chem-bla-ics

Programming in the Life Sciences #6: functions

Egon Willighagen

Published October 12, 2013

Citation

Willighagen, E. (2013, October 12). Programming in the Life Sciences #6: functions. *Chem-bla-ics*. https://doi.org/10.59350/hrzhb-m7g26

Keywords

Pra3006, Javascript, Html

One key feature of programming languages is the following: first, there is linearity. This is an important point that is not always clear to students who just start to program. In fact, ask yourself what the algorithm is for counting the chairs in the room where you are now sitting. Could a computer do that in the same way? How should your algorithm change? A key point is, is that the program is run step by step, in a linear way.

However, we very easily jump to functions. In fact, we use so many libraries nowadays, this linearity is not so clear anymore. Things just happen with magic library calls. But at the same time, the library calls make our life a lot easier: by using functions, we group functionality in easy to read and easier to understand blobs.

OK, the previous example showed that we could use the HTML `@onClick` attribute to provide further detail. But I did not show how. This is how:

```
html += "Name: <span onClick=\"showDetails('" +
  escape(dataJSON) + "\')\">" +
  response[i].prefLabel + "</span>";
```

This code adds the `@onClick` attribute and a function call to the `showDetails()` method which takes one parameter, where we pass escaped JSON. That is non-trivial, I understand, and may be due to my limited knowledge of JavaScript. The escaping of the JSON is needed to make quotes match in the generated HTML. In the function later, we can unescape it and get the original JSON again. Importantly, the dataJSON data contains all the details I like to show.

Now, this functions needs to be defined. Yes, plural, because two functions are used in this code snippet: `showDetails()` and `escape()`. The last is defined by one of the used libraries. The `showDetails()` function, however, I made up. So, I had to define it elsewhere in the HTML document, and it looks like:

```
var showDetails = function(dataJSON){
  data = JSON.parse(unescape(dataJSON));
  document.getElementById("details").innerHTML =
    data._about;
};
```

This example actually gives the exact same output as the code in the previous post, but with one major difference. We now can extend the function as much as we like, but the code to output the list of found compounds does not have to get more complex than it already is.