ЛАБОРАТОРНА РОБОТА No5

**Система автоматичного створення довідника користувача та оформлення коду за допомогою Coding Convention.**

**Coding Convention**

Если спросить программиста на python что самое главное в языке pyhton, то они ответят что это легкость понимание написанного кода. И это является одной из ключевых фишек python. И исходя из того факта что код гораздо чаще читается чем пишется, то давайте рассмотрим что является хорошими правилами в написании radability кода. Хочу заметить что в разных компаниях и проектах эти правила варьируются.

* **Самая явная манера кода предпочтительней над неявной.**
* **Одна строчка = одно выражение** (не пытайтесь все ваши мысли поместить в одну линию)

1) Формат документування для Python sphinx на прикладі:

2. def search(name):

"""

Searching items by name

:param name: String query string(key)

:return: Flask.Response -- JSON Item objects

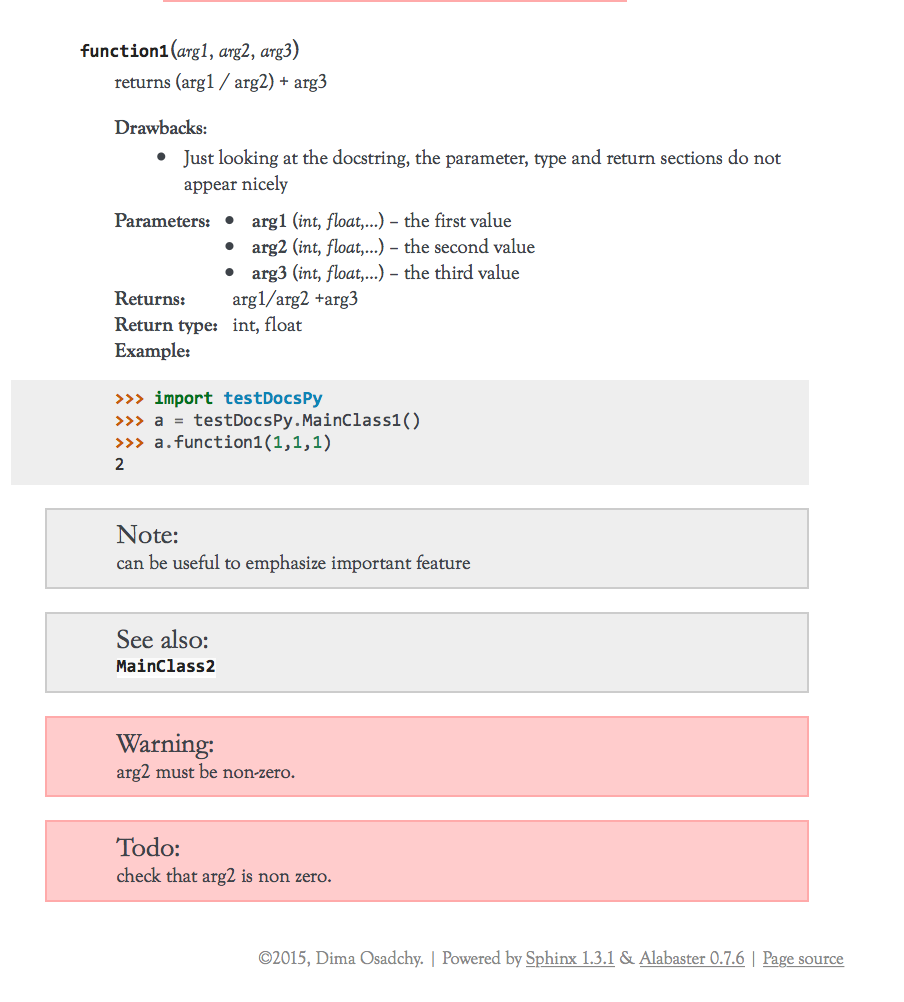
"""

searched = Item.objects(name\_\_icontains=name).all()

return searched.to\_json()

1. 2) Команди для авто-генерації довідника користувача:

2. sphinx-quickstart

3. create RST file for your program( example Code.rst )

3.1 Your file body :

Auto Generated Documentation

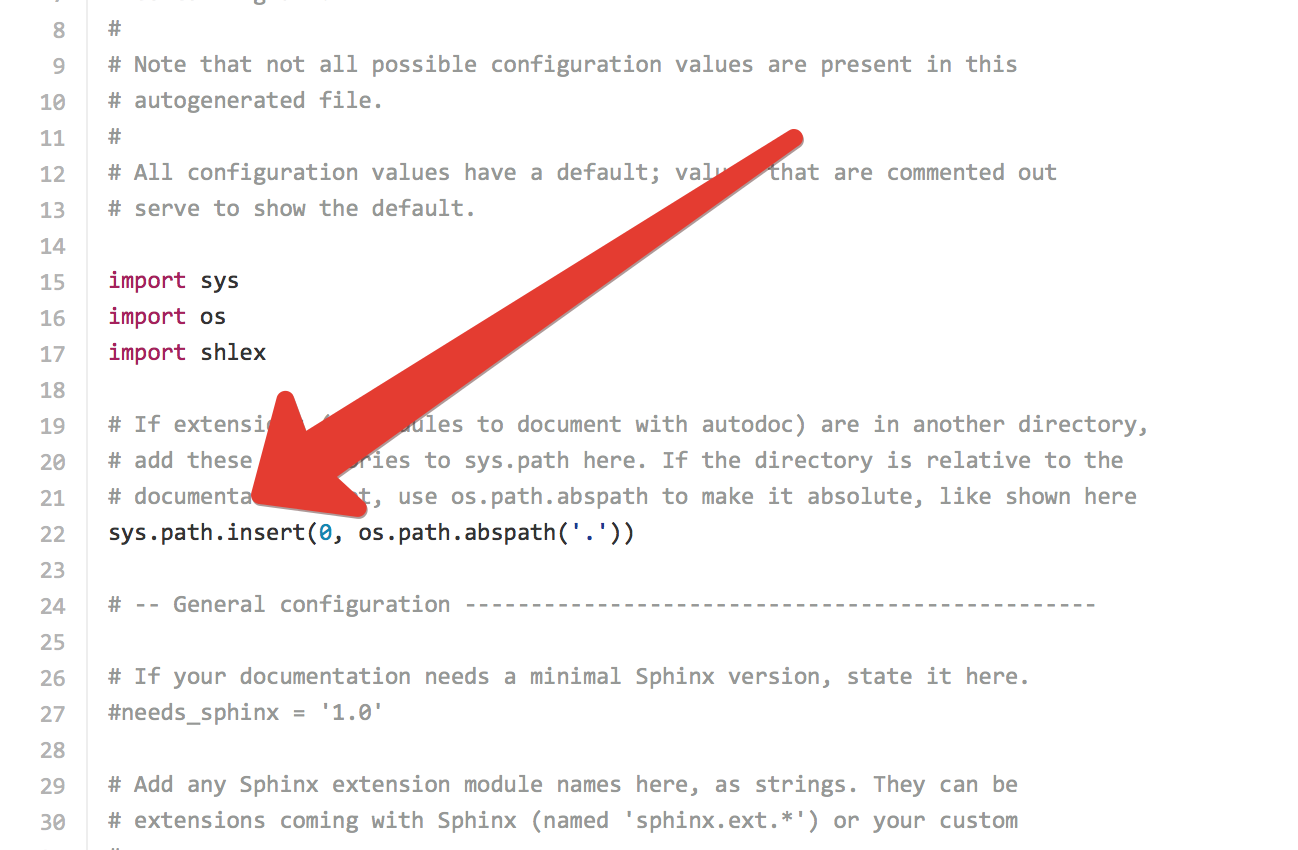
============================

.. automodule:: testDocsPy

:members:

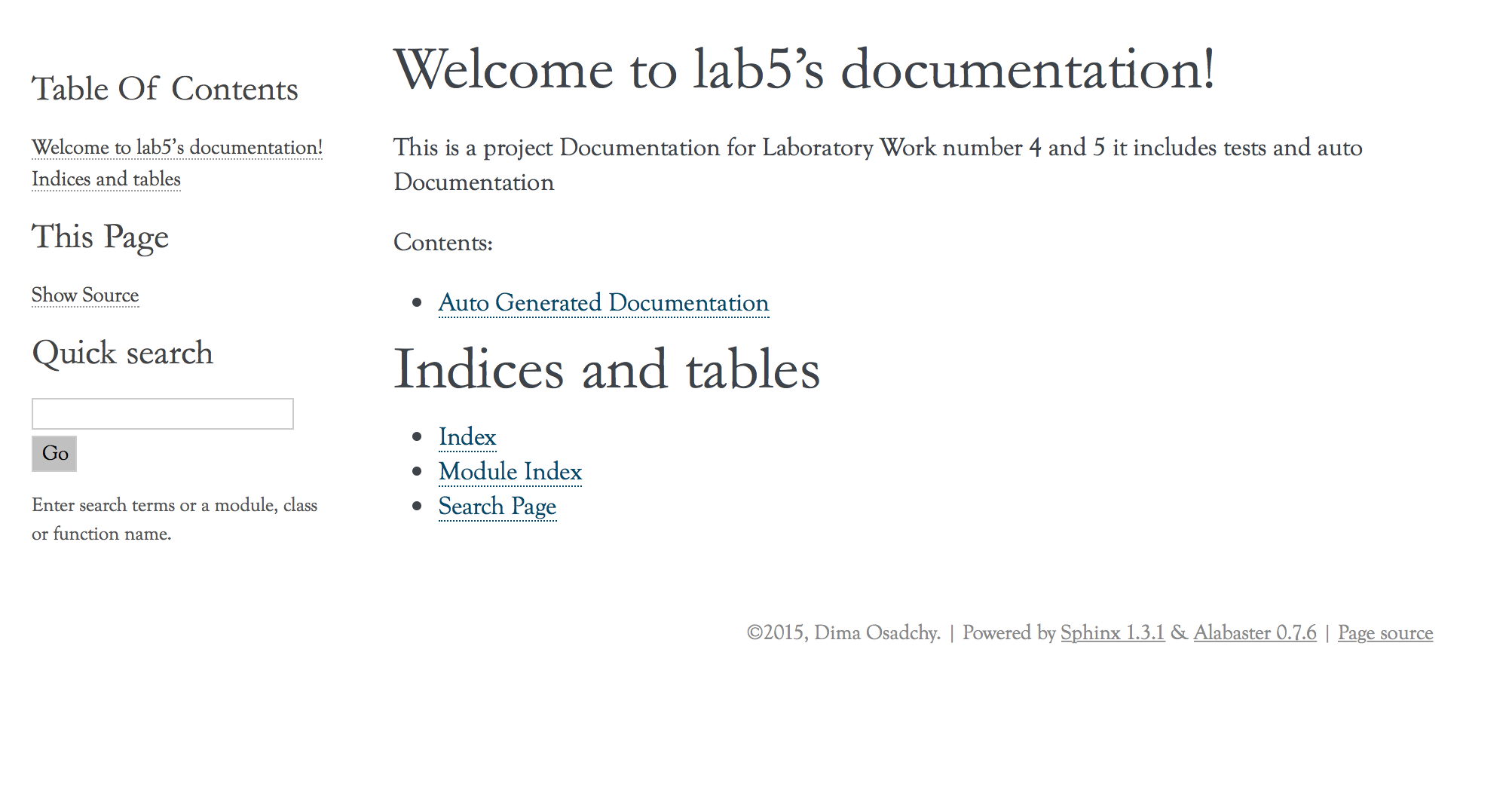
testDocsPy - my program name (testDocsPy.py)

4. uncomment this line - sys.path.insert(0, os.path.abspath(‘.')) in file conf.py

and specify your program name

5. make clean - in terminal

6. make html

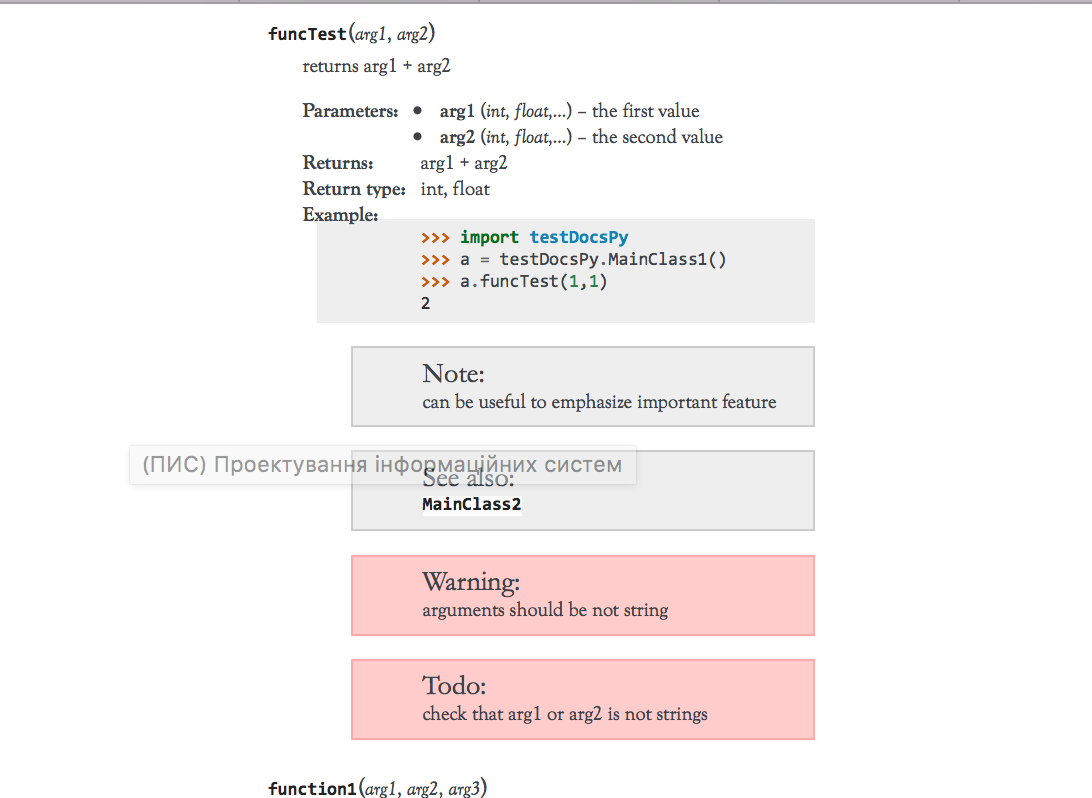
7. review your work in build/html

Самые популярные системы автоматической документации

• Sphinx documentation tool: <http://sphinx.pocoo.org/ext/autosummary.html>

• [autodoc](http://sphinx.pocoo.org/ext/autodoc.html), a Sphinx-based processor that processes/allows reST doc strings,

• [pdoc](https://github.com/BurntSushi/pdoc), a simple command line tool and library to auto generate API documentation for Python modules. Designed to replace epydoc and works on both Python 2 and 3. Includes support for cross-linking across modules, documentation for variables and namedtuples, and a built-in HTTP server to view documentation of local packages.

• [PyDoc](https://wiki.python.org/moin/PyDoc), <http://pydoc.org/> documentation browser (in HTML) and/or an off-line reference manual. Also in the standard library as [pydoc](http://docs.python.org/lib/module-pydoc.html)

• pydoctor, <https://launchpad.net/pydoctor> replacement for now inactive Epydoc, pydoctor was born for the needs of Twisted project

• Doxygen < [http://www.doxygen.org](http://www.doxygen.org/) > can create documentation in various formats (HTML, LaTeX, PDF, ...) and you can include formulas in your documentation (great for technical/mathematical software). Together with Graphviz < <http://www.research.att.com/sw/tools/graphviz/> > it can create diagrams of your code (inhertance diagram, call graph, ...). Another benefit is that it handles not only Python, but also several other programming languages like C, C++, Java, etc.

Я остановил свой выбор на sphnix так как это самая поддерживаемая система автоматической документации.

**Conclusion: We learned how to use auto documentation using sphinx as an example. Also we learned how to set up it and configure**