






آموزش پلاگین TagSort.JS در JavaScript

Tagsort.js افزونه ای جهت نمایش تگ ها و فیلتر کردن ایتما براساس تگ میباشد

PHP JavaScript React Memcached RocksDB Cassandra Flux MySQL jQuery Scala Rails Hadoop Redis Node.js Python
Java Objective-C PostgreSQL MongoDB nginx Ruby SASS Go .NET

 Facebook PHP / JavaScript / React / Memcached / RocksDB / Cassandra / Flux	 Twitter MySQL / JavaScript / jQuery / Memcached / Scala / Rails / Hadoop / Redis	 Netflix MySQL / Node.js / Python / JavaScript / React / Java / Cassandra	 Uber MySQL / Node.js / Python / Java / Objective-C / PostgreSQL / Redis / MongoDB	 Dropbox MySQL / Python / Memcached / nginx
---	---	---	---	--

در ابتدا لینک فایل های مورد نیاز را در پروژه قرار میدهم

```
<link href = "styles/tagsort.min.css" rel = "stylesheet" type = "text/css"></link>
```

```
<script src = "scripts/jquery-2.1.3.min.js"
```

```
  type = "text/javascript" charset = "utf-8">
```

```
</script>
```

```
<script src = "scripts/tagsort.min.js"
```

```
  type = "text/javascript" charset = "utf-8">
```

```
</script>
```

از CSS سفارشی برای استایل دهی به itemها استفاده میکنیم.

```
html,body {  
  margin: 0;  
  padding: 0;  
}
```

```
body {  
  background-color: #FFF;  
  font-family: "HelveticaNeue-Light", "Helvetica Neue Light",  
  "Helvetica Neue", Helvetica, Arial, "Lucida Grande", sans-serif;  
  font-weight: 300;  
}
```

```
.container {
```

```
width: 80%;
margin: 0 auto;
}

h1 > small a {
  color: #AAA;
  text-decoration: none;
  font-size: 70%;
  margin-left: 10px;
}

h1 > small a:hover {
  color: #000;
}

.item {
  box-sizing: border-box;
  float: left;
  position: relative;
  min-height: 1px;
  padding-left: 15px;
  padding-right: 15px;
  width: 20%;
  margin-bottom: 40px;
  height: 360px;
  max-height: 360px;
  overflow: hidden;
}

.item .wrapper {
  background-color: #f4f4f4;
  padding: 8px;
  height: 100%;
}

.item .wrapper img {
  width: 100%;
}

.item .wrapper .item-tags {
  color: #AAA;
}
```

```

font-size: 12px;
line-height: 1.8;
}

.tagsort-tags-container {
margin: 40px 0;
}

```

در قسمت HTML، ایتیم‌ها را درون یک عنصر والد قرار می‌دهیم و به هر کدام کلاس `item` را می‌دهیم تا بتوانیم از آنها برای اعمال افزونه استفاده کنیم. هر `item` با استفاده از پارامترهای زیر اطلاعات شماره عنصر و تگ‌های آن را برای افزونه ارسال می‌کند.

```
data-item-id = "1" data-item-tags = "PHP,JavaScript,React,Memcached,Rocks"
```

```

<div class = "container">

  <div class = "tags-container row"></div>

  <div class = "item col-md-3" data-item-id = "1"
    data-item-tags = "PHP,JavaScript,React,Memcached,RocksDB,Cassandra,Flux">
    <div class = "wrapper">
      <img src = "images/facebook.jpg" alt = "Facebook">
      <h2>Facebook</h2>
      <p class = "item-tags"></p>
    </div>
  </div>

  <div class = "item col-md-3" data-item-id = "2"
    data-item-tags =
  "MySQL,JavaScript,jQuery,Memcached,Scala,Rails,Hadoop,Redis">
    <div class = "wrapper">
      <img src = "images/twitter.jpg" alt = "Twitter">
      <h2>Twitter</h2>
      <p class = "item-tags"></p>
    </div>
  </div>

  <div class = "item col-md-3" data-item-id = "3"
    data-item-tags = "MySQL,Node.js,Python,JavaScript,React,Java,Cassandra">

```

```
<div class = "wrapper">
  <img src = "images/netflix.jpg" alt = "Netflix">
  <h2>Netflix</h2>
  <p class = "item-tags"></p>
</div>
</div>
```

```
<div class = "item col-md-3" data-item-id = "4"
  data-item-tags = "MySQL,Node.js,Python,Java,Objective-
C,PostgreSQL,Redis,MongoDB">
  <div class = "wrapper">
    <img src = "images/uber.jpg" alt = "Uber">
    <h2>Uber</h2>
    <p class = "item-tags"></p>
  </div>
</div>
```

```
<div class = "item col-md-3" data-item-id = "5"
  data-item-tags = "MySQL,Python,Memcached,nginx">
  <div class = "wrapper">
    <img src = "images/dropbox.jpg" alt = "Dropbox">
    <h2>Dropbox</h2>
    <p class = "item-tags"></p>
  </div>
</div>
```

```
<div class = "item col-md-3" data-item-id = "6"
  data-item-tags = "Python,Java,Cassandra,Hadoop,PostgreSQL,Hadoop">
  <div class = "wrapper">
    <img src = "images/spotify.jpg" alt = "Spotify">
    <h2>Spotify</h2>
    <p class = "item-tags"></p>
  </div>
</div>
```

```
<div class = "item col-md-3" data-item-id = "7"
  data-item-tags = "MySQL,Javascript,jQuery,Ruby,Redis,nginx,Rails,SASS">
  <div class = "wrapper">
    <img src = "images/kickstarter.jpg" alt = "Kickstarter">
    <h2>Kickstarter</h2>
    <p class = "item-tags"></p>
  </div>
```

```

</div>

<div class = "item col-md-3" data-item-id = "8"
  data-item-tags = "Go,Ruby,MySQL,Redis,Memcached,SASS,Rails,nginx">
  <div class = "wrapper">
    <img src = "images/digitalocean.jpg" alt = "DigitalOcean">
    <h2>DigitalOcean</h2>
    <p class = "item-tags"></p>
  </div>
</div>
</div>

```

در نهایت کد زیر را درون تگ script قرار می‌دهیم. که اعمال افزونه روی عنصر نگهدارنده item هاست. در option های این افزونه مشخص می‌کنیم جهت فیلتر کردن روی عناصر با کلاس item تمرکز کند. هر تگی که در این عناصر یافت درون تگ span قرار دهد. و نتایج پیدا کردن تگ های هر عنصر با کلاس item را در عنصری با کلاس item-tags در همان عنصر قرار دهد. همینطور سرعت حرکت عناصر را نیز بر حسب میلی ثانیه مشخص می‌کنیم.

```

$(function(){
  $('div.tags-container').tagSort({selector:'.item', tagWrapper:'span',
  displaySelector: '.item-tags', displaySeperator: ' / ', inclusive:
  false, fadeTime:200});

```