

Re: UI toolkits for Python

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-10/msg02290.html>

- *From:* Steve Holden <steve@xxxxxxxxxxxxxx>
 - *Date:* Wed, 19 Oct 2005 09:46:37 +0100
-

Claudio Grondi wrote:

"Steve Holden" <steve@xxxxxxxxxxxxxx> schrieb im Newsbeitrag
news:mailman.2143.1129532422.509.python-list@xxxxxxxxxxxxxx

Claudio Grondi wrote:

[...]

Do I miss here something?

While you are correct in saying (I paraphrase) that HTML interfaces nowadays can offer a rich graphical interface, it can be quite difficult to manage state maintenance between the two components (web server, web client) in the system.

The cause of confusion here is, that HTML interfaces don't necessary

need a

web server and HTTP to work. I mean, that Internet Browsers have an API which allow access to them directly, so they can be used

without

a server as a kind of GUI library supporting widgets programmed in HTML and JavaScript (I haven't used them yet in this form, but I

Re: UI toolkits for Python

think

it should be no problem - right or not?).

You are perfectly correct that interfaces can be programmed in the browser. As has already been said, with the adoption of standards like CSS2, it's even possible to make such stuff run across multiple browser platforms (at the moment, though, just try writing an interface that makes table rows appear and disappear in a cross-browser fashion: you'll find the stylesheet techniques you need to use for Mozilla and IE are very different). But this doesn't help you at all if you are trying to interface to Python logic.

A "proper" GUI runs all functionality inside a single process, and allows much easier control over complex interactions, creation of dynamic dialogues, and so on.

Complex and dynamic is in my eyes possible with HTML and JavaScript, so

I

still don't see where is a problem with this approach. I have programmed already a HTML and JavaScript driven server platform and know about (the solvable) problems with the back-button and sessions (it is sure not the easiest way of programming a GUI). The only disadvantage of not using plugins or Java is, that real time interactions are not possible, but this is in my eyes usually not the requirement for a standard kind of GUI with no gaming, no Virtual

Reality

and no timing of user response down to milliseconds (but consider, that

with

Re: UI toolkits for Python

a good speed of connection it is even possible to play blitz chess over

the

Internet).

So, back to the subject: with an HTML/Javascript interface, how do you propose to bolt Python logic into the same process? Or do you admit that the application that interacts with such an interface either has to be all JavaScript or in a separate process?

I haven't used it in such configuration yet, but I mean, that it is not a bad idea to use local DHTML files (i.e. HTML with JavaScript) combined with a Python driven HTTP server like e.g. Karrigell which makes it possible to access local files by executing Python scripts triggered by demand raised by submitting to it DHTML form data resulting from user input. So mixing JavaScript in local DHTML files for performing what can be done with JavaScript inside HTML and a HTTP server capable of executing Python scripts will do the job which JavaScript alone can't because of lack of access to the local file system and another parts of the computer system on which such kind of UI is executed.

Indeed. But this again partitions the problem into client functionality and server functionality residing in two separate processes, albeit on the same computer, with the previously-mentioned state maintenance problems. This issue isn't new, you know.

Another possible scenario I have in mind is to control the Internet browser directly from a Python script using DHTML as a language for definition of appearance and specification of necessary data processing of in the browser displayed UI. This way the Internet browser and HTML with JavaScript can be considered an UI toolkit for use in Python.
Hope with this above to have got Python back on topic.

That is, I suppose, a possibility, but you certainly can't claim it would be platform-independent. Unless you know of some common control interface respected by both Internet Explorer and Firefox ;-)

regards
Steve

--

Steve Holden +44 150 684 7255 +1 800 494 3119
Holden Web LLC www.holdenweb.com

Re: UI toolkits for Python

PyCon TX 2006

www.python.org/pycon/