SIO2 Coding Style Cheat Sheet

2011-10-14

view plain copy to clipboard print ?

```
from elixir import *
import math


def global_function(full_argument):
    """Short summary here. If wraps, the next line starts
       aligned to the text in the first line. Write something
       more than "Calculates a number", but not longer than 3 lines.

       Then a more elaborate description, specifying what are the
       arguments and what is returned, what additional conditions
       are required (and they should be ``assert``-ed). You may
       also use the following buletted format:

       Args:
         ``list``: list to convert to string
         ``delimiter``: string to use as a delimiter; if the description
           wraps, following lines are indented by 2 more spaces

       Returns:
         a string containing concatenated string representations of
         element of ``list``, delimited by the specified ``delimiter``

       Amount of documentation is not limited. Feel free to add examples here.
       See :func:`sio2.core.util.parameterizable_decorator`.
    """
    return (1, 7, 'abc')


class Evaluator(Entity):
    """Short summary here. If wraps, the next line starts
       aligned to the text in the first line. Write something
       more than "Constructor.", but not longer than 3 lines.

       Then a more elaborate description. Always use reStructuredText:

       Process :class:`EvaluationJob`. To start processing, call
       :meth:`startEvaluation`. It will return an instance of
       :class:`EvaluationResult` and will start evaluation in the background.
       Subscribe to appropriate events to be notified when processing ends.
    """

    id = Field(String(255), primary_key=True)
    startDate = Field(DateTime)
    activeResult = OneToOne('SubmissionResult', inverse='activeForSubmission')
    """Documentation for ``activeResult``, if needed.

       Need to write a longer text? Use the class docstring.
    """

    ClassVariable = None
    """Documentation for ``ClassVariable``, if needed."""

    _InternalClassVariable = None
    __PrivateClassVariable = None

    @staticmethod
    def ClassMethod(name, full_name=None):
        """Chooses an object by name.

           Returns the ``id`` for the given ``name`` and, optionally,
           ``full_name``.  Returns ``None`` if not found.
        """
        return (2 + 3 / (127 + 42)) * (67 + len(full_name)) * \
               (67 ** 3) * 98

    def instanceMethod(self, name):
        """Single-line docstring may be ended at the same line."""
        local_variable = "(%s)" % (name)
        self.instanceVariable = None

    def _internalInstanceMethod(self):
        self._internalInstanceVariable = None

    def __privateInstanceMethod(self):
        self.__privateInstanceVariable = None
```

1. **Be consistent.**
   Be consistent!
   Be consistent!!
2. Prefer "from xyz import abc" over "import xyz as abc".
3. The only * import allowed is "from elixir import *".
4. One import per line.
5. One statement per line.
6. 4 spaces per indentation, no tabs.
7. Up to 79 chars per line.
8. Use exactly one space to surround assignment, binary and arthmetic operators.
9. Break line after binary operators (etc. and, <=), not before.
10. Do not put space before comma, semicolon or colon.
11. Do not put space after an opening bracket and before a closing bracket.
12. Do not surround "=" with space in default parameters.
13. Database-mapped fields always start with a lower case letter.
14. Relational fields should be named like 'evaluator' rather then 'forEvaluator' or 'ofEvaluator'.
15. Generally strings are in single quotes. Translatable strings should be in double quotes, at least until we will find out how do we support gettext and such.