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.

> Do you know where the "polyfill" term is coming from? It was coined by Remy Sharp, a British Web programmer who had been using a brand of resin to fill holes in masonry.
>
>

You will find a list of polyfills on these Web sites:http://caniuse.com/ or http://html5please.com/. Or just do a search using the keyword 'polyfill' with the name of the feature.

Some libraries, such as http://modernizr.com/, are packed with many 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):

Contact manager

Welcome **Unfolded (we clicked on "show the form")**

Add / Modify a contact

▼ Click to show the form.
┌─New contact──────────────────────────────────┐
│ Last name: [   ] │
│   │
│ First name: [   ] │
│   │
│ Telephone :   [   ] │
│   │
│ Email: [   ] │
│   │
│ Birthdate: [jj/mm/aaaa   ] │
│   │
│ Children: 0 ────────○──────── 5   │
└───┘

Contact list   [OK]

First name	Last name	Phone	Email	Birthdate	Children
Jean	Marais	0613243443	jeanmarais@acteur.fr	17/08/1945	2

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

Contact manager

Welcome / Contacts   **The form is always visible, the <summary> and <details> elements are not implemented !**

Add / Modify a contact

Click to show the form.
┌─New contact─────────────────────────┐
│ Last name: [   ] │
│ First name: [   ] │
│ Telephone :   [   ] │
│ Email: [   ] │
│ Birthdate: [   ] │
│ Children: 0 ──────○────── 5   │
└───────────────────────────────────┘

Contact list   [OK]

First name	Last name	Phone	Email	Birthdate	Children
Jean	Marais	0613243443	jeanmarais@acteur.fr	17/08/1945	2

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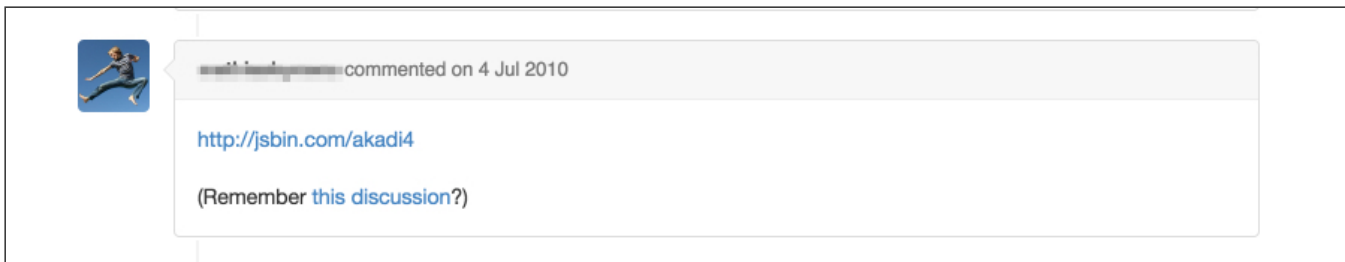
You can follow the link to see a running example, or directly copy and paste the code from the main page, then include it in your Web page using the `<script>` element:

Either use `<script src="mypolyfill.js">` (in this case put the polyfill in a file `mypolyfill.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:

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

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

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

Here is another example with a more complicated case (nested summary/details), that shows that this polyfill does not handle 100% perfectly nested elements.

Version 3: with a polyfill that is 100% accessible

The summary / details elements have also been designed to be accessible, thus usable with a keyboard only. This example sent to me by Steve Faulkner is 99% correct in terms of functionality (small problem of alignment with nested summary/details) and accessibility! **It's the recommended choice!**

Original example of use of the polyfill:

►   HTML5 detail/summary

►   2nd top-level `<detail>` element

►   3rd top-level `<detail>` element

The contact manager form with this polyfill: