Pyjamas

Support this project

Python-based Web Application Development Framework

Source Code

The pyjamas project is using gitolite to manage the git repositories. For read-only access, use the fo

```
git clone git://pyj.be/git/pyjamas.git
```

If you would like write-access, please simply ask on the mailing list. For developers, use the followin

```
git clone gitolite@pyj.be:pyjamas
```

There is also a public git browser (a pyjamas app!) at:

http://pyj.be/pygit

Pyjamas Developer Guidelines

In the Pyjamas repository is a file DEVELOPER.RULES. As long as you follow those rules, you can it is several separate projects. For example: modifying the javascript compiler has absolutely nothing requires testing on **eight** platforms (five web and three desktop) including compiling and testing usin nearly **ten** web browsers (Firefox 2 to 4; Opera 9 to 10.5; Safari 3 and 4; Google Chrome; IE 6 and a make the best efforts and use your judgement, and ask for help on the mailing list.

Lastly - please use the bugtracker to report bugs (regardless of how trivial or small); the mailing list t on.

How to set up a Web development environment

Web application development can be tricky: it can come as a bit of a shock when compared to pytho proper debugging assistance whatsoever, by default. You **will** need to install and/or enable a debug

- For Firefox, install both Venkman and Firebug.
- For IE, install the Microsoft Script Debugger.
- For Safari, go to the settings and enable "Development".
- For Chrome, Web Developer has been reported to make your life easier.
- Opera users, you are extremely lucky: Opera has stack tracing by default.

You should also note that the Pyjamas compiler has a "-d" option which will enable a python-like stac

Lastly, it is worth reiterating that Pyjamas Desktop runs as pure python. You should give serious consideration to developing it for the browser. The availability of python runtime stack traces and the simple fact that you have a lot of errors than (brain-damaged) browsers has generally found to make life much much easier.

Building User Interfaces with Pyjamas

To become familiar with the user interface side of Pyjamas, you might like to refer to the examples or

You might find the ui module class hierarchy useful. The `ui` module contains all of the main classes y quite confusing because of the number of classes defined. However, there is API documentation, alc

You might also have a look at the GWT Documentation for widgets that have ported to pyjamas.

Sources Overview

The pyjs repo contains both shared libraries (usable in python or javascript mode), and "runners" tha is a quick what-is-what.

/addons/: Contributed libraries, added to the pythonpath when translating code to js
/bin/: Executables created when bootstrapping appear here
/builder/: Just ignore that for now
/contrib/: Miscellaneous helper scripts
/dev/: Just ignore that for now
/doc/: The content of pyj.be is here
/examples/: Lots of examples with their build scripts (also used to test all is ok)
/examples/libtest/: Used for unit-testing, build it and launch it to have in-browser tests performed
/library/: All common widgets and utilities, with platform overides when necessary
/library/gwt/: libs tracking original gwt sources, without improvements
/library/pyjamas/: libs mirroring and cross-linking gwt/ ones, to add pyjamas-specific features
/pgen/: Python parsing suite recoded in python
/pygtkweb/: Just ignore that for now
/pyjd/: Desktop runner (executes non-translated code, on several possible backends)
/pyjs/: Actual python-to-js tools: translator, browser linker, python builtins and stdlib recoded for java
/pysm/: Spidermonkey runner (executes js code on that js engine)
/pyv8/: Google V8 runner (executes js code on that js engine)
/tests/: Just ignore that for now
/bootstrap: The script through which everything starts
/test.py: Very useful to easily launch unit-tests (especially libtest) on several engines

Key points to remember:

- the "/pyjs" part is only used in translated mode, other libraries are used both for translated (bro
- each widget is split between "/library/pyjamas/ui/" and "/library/gwt/ui/" trees, to differenciate le
- for testing, "/test.py" and compiled "/examples/libtest" are your best allies