

Coding Standards

1. Spacing

- Indent code with 4 spaces
- Existing code base contains mixed indention format
- Always indent code inside a method, loop, and sub code block
- Limit the line length to 79 characters
- Insert blank lines between each section of code
 - Between methods
 - Between loops, group of related statements inside a method

Whitespace

- Not recommended
 - after a parenthesis or brackets
 - before commas, semicolons, or colons
- Recommended between arithmetic operators and values

```
for (var i = 0; i < 10; i++) {  
    var num = 0;  
    doSomething(num, MAGIC_NUMBER);  
}
```

2. Naming Conventions

Class, Object

- Capitalise the first letter of each word of all classes and constructors

Python

```
class VideoDict:  
    // code ..
```

ActionScript

```
class upstage.util.LayerPickerSet extends MovieClip  
{  
    // code ..  
}
```

Normal Methods and Variables

- Use camel casing
- Use readable names that clearly define the purpose of the function or variable

```
// methods
doSomething()
getHighScore()

// variables
firstName
pngImage
```

Constants and Static Variables

- Use all capitals, and separate words with underscores
- Readable and descriptive name

Python

```
SWF_URL      = '/swf/'
MEDIA_URL    = '/media/'
```

ActionScript

```
public static var SCREEN_WIDTH    :Number = 320;
public static var SCREEN_HEIGHT  :Number = 200;
```

3. Commenting

- Do not state the blatantly obvious
- Use inline comments sparingly as it can distract the developer

Documentation Strings

- Put function/method documentation block where appropriate

Python

```
class Thing:
    """Base representation of a thing on stage"""
    // code ..

    def get_pos(self):
        """Retrieve tuple of x, y, z coordinates"""
        // method code ...
```

ActionScript

```
class upstage.Sender
{
    /**
     * @brief Constructor
     */
    function Sender(transport :Transport)
    {
        // code ..
    }

    /**
     * @brief Encode a message & send via <code>Transport</code>;
     */
    private function send(mode :String) :Void
    {
        // code ..
    }
}
```

Block Comment Format

Python

```
# Author: Craig Daniels (19-10-10)
# Purpose: Creates new players
# Modified by: Albert Turn (13-12-10) – Altered function to set player ids
```

ActionScript

```
/**
 * Author: Craig Daniels (19-10-10)
 * Purpose: Creates new players
 * Modified by: Albert Turn (13-12-10) – Altered function to set * player ids
 */
```

4. Miscellaneous

Import Statement

- Put each import statement on separate lines
- Group related import statements close together, separate each group with a blank line
 - Group 1: Standard library
 - Group 2: Related third-party modules/libraries
 - Group 3: Local modules/libraries

Python

- Importing multiple submodules from a module can be on the same line

```
import os
import sys
from module1 import moduleA, moduleB
```

ActionScript

- Import only the packages you will use
- State the package explicitly, do not use *

```
import mx.transitions.Tween;
```

ActionScript

- Separate commonly used objects / functions into an external file, and use #include directive
- Avoid attaching code to movie clips or buttons, instead use function calls when defining the action

UpStage Team

Version 1 – 6/8/13 – Nitkalya Wiriyanuparb

Last edited by dekpient, 3 months ago