



python.org
Redesign Proposal

T: +44 0113 225 2500 M: +44 07980 594768
E: info@pollenation.net W: www.pollenation.net

Introduction

The current Python site has remained in the current look and feel since late August 1998. During PyCon 2003 the need to make changes to python.org was clearly identified (see Appendix A). The overhaul is taking place in (at least) two phases. These phases are not strongly defined but should be considered as :-

- Phase One : working optimisations to the current site
- Phase Two : comprehensive redesign

This document is intended to address the background behind the need for change and a proposal for design and web build of Phase Two. In addressing this, the proposal will touch on areas of information architecture due to it's link with usability and navigational metaphors.

PSF/PWC Requirements

The main identified requirements drawn from the initial PyCon meeting notes and mailing list comments by PSF/PWC board members :-

- make it look more "professional".
- make the site more obvious to use, particularly for beginners.
- prioritise links to download and documentation
- introduce an obvious site search facility
- reduce the clutter in commercial exits
- head towards valid markup
- promoting Python, leverage evangelism
- use existing metrics to focus improvements
- simplify the homepage
- documentation types confusing
- simple search on home page
- Should bear in mind that mirroring is important
- No 3d rotating head of Guido with glowing lights around him (Kevin Altiss)
- All sections should adhere to new design templates (Kevin Altiss)

- a google site search on python.org is very ineffective right now (Kevin Altiss)
- make it work in lynx (Aahz)
- we want blue (Aahz)
- reduce the maintenance hassles for the webmasters (Kevin Altiss)
- top global nav and left hand secondary nav (Aahz)
- front page under 60k (Aahz)
- breadcrumb trail across the site (Aahz)
- Steve Krug – Don't Make Me Think (Guido Van Rossum)
- No flash, No dark backgrounds (Guido Van Rossum)
- Don't design by committee (Guido Van Rossum)
- Browsing then searching then sitemap (Guido Van Rossum)
- home page should have a lot more "sell" with convenient links to the developer resources (Kevin Altiss)
- there should not be two sites. Python.org is *the* homepage (Aahz)

This is not a comprehensive list of requirements but is kept together as a distillation of comments from the wiki and the mailing lists. The following section will restate the core requirements and all further work will follow.

Core Requirements

If we break the requirements stated in the previous sections into their underlying needs we reduce them down to four points :-

- Usability / Clarity
- Professional Design
- Both Marketing / Informational Purposes
- Best Practises in Web Development

Each of these is addressed below

Usability / Clarity

The site must be as usable as possible in both information architecture, design and message. In order to achieve this, it is typical to address the most common use cases.

Appendix B details a large list of possible users and scenarios. From these we have distilled the following major use cases.

- 1) Non-programmer wants non-technical information to assess Python
- 2) Non-Python Programmer wants to assess Python
- 3) Beginner wants to start / progress with Python
- 4) Python programmer wants general best practise information
- 5) Python programmer wants to solve a particular problem
- 6) Python programmer using site as reference
- 7) Python programmer wants to start contributing
- 8) Python programmer browsing for news
- 9) Python Contributor day to day activities
- 10) Non-Programmer wants to know what this Python thing is
- 11) Teacher wants teaching resources / information
- 12) Journalists sourcing material

There may be various sub-cases of these activities and many more use case but these cover the majority needed to address usability. Appendix C expands on these use cases.

Some of the less obvious web practises that we work to are listed below, these are learnt through research and experience, Appendix D lists resources used

- Page should have a time to first click of less than 8 second
- Answer : Where am I? Where do I go Next? How do I get back?
- A preferred, universal vocabulary should be created
- Don't hijack the browser
- Recognise that most people will be deep linking into the site
- Scrolling should not be needed until a user stops to read
- Pages should print clearly
- The site content should benefit accurate search indexing

The first of these is to do with how fast a site is preceived to be, not what size all of it's elements add up to. This actually means the time until you can usefully use the

navigation or content. This affects sites that use graphics as navigational elements. In these site, all of the headers graphics are loading before it gets around to loading the menu item graphics. This typically means the time to first click is similar to the time to download the whole page. If a page uses html text as the menu items, the page will be usable before even the whole html has loaded. In this way the page may be usable before even half of all page elements are downloaded. This increases the 'perceived' speed of a site.)

Appendix A

Kevin Altis would like to see a 10x increase in the Python user-base in the next 2 years.

Promoting Python

- leverage evangelism (what are the tenets of evangelism)
- website is very high visibility - don't have phone number or press contact

I suggest that people interested in learning more about effective evangelism watch and read what Guy Kawasaki has to say on the subject. All of his books are quite readable, pretty short, and contain many amusing anecdotes. See the link below for more info; the signal to noise ratio is very good. [ka]

<http://pythonology.org/pipermail/marketing-python/2003-March/001124.html>

The main point about the website is that since we don't have traditional marketing and PR dollars, nor do we have normal phone and email contacts for people to get immediate answers about Python, the website needs to do that work for us. The site is the face of Python to the rest of the world. [ka]

Who is the customer?

- people who don't use python today
- people who do use python today - information source
- lots of different types of customers
- identify use cases for groups of users

We need to establish metrics

track usage to ensure modifications that are improvements

Adoption rate of python

increase in traffic and new users of python

By establishing some metrics in the web site stats such as total unique IP addresses, page views, and downloads per month, etc. we can see how our "popularity" is growing. Metrics should also tell us whether some of the "improvements" made to the site really did what we thought they would or ended up having a negative impact on usage. That way we aren't just guessing and modifying blindly. [ka]

Not focused on technical solutions

- identify problems
- identify customers
- how are people using website
- what is missing
- what are websites we like (similar customer bases)

At the Open Space, I didn't want to focus on the technical issues of implementing the site, but that conversation is okay to carry on in parallel here on the list. [ka] Kevin is comfortable with the site redesign process

Product

Mockups, templates estimate work to estimate effort

Before implementing any actual changes, we should make templates and mockups to get feedback before making the changes live. [ka]

Current implementation problems

No database backend

I expect Aahz, Steve, Thomas, and other pydotorg maintainers to articulate the current problems they have maintaining the site, time sinks, and ideas for where we might make site maintenance easier and improve their lives. :) [ka]

Make recommendations to current web maintainers and PSF members

Process

establish a working group empowered to make changes to the website the website is owned by PSF That's what this mailing list is about. [ka]

Problems

- ownership of domain

- clarifying process to change website
- get authority delegated from PSF to implement changes

Recommendations

5-6 people balanced between management and implementation

Where is the effort to do the work going to come from

Ongoing management is required to ensure that maintenance on an ongoing basis gets done

Behind

- job postings that don't get uploaded
- 50% of the mail to webmaster doesn't get answered
- webmaster workflow
- multiple responses

Personnel

- Who is available to do work
- How much time do they have
- What are the skill sets

Split off the PR effort from the language development effort

Introducing Python Video - done

Appeal to the people who have the skills that are required

Audience

- People who don't use python today
 - recommendation from another source
 - resource for further information

- Advanced developers
- defectors
- journalists who are trying to get more information
- Managers (tool selection)
- teachers
- python core team PythonLabs
- python contributors (community)
- newbies (programming newbies) non-programmers
- community of exiting python users
- job seekers
- event page
- books
- user group
- trainers

What is python

How do I start doing python

Metrics

identify customers

python in business

- teaching/learning python
- then track traffic to the sub pages to discover customers

wide diversity

- more general launching page
- home page is too complicated

home page

- asking "what were you looking for. Did you find it"
- search - used to determine what they were looking form

Analysis of search term metrics

will be difficult because the access to the log is hard to get.

We still have assets at CNRI seach.cnri

Anyone interested in joining the python.org maintenance team

- Aahz says the team members are restricted to well known community members
- Give the appearance of needing help
- Give the appearance of being open to joining
- Look for volunteers

Specific specialized areas of responsibility: Python in education

Kirby Urner for the edu-sig/Python in Education pages can be our test case of having page or section maintainers not directly involved in the maintenance of the rest of python.org. [ka]

py.org experience

Zope3

ht to html still works well

Wiki

Needs a strong editor

Which maintenances take the most time

Where is the time going now?

- job listings
 - add a form to allow for posting jobs
 - add a secure form to allow for removing jobs

- event listings
 - add a form to allow for posting events
- broken links
 - webmaster mail
- help@python.org ->tutor
- reflector (a static page of suggestions)

Jeff Epler to work with Steve to do the PyCon website

Credit card donations

- Paypal
- ShareIt
- donate.python.org

python.org needs a certificate

40% Documentation is the number one hit

RDF file is also getting a lot of hits 750000 hits

unique IPs in march 180,000 <http://www.python.org/wwwstats>

Similar sites

- php.net
- mysql.com
- postgresql.org

Web site computer resources are adequate for current needs

- 10% load on the machine
- hot backup could be provided

Continuing the discussion

PSF board meetings forming a subcommittee or group of volunteers to carry on the work

downloads of documentation

- faster to have unified download page
- too confusing

"Get Python Now"

- Make the download of documentation very straightforward
- Matrix of choices

python.org SIG design mailing list -> next action reimplementing SIGs

pythonology.org

- a good new user site
- only if the support of the pages will come from the original author

Appendix B

Definition of Target Audience

There are several key groups we will try to reach simultaneously:

- (a) managers
- (b) programmers
- (c) journalists
- (d) educators/students
- (e) specific user domains
- (f) non-programmers

Managers

Managers are personnel charged with making higher level decisions about how software development takes place in their company or organization.

Managers are typically concerned about:

- * Keeping development costs under control
- * Improving success rates for projects (i.e., doing a good job)
- * Keeping maintenance costs down
- * Keeping deployment and support costs down
- * Overall, keeping total cost of ownership reasonable
- * Finding employees to work on development and maintenance
- * Not getting fired over choosing a non-mainstream solution that fails
- * Increasing chances of getting other interesting projects
- * Obtaining power equal to their responsibility (autonomy to make decisions)
- * Gaining respect in the company
- * Some will seek to increase their sphere of influence in the company
(move up the corporate ladder)

Additional attributes:

- * Often have very limited time available for investigating technologies or solving problems (5 or 15 minute blocks)
- * May prefer passive role, wanting to be "sold to"
- * May not have deep technical knowledge
- * Likely to be influenced by professional looking materials
- * Less familiar with internet, web searching, etc.
- * Held very accountable for bad decisions made by self or staff under them

Programmers

Programmers are personnel that actually design, write, and maintain code.

This group is typically concerned about:

- * Delivering under tight deadlines
- * Reducing technical head aches
- * Finding interesting/fun/new/useful/cool solutions and tools
- * Finding off-the-shelf components to use instead of hand-coded modules and integrating those into their application
- * Keeping their resumes full of "hot" and up-to-date technologies
- * Going home on time
- * Increasing chances of getting other interesting projects
- * Obtaining power equal to responsibility (autonomy to make decisions)
- * Gaining respect in the company

Additional attributes:

- * Often skeptical of marketing claims or advertisements
- * Actively and persistently/patiently seeks out information to solve problems, working until the problem is solved
- * Tend to look for technically-oriented and technically rich marketing

materials, or want to go directly to download and documentation

- * Very familiar with internet, web searching, etc.
- * Don't as often require slick looking materials (content counts more)
- * Held somewhat accountable for mistakes, often with substantial leeway before endangering their job

Journalists

Journalists are not directly end-users of Python but they control a large percentage of our information channel to our target markets. Journalists typically are concerned about:

- * Obtaining content that will sell copies of their publication
- * Printing/publishing a good story, something new and interesting that hasn't yet been seen elsewhere
- * Establishing themselves as an expert in some domain (e.g., Open Source)
- * Getting a rich, informative story together very quickly (sometimes w/o much in depth knowledge of their own)
- * Gaining and retaining readership

Additional attributes:

- * Often have limited technical knowledge
- * Very cynical of marketing (but will still remember a first impression)
- * Often will gladly take pre-written stories or fragments, if well written
- * Usually very familiar with internet, web searching, etc.
- * The story comes first, accountability second
- * Do not seek out stories like Python: We need to reach out to them
- * Need a human contact (not an e-mail address or web form)
- * Need experts, sources, and people willing to tell personal stories
- * Need credible-looking supporting materials (particularly the web site)

- * Need a *very* concise backgrounder (our proposition in 100 words or fewer)

Educators and Students

Educators and students are concerned with:

- * Speeding, facilitating, or enriching the learning process
- * Managing complexity in intro courses and in teaching language concepts
- * Allowing domain-specific courses (AI, 3D graphics, data structures, algorithms, etc) to focus on the domain and not the language
- * Learning languages and tools that will help them get a job
- * Often, keeping cost of tools down
- * Finding tutors that know the subject area or language

Additional attributes:

- * Avoid heavily commercial products when possible
- * Never have enough time to build course material
- * Usually familiar with internet, web searching, etc.
- * Tenure and reputation are very important factors; enjoy more autonomy after tenure or when well established

Specific User Domains

See Domains of Usage below for details on each type of development that can benefit from using Python. Some of these likely should be designated as targets that require separate or specialized promotional treatment.

These will be fleshed out later.

Non-programmers

Technical or scientific non-programmers aka "power users" are often in a position where they would like to get certain work done without depending on programmers. Python has been used by such users for tasks such as simulation, 3D graphics/design tasks, and basic scripting. Some of these learn-as-you-go engineers have produced some very successful systems.

This group is concerned about:

- * Lack of a daunting learning curve
- * Ability to focus on the problem rather than the code
- * Existence of some sort of materials specific to their domain that help them bootstrap into using Python
- * Good libraries and support tools for their domain
- * A supportive user community
- * Finding alternatives to older tools such as FORTRAN

Appendix C

Appendix D

- "Building Accessible Website" - Clark
- "Don't Make Me Think" - Krug
- "Homepage Usability" - Nielsen & Tahir
- "Designing Web Usability" - Nielsen
- "Practical Information Architecture" - Reis
- "Information Architecture" - Wadtke
- "Top 100 Internet Mistakes" - Peter Burns
- "CSS - Separating Content From Presentation" - Briggs, Costello
- "Collaborative Web Development" - Brudman
- "WWW Layout" - Glenwright
- "WWW Type" - Pring
- "Reality Chec"k - Wieners & Prescovitz
- "Hotwired Style" - Veen
- "Secrets of Successful Websites" - Siegel
- "Click Here" - Pirouz
- "Accessible Websites" - Thatcher, Bohman, et al
- "Web Metrics" - Sterne
- "Submit Now" - Chak
- "Designing with Web Standards" - Zeldman
- "Designing CSS Pages" - Schmidt
- "Usability" - Braun, Gadney, et al
- "Usability for the Web" - Brinck, Gergle, Wood
- "Eric Meyer on CSS" - Meyer
- "Customer Centred Design" - Hyatt