

# Browser compatibility

## Modern Frameworks

Nowadays, modern frameworks have been dropping support for old browsers, such as explorer 8 or 9. This guide would help you to improve the compatibility of your webs. Here exists some compatibility table about modern technologies

### Angular 1

Angular supports all browsers with support ES5 Methods, including Internet Explorer 9 and above.

Compatibility	IE8	IE9	IE10	IE11+	Chrome*	Firefox*	Safari 10+*
Without Polyfill	*	✓	✓	✓	✓	✓	✓

\*Since 1.3 version Angular has no support for IE8. You can use IE8 with v1.2 and previous versions.

### Angular 2

Compatibility	IE8	IE9	IE10	IE11+	Chrome*	Firefox*	Safari 10+*
Without Polyfill	~	~	✓	✓	✓	✓	✓

### React

React supports all browsers with support ES5 Methods, including Internet Explorer 9 and above.

Compatibility	IE8	IE9	IE10	IE11+	Chrome*	Firefox*	Safari 10+*
Without Polyfill	*	✓	✓	✓	✓	✓	✓
With Polyfill	✓	✓	✓	✓	✓	✓	✓

\*Since v15 version REACT DOM has no support for IE8. You can use IE8 with v14 and previous versions.

Polyfills: - (Optional) Add support for ES5 in old browsers: <https://github.com/es-shims/es5-shim>

### Polymer

Due to the lack of support for some modern browsers, is mandatory to include polyfills in order to add support for every feature of polymer

Without the polyfills, Polymer works in these browsers:

Without Polyfill	IE8	IE9	IE10	IE11+	Chrome*	Firefox*	Safari 10+*
Custom Elements	~	~	~	~	✓	~	~
HTML Imports	~	~	~	~	✓	~	~
Shadow DOM	~	~	~	~	✓	~	~
Templates	~	~	~	~	✓	✓	✓
With Polyfill	IE8	IE9	IE10	IE11+	Chrome*	Firefox*	Safari 7+*
Custom Elements	~	~	~	✓	✓	✓	✓
HTML Imports	~	~	~	✓	✓	✓	✓
Shadow DOM	~	~	✓	✓	✓	✓	✓

Polyfills: - (Mandatory) webcomponents: <https://github.com/WebComponents/webcomponentsjs> \* (Recommended) webcomponents-lite.js:  
Add shady dom in order to work with Polymer \* webcomponents.js: Add full implementation of Shadow Dom

## Adding Shims or Polyfills

We recommend conditionally loading the polyfills in your application: using the server or feature-detecting on the client whether the browser supports the missing features natively, and then only loading the polyfills if it doesn't. An advantage of using standards-based features is the payload necessary to run your application will continue to decrease as browsers implement the standard.

This is an example loading conditionally the Polymer polyfills

```
(function() {  
  if ('registerElement' in document  
      && 'import' in document.createElement('link')  
      && 'content' in document.createElement('template')) {  
    // platform is good!  
  } else {  
    // polyfill the platform!  
    var e = document.createElement('script');  
    e.src = '/bower_components/webcomponentsjs/webcomponents-lite.min.js';  
    document.body.appendChild(e);  
  }  
})();
```