How to add retro-compatibility, an example

INTRODUCTION

A polyfill is a piece of JavaScript code you include in your HTML page, that will emulate HTML5 features not yet implemented in browsers. Usually a polyfill starts emulating a feature only when native support is not detected.

You can find a list of polyfills on http://caniuse.com/ or on http://html5please.com/ Web sites, and more generally if you look for the name of the feature and add the "polyfill" search keyword to your Google, Bing, Yandex etc. request.

Some librairies such as http://modernizr.com/ are packed with a lot of polyfills that will be loaded on demand only when the browser is not supporting a native feature.

EXAMPLE WITH < DETAILS > AND < SUMMARY >

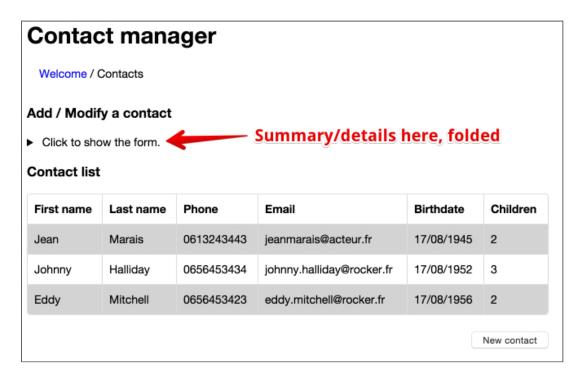
<details> and <summary> are still not supported by all browsers. How can we use
them in a Web site? The answer is: use a polyfill!

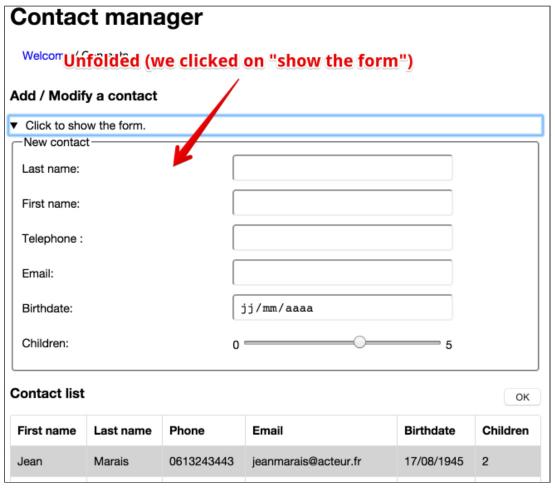
Version 1: no polyfill

We wrote a small contact manager application (just the layout, it does not really work), composed of a form and a table for displaying contacts. The form is in a <details>... <summary> elements.

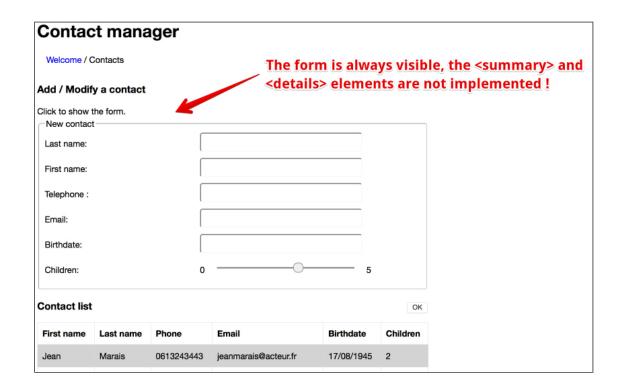
Online version: http://jsbin.com/tupuxo/1/edit

Result on Google Chrome (which does support summary/details):





On Firefox or Internet Explorer the form is always visible, as <summary> and <details>are not supported yet:



Version 2 with a light polyfill

Online version (try it now on Internet Explorer and Firefox, it works as expected as summary/details are at this time emulated by JavaScript code): http://jsbin.com/tupuxo/7/edit

Explanations: we used a light polyfill, the one indicated by caniuse.com (https://gist.github.com/remy/370590):



The link points to the JavaScript code you must include in your Web page. But, in the comments of the polyfill page, you will find an online example at jsbin.com:



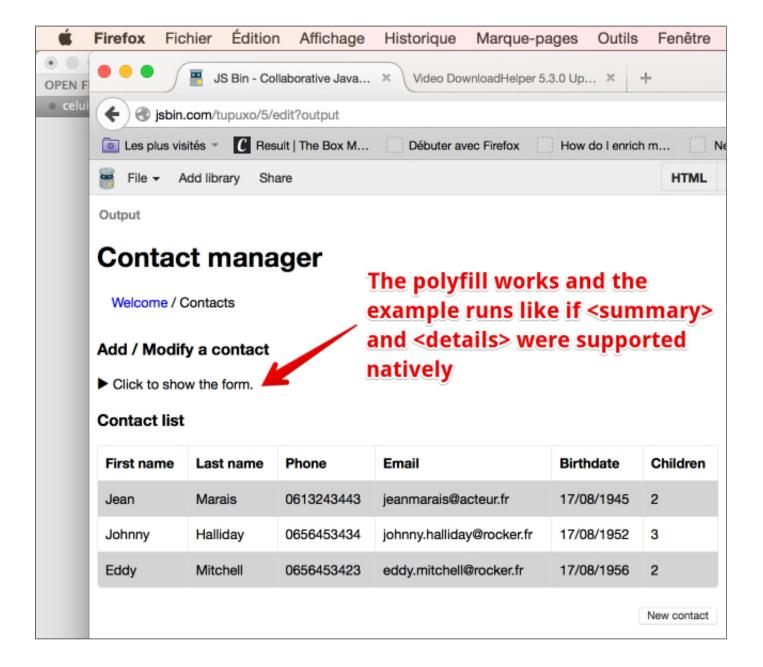
You follow the link to see a running example, or directly copy and paste the code from the main page, then include it in your Web page using the <script> element:

Either use <script src="mypolyfill.js"> (in this case put the polyfill in a filemypolyfill.js>) or if you want to try it on jsbin.com, just copy it into the jsBin script tab, or include it directly between <script>...</script>. This is what we did:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Contact manager</title>
<script>
    (function(e,t){function r(e){var t=null;... // POLYFILL CODE HERE!
    ...
    </script>

10. </head>
...
```

Now the example works on FireFox and Internet Explorer! This is a screenshot taken with FireFox:



Version 3 with a "heavy" polyfill

UPDATE! WARNING: this version apparently does not work anymore on IE. We contacted the authors of the polyfill as it used to work not so long ago. In between, try on IE the previous and next polyfills, they do work. This one is anyway an "heavy" one we do not really recommend. It was here to show how retro compatibility can become an heavy task. Ifyou have retro compatibilities constraints for a commercial project, and if your customer asks for old IE compatibility, do not forget to charge him extras for that!

Actually, the previous version works on recent browsers that do not implement<summary> and <details>, but not on "old" browsers such as Internet Explorer 7. Some "heavy" polyfills exist and add support even to these older browsers and provide a better JavaScript API and CSS styling emulation.

This time we searched on the Web for the most complete polyfill and found this one: https://github.com/termi/Element.details. The Web site includes a demo, we copied and pasted the required code from there.

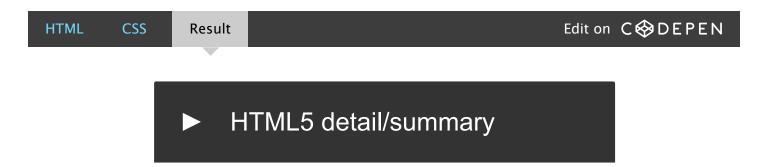
Here is the online version: http://jsbin.com/nogate/1/edit, try it if you can, with an old browser (even older than 2012).

Look at the source code (the beginning of the HTML file), you will see why we call it "heavy";-)

Version 4 with a polyfill that is 100% accessible

The previous version worked well but requires a mouse to open the accordion. The summary / details elements have been designed to be accessible, thus usable only with a keyboard too. This example sent to me by Steve Faulkner is 100% correct in terms of functionality and accessibility! **It's the recommended choice!**

Original example of use of the polyfill:



The contact manager form with this polyfill:

HTML CSS Result Edit on C DEPEN