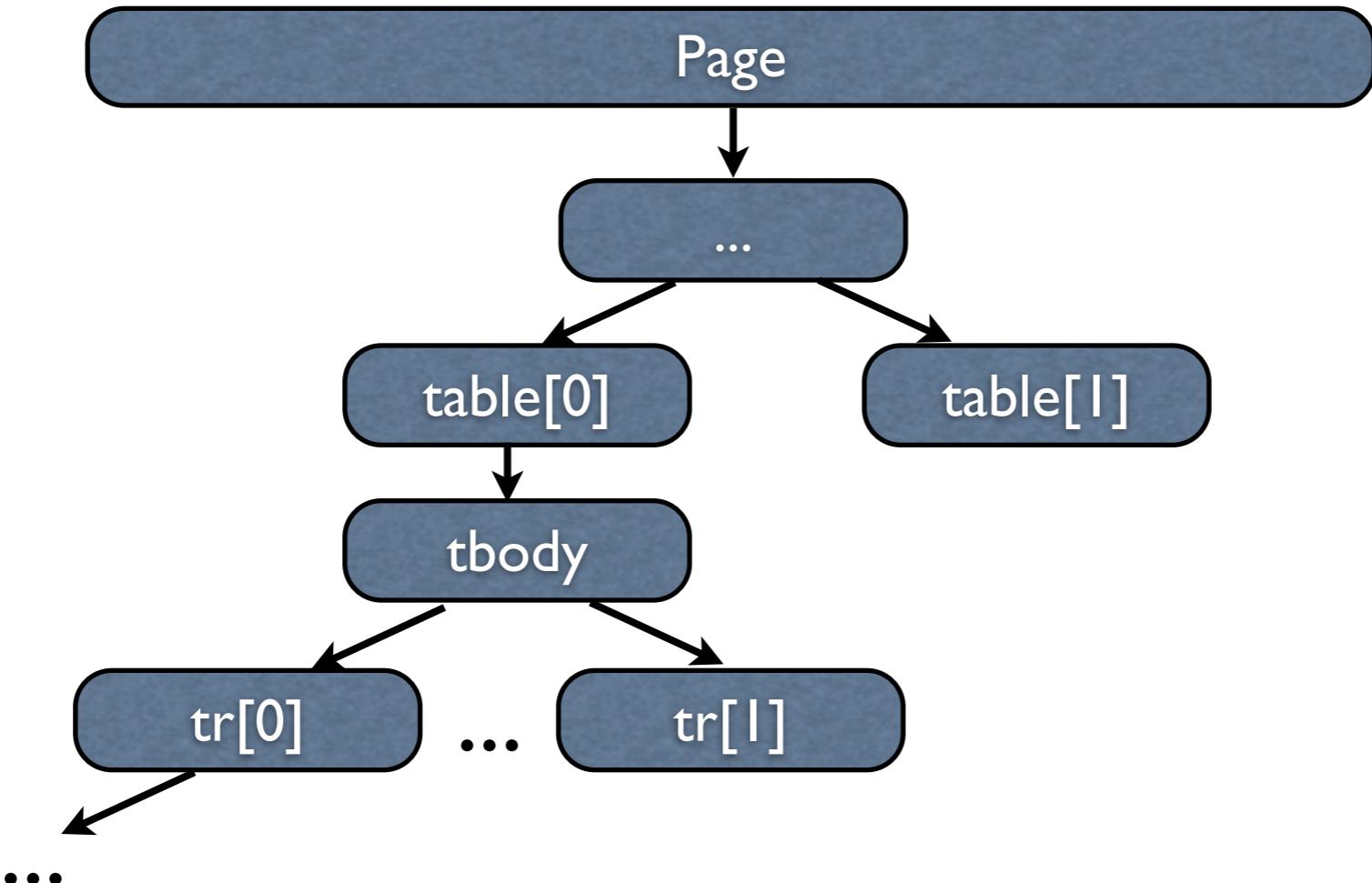


sqlite, scraping, python

putting it all together



Screenshot of a browser developer tools interface showing the DOM structure and a table of movie reviews.

DOM Structure:

```

tMeterScore < tMeterIcon < td < tr.hover < tbody < table.left < div.content_body < div#top...tent_box < div.col < div.body_main < body.body

```

Table Data:

Rank	Tomatometer	Title	No. of Reviews
1.	99%	The Wizard of Oz (1939)	101
2.	100%	Citizen Kane (1941)	66
3.	100%	The Third Man (1949)	60

Right Panel Message:

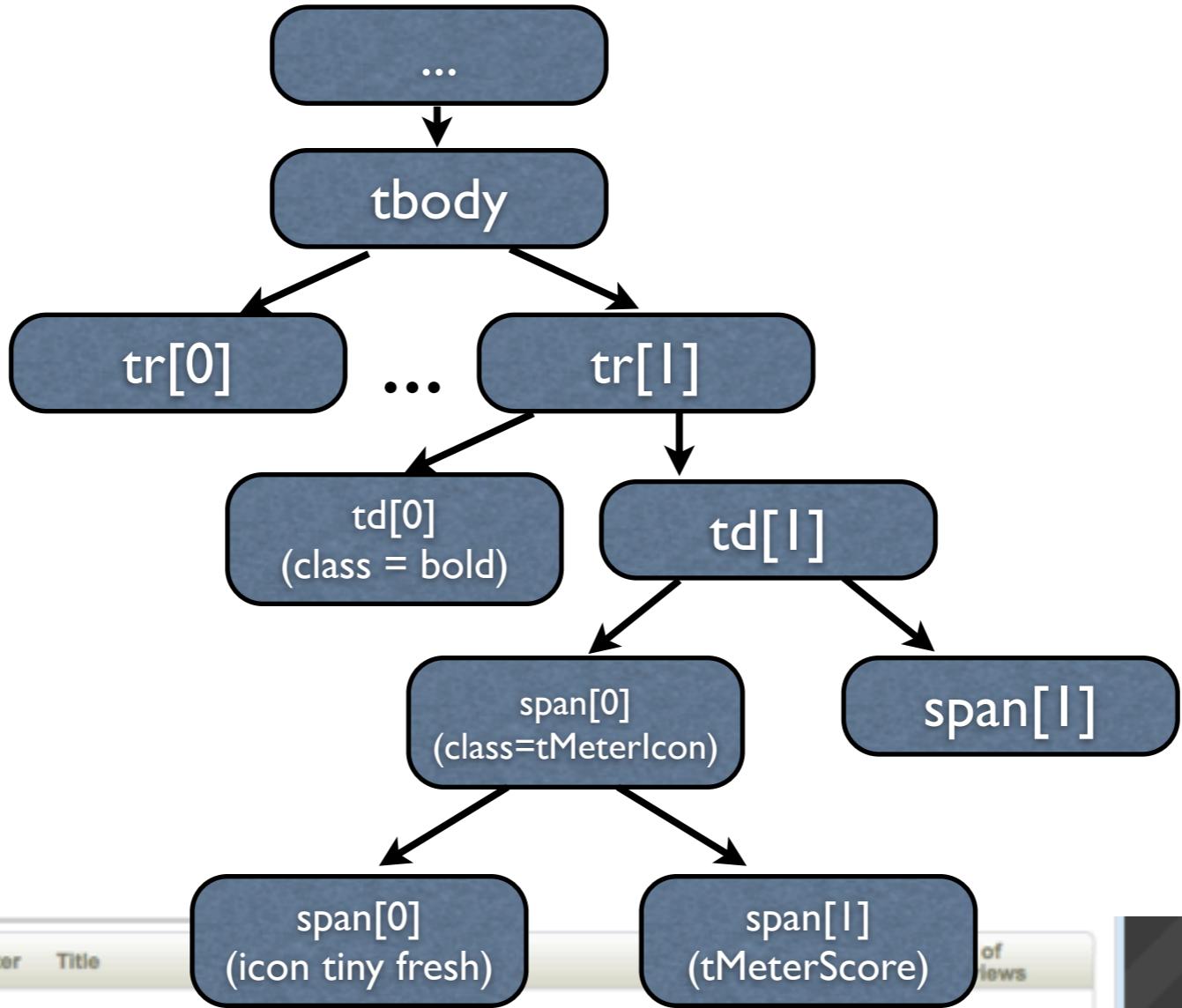
This plugin is vulnerable and should be updated.
[Activate Adobe Flash.](#)
[Check for updates...](#)

Style Panel:

```

.movie_list {
    .tMeterScore,
    .rt_table
}
.tMeterScore {
    display: inline-block;
    text-align: center;
    width: 31px;
}

```



Rank	Tomatometer	Title	of
1.	99%	The Wizard of Oz (1939)	101
2.	100%	Citizen Kane (1941)	66
3.	100%	The Third Man (1949)	60



HTML

```

<div id="top_movies_main" class="content_box">
  <div class="content_header clearfix">
    <div class="content_body clearfix">
      <div class="media_block bottom_divider">
        <form method="get" action="/top/bestofrt/">
          <table class="left rt_table" comp="RowClickable">
            <tbody>
              <tr class="">
                <td class="bold">1.</td>
                <td>
                  <span class="tMeterIcon tiny">
                    <span class="icon tiny fresh" title="Fresh"></span>
                    <span class="tMeterScore">99%</span>
                  </span>
                </td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>

```

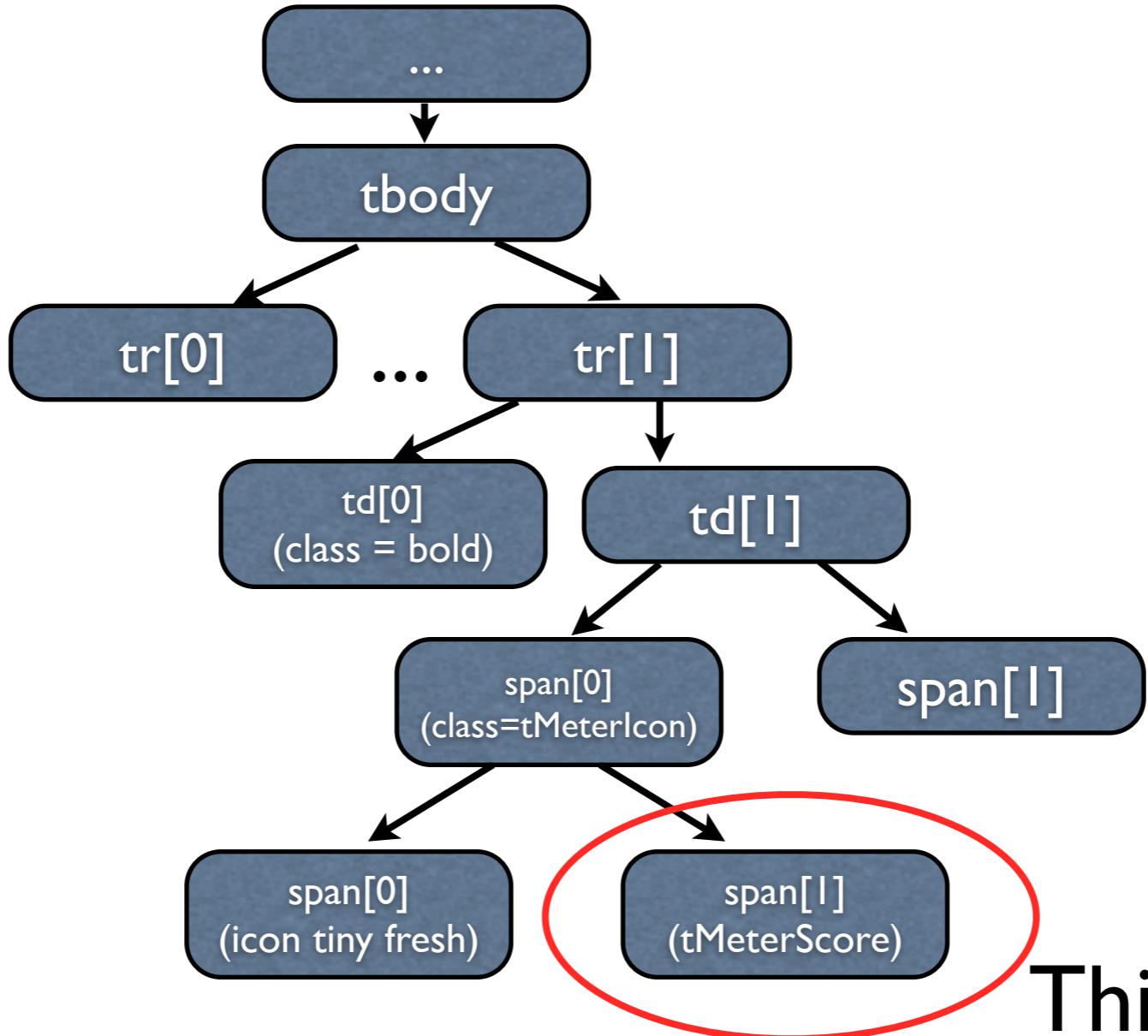
Style

```

.movie_list {
  .tMeterScore,
  .rt_table
  .tMeterScore {
    display: inline-block;
    text-align: center;
    width: 31px;
  }
}

.movie_list {
  .tMeterScore,
  .rt_table
  .tMeterScore {
    display: inline-block;
    text-align: center;
    width: 31px;
  }
}

```



This is where our value lives!

Screenshot of a browser developer tools DOM inspector:

Elements:

- <div class="top_movies_main" style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; margin-bottom: 10px;>
- <div class="content_header clearfix">
- <div class="content_body clearfix">
 <div class="media_block bottom_divider">
 <form method="get" action="/top/bestofrt/">
 <table class="left rt_table" comp="RowClickable">
 <tbody>
 <tr class="">
 <tr class="">
 <td class="bold">1.</td>
 <td>

 99%

 </td>
 </tr>
 </tbody>
 </table>
 </div>
 </div>
</div>

Style:

```

.movie_list {
  .tMeterScore,
  .rt_table
}

.tMeterScore {
  display: inline-block;
  width: 31px;
}

.icon {
  width: 1em;
  height: 1em;
  vertical-align: middle;
}

.tiny {
  font-size: small;
}

```

Now that you've seen the tree, let's see how the code actually traverses this tree!

The screenshot shows a browser developer tools interface with the "HTML" tab selected. The left pane displays the DOM tree for a page, and the right pane shows the corresponding Python code for traversing the tree.

DOM Tree:

```
span.tMeterScore < span.tMeterIcon < td < tr.hover < tbody <
  <tr class="top_movies_main" style="background-color: #f0f0f0;">
    > <div class="content_header clearfix">
    > <div class="content_body clearfix">
      > <div class="media_block bottom_divider">
      > <form method="get" action="/top/bestofrt/">
      > <table class="left rt_table" comp="RowClickable">
        > <tbody>
          > <tr class="">
          > <tr class="">
            > <td class="bold">1.</td>
            > <td>
              > <span class="tMeterIcon tiny">
                > <span class="icon tiny fresh" title="Fresh">
                > <span class="tMeterScore">99%</span>
              </span>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </tr>
</tbody>
```

Python Code:

```
22 def Get_tomato_meter_rating(soup):
23     temp_rating_list = []
24     # look at all tables on the page
25     for table in soup.find_all('table'):
26         # from the list of all tables, find the 1 table where 'class' == 'left rt_table'
27         if table.get('class')[0] == "left" and table.get('class')[1] == "rt_table":
28             # find all table cells within this one table
29             for td in table.find_all('td'):
30                 # find all links within table cells
31                 for link in td.find_all('span'):
32                     # find the one link inside of the span we want
33                     if link.get('class') and link.get('class')[0] == "tMeterScore":
34                         # get the value as a string
35                         rating = link.text
36                         try:
37                             # remove the last character.
38                             # the last char is % sign.
39                             rating = rating[:-1]
40                         except:
41                             print "I couldn't remove the percent sign!"
42
43                         temp_rating_list.append(rating)
44
45 return temp_rating_list
```

The code highlights the logic for finding the "tMeterScore" class within the "td" elements of the "left rt_table" and extracting the rating value.