

Why does Jmbo exist?

- Sites (especially mobi) have recurring elements
- We want a standard set of templates for all devices
- Turnaround time
- Make it easy for novice developers to be productive
- Bring designers closer to the project
- The usual project process is full of pain

The usual project process

- Creative / strat has a beautiful vision
- Write a scope of work promising a lot
- Wireframe
- Designer does HTML and CSS
- Developer must integrate
- QA etc etc
- Deploy and support

Things always go wrong

- Unrealistic promises → new unplanned features and code
- Horrible HTML and CSS
- Upside: most developers are CSS gurus
- Developer must co-ordinate and fight with all involved parties

The Jmbo project process

- The toolkit is known
- Developer is involved from the start to identify problem areas
- UX can create wireframes which focus on layout, not detail
- Designer works with developer when development starts
- Designer does not write HTML, only CSS
- Designer can continue working in parallel
- Rapid QA releases since there are no bottlenecks

No HTML? Really?

- It depends
- Identify what Jmbo offers you, then add custom bits as required
- Write custom HTML in same style as Jmbo, then you get most styling for free

The stack

- Python
- Django
- The usual web elements (JS, JQuery, CSS etc.)
- Postgres with PostGIS preferred, MySQL will work, SQLite with Spatialite for demos and the insane
- buildout
- supervisord
- nginx
- gunicorn
- User agent detection and switcher
- HAProxy

Links

<http://pypi.python.org/pypi/jmbo>

Core CMS product.

<http://pypi.python.org/pypi/jmbo-foundry>

A.k.a. Jmbo Go. Present jmbo-* apps as a logical whole. Build pages.

<http://pypi.python.org/pypi/jmbo-skeleton>

Replacement for paste. Get a demo or new project up and running quickly.