# Source Code Highlight Filter

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| 8.2.7 | 4 July 2008 | | |

# Contents

The *AsciiDoc* distribution includes a source code syntax highlight filter (`source-highlight-filter.conf`). It uses GNU source-highlight to highlight HTML outputs; DocBook outputs are highlighted by toolchains that have `programlisting` element highlight support, for example *dblatex*.

**Tip**
If the source *language* attribute has been set (using an *AttributeEntry* or from the command-line) you don't have to specify it in each source code block.

# 1 Examples

## 1.1 Source code paragraphs

The `source` paragraph style will highlight a paragraph of source code. These three code paragraphs:

```
[source,python]
if n < 0: print 'Hello World!'

:language: python

[source]
if n < 0: print 'Hello World!'

[source,ruby,numbered]
[true, false].cycle([0, 1, 2, 3, 4]) do |a, b|
    puts "#{a.inspect} => #{b.inspect}"
```

Render this highlighted source code:

```
if n < 0: print 'Hello World!'
```

```
if n < 0: print 'Hello World!'
```

```
1 [true, false].cycle([0, 1, 2, 3, 4]) do |a, b|
2     puts "#{a.inspect} => #{b.inspect}"
```

## 1.2 Unnumbered source code listing

This source-highlight filtered block:

```
[source,python]
---------------------------------------------------------------------
''' A multi-line
    comment.'''
def sub_word(mo):
    ''' Single line comment.'''
    word = mo.group('word')    # Inline comment
    if word in keywords[language]:
        return quote + word + quote
    else:
        return word
---------------------------------------------------------------------
```

Renders this highlighted source code:

```
''' A multi-line
    comment.'''
def sub_word(mo):
    ''' Single line comment.'''
    word = mo.group('word')      # Inline comment
    if word in keywords[language]:
        return quote + word + quote
    else:
        return word
```

### 1.3 Numbered source code listing

This source-highlight filtered block:

```
[source,ruby,numbered]
----------------------------------------------------------------------
#
# Useful Ruby base class extensions.
#

class Array

  # Execute a block passing it corresponding items in
  # +self+ and +other_array+.
  # If self has less items than other_array it is repeated.

  def cycle(other_array)  # :yields: item, other_item
    other_array.each_with_index do |item, index|
      yield(self[index % self.length], item)
    end
  end

end

if $0 == __FILE__
  # Array#cycle test
  # true => 0
  # false => 1
  # true => 2
  # false => 3
  # true => 4
  puts 'Array#cycle test'
  [true, false].cycle([0, 1, 2, 3, 4]) do |a, b|
    puts "#{a.inspect} => #{b.inspect}"
  end
end
----------------------------------------------------------------------
```

Renders this highlighted source code:

```
1  #
2  # Useful Ruby base class extensions.
3  #
4
5  class Array
6
7    # Execute a block passing it corresponding items in
8    # +self+ and +other_array+.
9    # If self has less items than other_array it is repeated.
10
```

```
  def cycle(other_array)  # :yields: item, other_item
    other_array.each_with_index do |item, index|
      yield(self[index % self.length], item)
    end
  end

end

if $0 == __FILE__
  # Array#cycle test
  # true => 0
  # false => 1
  # true => 2
  # false => 3
  # true => 4
  puts 'Array#cycle test'
  [true, false].cycle([0, 1, 2, 3, 4]) do |a, b|
    puts "#{a.inspect} => #{b.inspect}"
  end
end
```

## 2 Installation

If you want to syntax highlight *AsciiDoc* HTML outputs you need to install GNU source-highlight (most distributions have this package). It's not required if you are generating DocBook output (DocBook syntax highlighting is handled by the DocBook toolchain).

Test the filter by converting the test file to HTML with *AsciiDoc*:

```
$ asciidoc -v ./filters/source-highlight-filter-test.txt
$ firefox ./filters/source-highlight-filter-test.html &
```