. The many ways to link the pronoun “this” . . . . .	119
7.2.2. How to explicitly bind the pronoun? . . . . .	121
7.3. Operator: new . . . . .	121
7.4. Punctuation signs . . . . .	122
7.5. JavaScript usual design patterns. . . . .	123
7.5.1. Programming idioms . . . . .	124
7.5.2. Creational pattern: “Assign/Create Combo” . . . . .	125
7.5.3. Structural pattern: singleton or namespace pattern . . . . .	127
7.5.4. Another structural pattern: the Decorator pattern . . . . .	128
7.5.5. Behavioral pattern: the observer or publish/subscribe pattern. . . . .	130
7.6. Metaprogramming with ES6. . . . .	131
7.6.1. “Reflection” by “Symbols” . . . . .	131
7.6.2. New tool for measuring code performance. . . . .	131

---

<b>Part 2. Client-Side JavaScript</b> . . . . .	133
<b>Introduction to Part 2</b> . . . . .	135
<b>Chapter 8. JavaScript in the Web Page</b> . . . . .	137
8.1. Ecosystem of the web page: the HTML sequence . . . . .	137
8.1.1. Structure and semantics/layout and presentation . . . . .	137
8.1.2. Reminder about HTML5 tags. . . . .	138
8.2. Building the web page DOM: the layout engine . . . . .	140
8.2.1. DOM tree built by the layout engine: selecting nodes via CSS . . . . .	141
8.2.2. CSS rules and relationship with JavaScript selection methods . . . . .	142
8.3. Dynamic behavior of the web page: the script engine . . . . .	143
8.4. Interface with the DOM . . . . .	145
8.4.1. DOM interface 1: selecting elements . . . . .	145
8.4.2. DOM interface 2: reading/writing/creating an element. . . . .	146
8.4.3. Methods for HTML DOM document and element prototypes . . . . .	148
8.5. The events in client side JavaScript. . . . .	150
8.5.1. The browser event loop . . . . .	150
8.5.2. Handling DOM events. . . . .	151
8.6. Interacting with the DOM: to link elements/events . . . . .	153
8.6.1. Waiting for the DOM . . . . .	153
8.6.2. Example: to build an HTML list . . . . .	153
8.6.3. Using events: modifying attributes and class names of an element. . . . .	154
8.6.4. Dispatching events, creating a CustomEvent. . . . .	155
<b>Chapter 9. Graphic and Multimedia Tools</b> . . . . .	157
9.1. To draw in the web page . . . . .	157
9.1.1. The elements <figure> and <canvas> . . . . .	158
9.1.2. 2D curve plot . . . . .	158
9.2. SVG language . . . . .	161
9.3. Handling time in graphics animation. . . . .	163
9.3.1. Methods setTimeout, setInterval, requestAnimationFrame . . . . .	163
9.3.2. Performance considerations, generator functions . . . . .	165
9.4. Data persistence between client sessions . . . . .	166
9.4.1. Http cookies . . . . .	166
9.4.2. Local storages . . . . .	167
9.5. Note about “JavaScript frameworks” (jQuery, d3, etc.) . . . . .	168
9.5.1. A few words about jQuery . . . . .	168
9.5.2. Recommendation . . . . .	169

---

<b>Chapter 10. AJAX Technology (Asynchrony)</b> . . . . .	171
10.1. Architecture for client–server data exchange . . . . .	171
10.1.1. The object XMLHttpRequest . . . . .	172
10.1.2. Using XMLHttpRequest: several steps . . . . .	172
10.2. Remarks about HTTP . . . . .	173
10.3. “Promises” and asynchronous programming . . . . .	173
10.3.1. Example: promisifying XMLHttpRequest . . . . .	174
10.3.2. Chaining promises . . . . .	175
10.3.3. Parallel processing of several promises . . . . .	175
10.3.4. Fetch: the promise to fetch AJAX . . . . .	176
10.3.5. About the “Same Origin Policy” . . . . .	177
10.4. The exchange format: JSON . . . . .	177
10.4.1. A very useful application of JSON: converting data from a spreadsheet . . . . .	178
10.4.2. Exporting spreadsheet data into JSON format . . . . .	179
10.4.3. Differences between JSON and the Javascript object Notation . . . . .	182
10.5. JavaScript Object Notation with Padding . . . . .	184
10.6. A parallel JavaScript: the “worker” . . . . .	185
<b>Part 3. Applications</b> . . . . .	187
<b>Introduction to Part 3</b> . . . . .	189
<b>Chapter 11. Chronological Data</b> . . . . .	191
11.1. Accessing a JSON file via Ajax . . . . .	191
11.1.1. Quick presentation of the Quandl API . . . . .	191
11.1.2. Processing an example with promises . . . . .	192
11.2. Using open source graphic libraries . . . . .	195
11.2.1. Plot multiple data series against the same time axis . . . . .	195
11.2.2. Dynamic plot: simulating time evolution . . . . .	197
<b>Chapter 12. Relational Data</b> . . . . .	199
12.1. Aggregating tabulated JSON data . . . . .	199
12.1.1. Electoral data: administrative breakdown, political breakdown . . . . .	200
12.1.2. Aggregating data along the spatial dimension: votes by circonscription . . . . .	203
12.1.3. Aggregating data along the affiliations dimension: labels by candidate . . . . .	205
12.2. Joining data: multiple JSON files . . . . .	207

---

12.2.1. Advantage of the flexibility brought by the prototypal approach . . . . .	207
12.2.2. Coding the join on the electoral application . . . . .	208
12.3. Postprocessing: analysis . . . . .	210
12.3.1. Analyzing the affiliations . . . . .	210
12.4. The role of promises . . . . .	211
12.4.1. Performance considerations with the electoral application . . . . .	213
12.5. Using Google Gantt chart for a graphic visualization . . . . .	214
<b>Chapter 13. Cartographic Data . . . . .</b>	<b>217</b>
13.1. Cartographic application: using cartographic libraries . . . . .	217
13.1.1. Preparation of the map . . . . .	219
13.1.2. Creating a layer of markers . . . . .	220
13.1.3. Interacting and selecting features . . . . .	222
13.2. SVG-based cartography . . . . .	222
13.2.1. Description of the application . . . . .	223
13.2.2. Embedding the whole SVG document by direct copy . . . . .	224
13.2.3. Embedding the SVG code, element by element . . . . .	225
13.2.4. Joining relational data and SVG data . . . . .	225
13.2.5. Processing the combined information . . . . .	226
13.3. Getting coordinates from Wikipedia pages . . . . .	227
<b>Chapter 14. Data Served by JSONP . . . . .</b>	<b>229</b>
14.1. Serving RSS feeds through <i>Yahoo Query Language</i> . . . . .	229
14.2. Serving shared spreadsheets through Google spreadsheets . . . . .	231
14.2.1. Client-side code: HTML and script of the callback function . . . . .	231
14.2.2. Server-side code under the GoogleScript global object . . . . .	232
14.3. Serving images and their metadata through the Flickr API . . . . .	233
<b>Bibliography . . . . .</b>	<b>235</b>
<b>Index . . . . .</b>	<b>239</b>