

Apprendre à utiliser *Sphinx* en 1 heure et quelques slides

Séminaire Cr@ns 2013/2014

Lilian Besson

ENS Cachan et CRANS

26 décembre 2013

Lilian.BESSON@crans.org

Ces slides et les ressources utilisées pour les exemples sont disponibles en lignes
http://perso.crans.org/besson/publis/seminaire_crans/

Plan

- 1 Présentation
 - Plan
 - De quoi va-t-on parler ?
- 2 Conclusion et perspectives
 - Résumé

Surtout, n'hésitez pas à m'interrompre !

Documentation de code

Quel langage ciblé ?

↪ Python

(versions 2 ou 3)

Le plus utilisé pour les scripts du Crans.

Pourquoi documenter ?

(cf séminaire du 19 Février 2013)

propreté documenter nécessite un code **propre**,

partager plus facile de re-travailler sur un code documenté,

renseigner plus de transparence, montre un soucis de clarté.

Comment

Avec Sphinx, le générateur de documentation officiel pour le langage Python.

Référence : <http://sphinx-doc.org>.

Déjà déployé

Sphinx est déjà utilisé depuis Mars 2013 par le Crans, voir :

<http://doc.crans.org/>.

Un exemple concret :

Conclusion

Nous avons présenté

`sphinx-quickstart` pour facilement initialiser un projet Sphinx,
`reST` un langage à balise *léger* pour mettre en page les commentaires Python (*docstring*) et rédiger la documentation,

`sphinx-build` pour construire la documentation (à la main),
`Makefile` un Makefile générique utile pour facilement construire la doc,

`Jinja` une syntaxe simple pour changer l'apparence des pages web produites avec Sphinx.

Merci d'avoir écouté !

Des questions ?

Pour en savoir plus ?

- ↪ parcourir mon site (<http://perso.crans.org/besson>), fait avec Sphinx, pour des idées de personnalisation (intégrer Google Analytics, des raccourcis clavier, Google Translate et plein d'autres trucs !)
- ↪ regarder la bibliographie.

Bibliographie



Richard Fikes and Nils Nilsson.

[FN72] *Strips : A new approach to the application of theorem proving to problem solving.*

Artificial intelligence, 2(3) :189–208, 1972.



Yangmei Liu.

[Liu02] *A Hoare-style proof system for robot programs.*

AAAI, 2002.



Herbert Malte.

[Mal11] *pyperplan, a lightweight STRIPS planner, written in python 3.*

<https://bitbucket.org/malte/pyperplan>, 2011.