

# LSIS Practitioner Enquiry: Using e-learning to describe the software development process

Charles Boisvert: City College Norwich

## Background

Explaining the production of software, web pages, or even documents is an ongoing problem for teachers of I.T. It is a dynamic process, which is carried out incrementally; yet to describe it, we are frequently confined to static media such as books and handouts. The problem is particularly felt in programming, as it is at all times a difficult subject, that has to be addressed step by step. All too often, I prepare a lesson to find that even simple examples need a succession of handouts to present: a straightforward initial solution, followed by a series of improvements, until the work is explained but the students take home a barrage of materials that confuse them once they are away from live delivery.

## Development of teaching and learning resources

To try and improve on that situation, I developed a web site to present the kind of progressing material that is so common in the programming process. An alternative was to annotate material for the students to show the evolution of a solution; but, as figure 1 (below), makes clear, this is just as confusing, if not more, than offering successive print versions of the code.

```
<HTML>

<head> ... </head>

<body>

  <a href="http:// whatever"> link </a>
  <br>
  
</body>
</HTML>
```

onClick = "document.pic.src  
= 'other\_logo.gif';"

name = pic

Figure 1: annotating as a way to show the coding process

The solution I eventually adopted was to present the code in an online interface. Each example is presented as if the program code was being edited in front of the user, as shown by fig. 2.

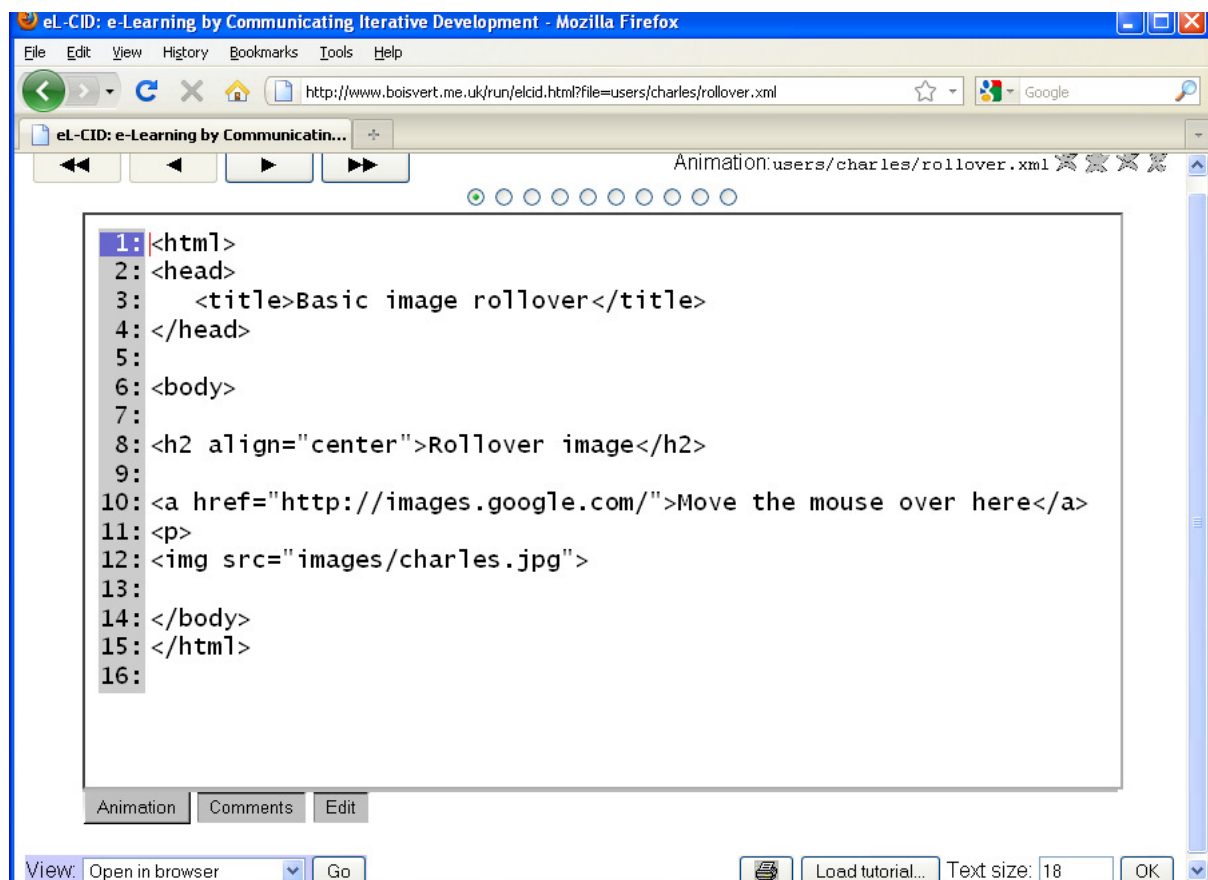


Figure 2: a tutorial

Arrow commands let users navigate, giving them control over the presentation; students may also execute the code and view the result at any intermediary stage, print it, as well as print tutor comments, and finally edit the code or copy it to develop it according to their own ideas.

In this way the tutorials offer a flexible alternative to print, pre-recorded video or live delivery. The advantage for the students is that materials are available to them in and out of the classroom, as it is with notes, textbooks and print; at the same time, they offer flexibility of views and expose the process of developing programs in a way that static media cannot.

## Website and Tutorials

The tutorials already written are gathered in a Web site (figure 4). In all there are over 40 examples, mainly on the subject of Web design, Web programming in ASP and in PHP, and database (SQL) skills.

Each tutorial is tagged with keywords, forming a flexible classification of the different topics covered. The home page shows each tutorial in a clearly visible section, with a link, the keywords and some summary information, as shown by figure 3. This makes the tutorials easy to scan.

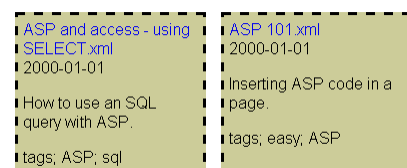


Figure 3: tutorial summary presentation

The keywords appear on the web site as a tag cloud. Larger links correspond to more tutorials available, while the links are also ordered by the number of visits. The two criteria help identify quickly more important keywords to search for. Finally, a breadcrumbs trail acts as a navigation tool.

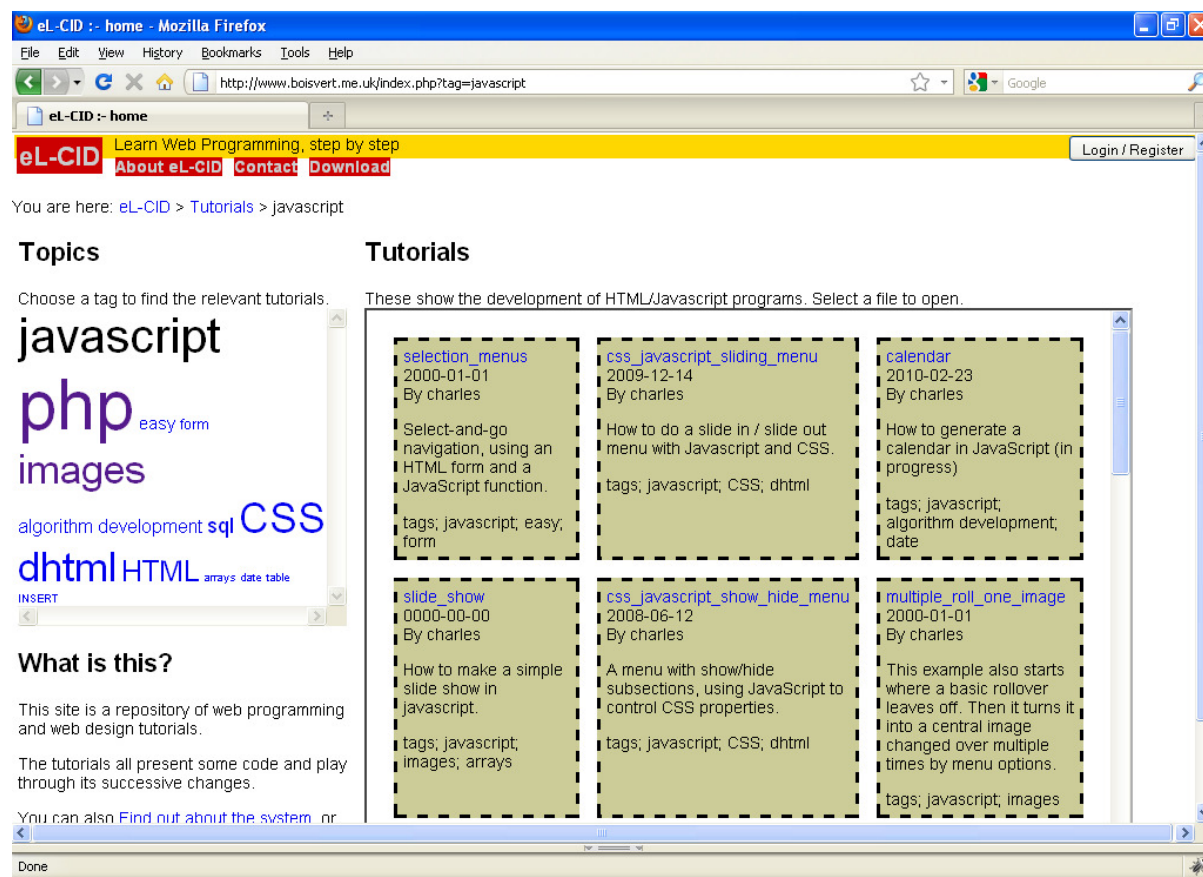


Figure 4: web site (<http://www.boisvert.me.uk>)

## Impact

I collected feedback from actual teaching use by monitoring usage statistics of the web site, through direct observation, and in an end-of-module questionnaire.

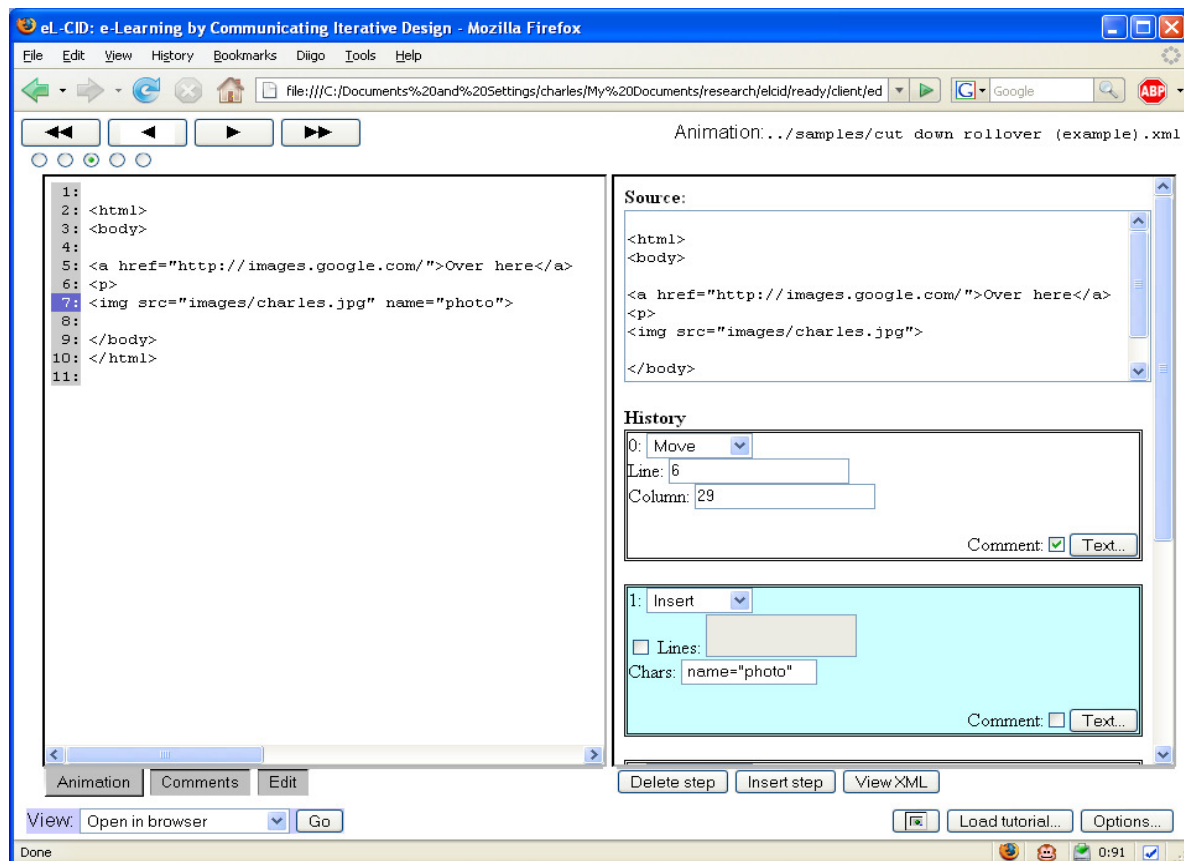
The site data showed a wide diversity of usage of the tutorials. This is usually the case with tools that support discovery learning, an indication that the tools were open to this form of learning. I expected complexity of the interface to be a problem, but in practice in the classroom the students navigated it without difficulty.

The questionnaire responses confirmed that the students found the interface easy to navigate, and that the tutorials available were useful; but the students found they lacked more advanced tutorials, as well as more tutorials covering a wider range of technologies. This is not surprising given the site usage. This also coincided with my personal experience of the technique: I found that composing tutorials was a painstaking process and that it was difficult to keep up with student demand. As a result, I created editing tools to speed up the development of the site.

Practitioner research has helped stay up to date in my field; sharpened my attention to student feedback; and forced me to consider the difficulties of each exercise that I made into a tutorial.

## Teacher tools

The editor helped building new programming animations; at the time I write, 43 tutorials are available, and more are being created.



**Figure 5: editing a tutorial**

The online editor lets the teacher set an initial version of “source” code together with its modifications - operations like ‘move’ the cursor or ‘insert’ text. Writing a tutorial in this way turns out to be a reflexive process: as I create tutorials, I perceive the difficulties that students face when they go through a similar development process, more than if I simply wrote a single example program.

## Find out more

The website and the existing tutorials are available to all (<http://www.boisvert.me.uk>). They offer a diverse set of examples and shed light on aspects of web programming as part of a blended learning method.

## Contact

This study was carried out by Charles Boisvert at City College, Norwich. If you have any questions or comments, please e-mail Charles at: [c.boisvert@shu.ac.uk]