chem-bla-ics

Programming in the Life Sciences #11: HTML

Egon Willighagen

Published October 30, 2013

Citation
Willighagen, E. (2013). Programming in the Life Sciences #11: HTML. In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/tqs3s-x7289

Keywords
Pra3006, Html

Abstract

HTML (HyperText Markup Language), the language of the web, is no longer the only language of the web. But it still is the primary language in which source code of webpages is shared. Originally, HTML pages were always static: the only HTML source of a web page was that was downloaded from a website. Nowadays, much HTML the is visualized in your web browser, is generated on the fly with JavaScript.

Copyright

Copyright © Egon Willighagen 2013. Distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

HTML (HyperText Markup Language), the language of the web, is no longer the only language of the web. But it still is the primary language in which source code of webpages is shared. Originally, HTML pages were always static: the only HTML source of a web page was that was downloaded from a website. Nowadays, much HTML the is visualized in your web browser, is generated on the fly with JavaScript. In fact, that is exactly what you will learn to do in this course.

HTML has many dialects, and HTML5 is the upcoming next version. The features have become so extensive that we will not have capture half of them; instead, we will stick to the bare minimum needed. But even at an minimum, writing a web page with HTML code is basically writing source code. The compiled version is the view of the webpage your web browser shows you. One important difference is that HTML is much more like a data model representation than it is like computational instructions. That is, rather than saying things like `put("String", xCoord, yCoord)`, we define what is to be shown in in what order with general instructions. Well, in pure HTML that is. Cascading Style Sheets (CSS) is quite outside the scope of this course.

A minimal HTML page looks like:

```
<html>
  <head>
  </head>
  <body>
  Hello world!
  </body>
</html>
```

When we think about this structure, we notice that it is not unlike the key-value maps we covered earlier. For example, compare it to this JSON:

```
{
  "html": {
    "head":{},
    "body":{value:"Hello world!"}
  }
}
```

Even if we introduce HTML attributes:

```
<html>
  <head>
  </head>
  <body>
  <h1><a name="hello">Hello world!</a></h1>
  </body>
</html>
```

The JSON equivalent would be:

```
{
  "html": {
    "head":{},
    "body":{
      "h1":{
        "a":{
          attributes:{"name":"hello"},
          value:"Hello world!"
        }
      }
    }
  }
}
```

So, while these are quite different languages than programming languages, we can clearly see they have been made up by the same (computer science) people. But in my opinion, this is an advantage: because we only need to learn the underlying patterns and can then much more easily switch between different language.

Now, returning to the HTML example, we introduce a bit of terminology. Let's start with the last example:

```
<h1><a name="hello">Hello world!</a></h1>
```

This HTML code example shows the `<h1>` **element** which has one **child element** `<a>`. This child element has an **attribute** `@name`. Elements can contain string content, such as the `<a>` element has, and one or more child elements (and any combination of that). Attributes can only have string content. The HTML specification defines in detail which elements can be child elements of other elements. For example, the `<head>` element can only be a child element of `<html>`. Similarly, each HTML element can only have specific attributes, though some attributes can be attached to any element.

There is plenty of documentation on the web, but there are also tools that can help us write HTML. For example, the http://validator.w3.org/. This website detects errors in your HTML code, and is quite helpful if you are new to editing HTML, as well as useful if you have a lot of HTML experience.

HTML elements you may find useful include the following:

- `<h1>`, `<h2>`, ..., `<h5>`: these are header and can be used to make sections
- `<p>`: indicates a paragraph
- `<div id="someID">`: indicates a section of text. The content of any element with an id attribute can be replaced by any appropriate HTML content with JavaScript

- `<a href="http://...">some link</a>`: this is used to make hyperlinks, href means hyperlink reference
- `<a name="mark1">some text</a>`: this is used to create bookmarks. with `<a href="#mark1">jump to section Mark 1</a>`
- `<script>`: used to include JavaScript code in your HTML page
- `<head>`: this HTML blob contains metadata, a list of libraries to be loaded, but also JavaScript which is executed before the HTML `<body>` is processed
- `<body>`: this contains the HTML that is depicted in your browser window

Keep the HTML simple; the programming is more important.

Exercise: below is part of the HTML/JavaScript source code behind this app. Please indicate which lines are HTML source code, and what is JavaScript.

```
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<!--

Copyright (c) 2013  Egon Willighagen <egon.willighagen@maastrichtuniversity.nl>

 Permission is hereby granted, free of charge, to any person
 obtaining a copy of this software and associated documentation
 files (the "Software"), to deal in the Software without
 restriction, including without limitation the rights to use,
 copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to permit persons to whom the
 Software is furnished to do so, subject to the following
 conditions:

 The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
 HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
 FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 OTHER DEALINGS IN THE SOFTWARE.

-->
<head>
```

```
  <title>OpenPHACTS Jasmine Spec Runner</title>
  <script src="lib/jquery-1.9.1.min.js"></script>
  <script type="text/javascript" src="lib/purl.js"></script>

  <!-- include source files here... -->
  <script type="text/javascript" src="src/OPS.js"></script>
  <script type="text/javascript" src="src/ConceptWikiSearch.js"></script>

  <!-- setup -->
  <script type="text/javascript">
  // get the app_key and app_id from the webpage call -->
var prmstr = window.location.search.substr(1);
var prmarr = prmstr.split ("&");
var params = {};
for ( var i = 0; i < prmarr.length; i++) {
    var tmparr = prmarr[i].split("=");
    params[tmparr[0]] = tmparr[1];
}
  </script>
</head>

<body>
  <h3>Output</h3>
  <h3>Search Results</h3>
  <p><div id="table"></div></p>
  <h3>Compound Details</h3>
  <p><div id="details"></div></p>
  <h3>JSON reply</h3>
  <p><div id="json">Nothing yet</div></p>
  <script type="text/javascript">
var searcher = new Openphacts.ConceptWikiSearch(
  "https://beta.openphacts.org",
  params["app_id"], params["app_key"]
);
var callback = function(success, status, response){
  document.getElementById("json").innerHTML = JSON.stringify(response);
  html = "<table>";
  for (var i=0; i<response.length; i++) {
    html += "<tr>";
    html += "<td>";
    html += "Name: <span>" +
      response[i].prefLabel +
      "</span>";
    html += "</td>";
    html += "</tr>";
  }
  html += "</table>";
  document.getElementById("table").innerHTML = html;
};
searcher.byTag(
  'Aspirin', '5', '4',
  '07a84994-e464-4bbf-812a-a4b96fa3d197',
  callback
);
  </script>
</body>
</html>
```