Trending javascript libraries – Popper.js

Marc Antoni Román Martínez

DAM
Interface Development

Contenido

Introduction

Popper.js is a library used to position poppers in web applications. A popper is an element on the screen which "pops out" from the natural flow of your application. Common examples of poppers are: tooltips, popovers or drop-downs.

This library is useful to give more details or information to the users about an element like can be a button, an input from a formulary,… For example, we have a formulary with different checkbox and the user doesn't understand what's mean one checkbox, then we can implement a tooltip with a click or hover event to show to the user details about the checkbox.

Important information about the library

Popper.js is a positioning engine, its purpose is to calculate the position of an element to make it possible to position it near a given reference element. The engine is completely modular and most of its features are implemented as modifiers. We can customize the poppers completely thanks to the modifiers based structure.

The library Popper.js doesn't depend on any other library like jQuery or Bootstrap. This library is used by big companies like Twitter in Bootstrap v4 or Microsoft in WebClipper.

Lots of users need a simple way to implement powerful tooltips in their projects, for this reason, the creators of Popper.js created Tooltip.js. This library is a small library that makes it easy to automatically create tooltips using as engine Popper.js. Its functionality is like to the famous tooltip system of Bootstrap, in this way it will be easy to integrate it in your projects.

Installation of the library

To install the library Popper.js or Tooltip.js is easy, we have different ways to install it. In the next images we can see the ways to do it:

Popper.js:

Source	
npm	`npm install popper.js --save`
yarn	`yarn add popper.js`
NuGet	`PM> Install-Package popper.js`
Bower	`bower install popper.js --save`
unpkg	`https://unpkg.com/popper.js`
unpkg, minified	`https://unpkg.com/popper.js/dist/umd/popper.min.js`

Ways to install Popper.js

Tooltip.js:

Source	
npm	`npm install tooltip.js --save`
yarn	`yarn add tooltip.js`
Bower*	`bower install tooltip.js=https://unpkg.com/tooltip.js --save`
unpkg	`https://unpkg.com/tooltip.js`
unpkg, minified	`https://unpkg.com/tooltip.js/dist/umd/tooltip.min.js`

Ways to install Tooltip.js

Now, we are going to focusing in the installation with npm. To install it with npm we only have to insert the command "npm install popper.js/tooltip.js --save" in our project directory. Once we have executed this command, we only have to reference the library in our html file like this:

```
<script src="node_modules\popper.js\dist\umd\popper.min.js"></script>
```

Reference to the library Popper.js.

Once we have done it, we can use it now.

Usage

First of all, we are going to explain a basic concepts to use it. To use this library we need two elements in the html file: the reference element, which will be pointed, and the popper, which will contain the information. Once, we have these two elements, we can create the popper introducing the previous elements in the function popper. In the following javascript code we can see how create the popper:

```
var reference = document.querySelector('.my-button');
var popper = document.querySelector('.my-popper');
var anotherPopper = new Popper(reference, popper, {
  // popper options here
});
```

Javascript code to create a popper.

In the Popper function we can insert different option to customize our popper. There are many options like:

-Placement of the popper respect to the button.
-Behavior of it when it is near of the borders of the container.
-Hide option to hide the popper.
-PreventOverflow: preventing the popper from positioned outside the boundary.

We can see a few examples of its usage in the official web page of popper.js (https://popper.js.org/). But now, we are going to do our own example following the next steps:

-The first step is create our html, this html will be simple. In our html file we will include the library popper.js and the library jQuery, our css and js libraries, that we are going to create, too. In the body of the html file, we will put a button which will be show information when it's pressed.

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Popper</title>
    <meta charset=UTF-8>
    <meta property="og:description" content="Noticias, resultados y estadísticas de fútbol español e internacional">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel='stylesheet' href='css/css.css'>



    <script src="node_modules\popper.js\dist\umd\popper.min.js"></script>
    <script src="node_modules\jquery\dist\jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

    <script src="js/js.js"></script>

</head>
<body>
    <button class="myReference" type="button">Click Me!</button>

</div>
</body>
</html>
```

Our html file

-The next step is create the javascript file. When the document is ready, our button will be saved in a variable called reference, and when the button is pressed will active the event which will show our popper. In the event, we are going to create a span with the information of the popper and will add to the body. Once it is added, we are going to call the function of the popper library to create the popper. Inside the function, we will customize our popper with different options like placement and behavior.

```javascript
$( document ).ready(function() {

    var reference = document.querySelector('.myReference');

    $(reference).click(function() {

        var world = $('<span class="popper">World</span>');
        $('body').append(world);

        var popper = new Popper(reference, world, {

            placement: 'right',
            modifiers: {

                flip: {behavior: ['right', 'bottom', 'top']},

            },

        });

    });
});
```

Creating a popper with Javascript with the Popper.js.

-Finally, we have already our popper. Now we can customize it more with the css file. In our case this is our css file:

```css
.myReference{
  position: fixed;
  top:300px;
  left:300px;
}
.popper {
  background-color: rgba(255, 193, 7, 1);
  padding: 10px;
  text-align: center;

}

.popper[x-placement^="top"] {
  margin-bottom: 5px;

}
.popper[x-placement^="bottom"] {
  margin-top: 5px;
  .popper__arrow {
    width: 0;
    height: 0;
    border-style: solid;
    position: absolute;
    margin: 5px;
  }

}
.popper[x-placement^="right"] {
  margin-left: 5px;

.popper[x-placement^="left"] {
  margin-right: 5px;
}
```

CSS code of the popper example.

The result of this example is this:

Result of the popper example.

Conclusions:

My conclusion is that the library is useful and fast when you know how to use it. We can create poppers quickly with a simple way. But the bad side of this library, in my opinion, is that it is complicated to use and learn for a beginner due to there aren't enough examples about how to use it and the documentation doesn't help.

Webgraphy

-Website of popper.js: https://popper.js.org/

-Github of popper.js: https://github.com/FezVrasta/popper.js

-Redstapler's example: https://redstapler.co/popper-js-tutorial-manage-popup-like-pro/