



# kupu

- I. Client-side applications
- II. The Kupu project
- III. Demo

- There's an increasing need for client-side apps
- New browser generations allow a great deal already out-of-the-box.
- Choice of the platform
  - Java: too complicated, loads slowly: “write once, test everywhere”
  - VBScript, ActiveX, XUL: too platform-specific
  - so JavaScript it is!

- Common prejudices
  - “Who needs it?”
  - “That's for kids”
  - “JavaScript code looks horrible”
  - “I can't do with it what I need to do”
- Real advantages
  - Object orientation through prototypes
  - Dynamic typing
  - Anonymous functions

- Disadvantages
  - umm, it's not Python :)
  - bad design patterns are tempting
  - one big namespace
  - DOM and other XML standards often badly implemented or not compatible between browsers
  - debugging is a **major** pain
- Our motto:
  - “If you have to do it, do it right!”

- object orientation all the way
- MVC-like components:
  - Model: the HTML document
  - Controller: Kupu Editor, many tools
  - Views: toolboxes/drawers/popups
- Sarissa library solves many DOM problems
- Debugging: avoid error-prone code in the first place: unit tests!

- A WYSIWYG editor for your browser
- cross-browser (IE, Mozilla)
- cross-platform (Plone, Silva, Apache Lenya, Interchange, Midgard and more)
- OpenSource (BSD-style license)
- Official OSCOM project (<http://kupu.oscom.org>)
- “kupu” is Maori and means “word, statement, remark”.

kupu's philosophy:

- not just a replacement for `textarea`
- document-centric editing
- it's an application, not a form (if you want)
- user-friendly (application-like, no popups, etc.)
- extensibility and pluggability thanks to component architecture
- (quite) easily integrated into any type of CMS



- produces well-formed XHTML
- asynchronous saving w/o page reload
- context menu and/or “file dialogs” (drawers) for image, link, and table insertion
- based on standards
  - XML configuration
  - XML-based communication with the server
  - XML/XHTML/XSLT-based templating system
  - CSS layout
  - HTTP/WebDAV (e.g. for asynchronous PUT)

- integrated into Zope 2, Silva 1.0, Plone 2.x
- integrated into Apache Lenya, Interchange, Midgard, etc. by others
- kupu 1.1 soon to be expected
- short-term plans:
  - ATAG (editor accessibility), Zope3 integration, bootstrapping through XML
- you can help us with:
  - IE compatability, the build system, more integrations, etc.

- world domination ;-)
- better component architecture (interfaces?)
- better abstraction layers for Selection, Range
- get rid of execCommand (a lot of work!)
- improve event handling (event service)
- more independent and reusable components (tools)
- more reusable frameworks, in the spirit of Sarissa and ECMAUnit

- Out-of-the-box
- Silva
  - best example for “document-centric editing”
  - Toolboxes
- Plone
  - kupu as a form widget
  - Library Drawers
- Library Drawers in BXE
- Apache-Lenya

- Questions?