

Testing the **coder** package

Jérôme LAURENS

March 27, 2022

1 Have **pygments**?

```
Path=/Users/jlaurens/opt/anaconda3/bin/pygmentize
WITH PYGMENTS
Path=/Users/jlaurens/opt/anaconda3/bin/python
```

2 Testing the tag mechanism

Tests of the tag mechanism.

2.1 groups

```
THIS_IS_C=THIS_IS_C
THIS_IS_VOID=
```

2.2 Other

2.3 Inheritance

3 \CDR_tag_choices_set:

```
Tests of \CDR_tag_choices_set:
9
```

4 \CDR_tag_boolean_set:

```
Tests of \CDR_tag_boolean_set:n
```

5 int

```
Tests of int
SUCCESS
SUCCESS
SUCCESS
SUCCESS
```


6.1 default

6.2 tag/default

6.3 __pygments

TRUE==TRUE

FALSE==FALSE

6.4 tag/<name>:__pygments

6.5 __pygments.block

6.6 tag/<name>:__pygments.block

6.7 default.code

6.8 tag/<name>:default.code

6.9 default.block

6.10 tag/<name>:default.block

X X X

6.11 __fancyvrb

6.12 tag/<name>:__fancyvrb

6.13 __fancyvrb.block

6.14 tag/<name>:__fancyvrb.block

6.15 __fancyvrb.all

6.16 tag/<name>:__fancyvrb.all

7 \CDRCode

7.1 Fancyvrb basics

ABCDEABCDEBCDGBCDG

First_□verba ~~†~~**tim_□line.**

Second_□verbatim_□line.

See the verbatim on page ??

YYY **A_□B_□C**YYY YY YY

7.2 Mimic

A_□B_□C_□D A B C D

7.3 Coder

```
A
<fontsize=\Large>
B
<fontsize=\Large>
A
<fontsize=\Large>
B
<fontsize=\Large>

<ABC DEF>
1
<fontfamily=menlo>
2
<showspaces_\_true>
<showspaces_\_true>
3
<fontsize=\Large>
4
Expecting ERROR:
\PackageError coder
flower code engine unknown, replaced by `default'
See \CDRCODEENGINENew in the coder manual
,<engine=flower>
5
wavy(<engine=wave>)
LAST
<THIS IS THE LAST!!!>
```

7.4 With pygments

```
python==/Users/jlaurens/opt/anaconda3/bin/python
PATH==/usr/bin/python
PATH==/Users/jlaurens/opt/anaconda3/bin/python
```

```
PYGMENTS TEST
ABCD=AB CD
ABCD=AB CD\textbf{AB}
ABCD=AB CD\textbf{AB}
ABCD=AB CD\textbf{AB}
ABCD=AB CD\textbf{AB}
ABCD=AB CD\textbf{AB}
AB CD=AB CD\textbf{AB} et la suite
AB CD=AB CD\textbf{AB} et la suite
AB CD=AB CD\textbf{AB} et la suite
```

8 CDRBlock

ESSAI

[]
ESSAI

ESSAI

1	local f=function(arg)	X
2	return arg**arg	X
3	end	X

local f = function(arg)	X 0
return arg ** arg	X 1
end	X 2

local f = function(arg)	X
-- 2	X 1
-- 3	X
-- 4	X
-- 5	X
return arg ** arg	X
end	X 6

Next should not be void

def foo(arg):	X
1 return arg ** arg	X

Next should not be void

function foo(arg)	X
1 return arg**arg	X
end	X

8.1 Line numbering

8.1.1 fancyvrb linear

A

A	X
none: A	X
3 B	X
4 C	X
none: A	X
9 B	X
10 C	X
none: A	X
15 B	X
16 C	X

8.1.2 fancyvrb multi tags

A,B,C: 1 A=*1,B=1,C=1	X
B,C: A=2,B=*2,C=2	X
4 A=2,B=*3,C=3	X
C: A=2,B=4,C=*4	X
8 A=2,B=4,C=*5	X
C,B: A=2,B=4,C=*6	X
12 A=2,B=5,C=*7	X
13 A=2,B=6,C=*8	X
B,A: A=2,B=*7,C=9	X
14 A=3,B=*8,C=9	X
15 A=4,B=*9,C=9	X
A,C: A=*5,B=10,C=9	X
10 A=*6,B=10,C=10	X
11 A=*7,B=10,C=11	X

FAILED
 FAILED
 FAILED

8.1.3 fancyvrb properties

none: A	X
21 B	X
22 C	X

8.1.4 Display tags

tests start here:

PYGMENTS:true

PYGMENTS:true

TAGS:

TAGS:NONE,ENON

```
function foo(arg) return 0 end
```

X 33

NONE:2

ENON:2

```
function foo(arg) -- no tags expected
```

X 2

2

X 3

3

X 4

4

X 5

5

X 6

6

X 7

```
end
```

X 8

NONE:8

ENON:8

```
function foo(arg) -- no tags expected
```

X 1

2

X 2

3

X 3

4

X 4

5

X 5

6

X 6

```
end
```

X 7

NONE:8

ENON:8

```
function foo(arg) -- no tags expected
```

X

-- second line

X

-- third line

X

-- fourth line

X

-- fifth line

X

return arg ** arg

X

```
end
```

X

NONE,ENON: 33

```
function foo(arg) return arg ** arg end -- tagsXexpected
```

whatever it

isp,we are the

champions: 33

```
function foo(arg) return arg ** arg end -- tagsXexpected
```

<code>function foo(arg) --no_tags_expected</code>	X
<code>--second_line</code>	X
<code>--third_line</code>	X
<code>--fourth_line</code>	X
<code>--fifth_line</code>	X
<code>return arg**arg</code>	X
<code>end</code>	X

8.1.5 Exhaustive test

<code>function foo(arg) return arg ** arg end</code>	X
--	---

<code>function foo(arg) return arg ** arg end</code>	X
--	---

<code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X

<code>function foo(arg) return arg ** arg end</code>	X
--	---

<code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X

<code>function foo(arg) return arg ** arg end</code>	X
--	---

<code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X

<code>function foo(arg) return arg ** arg end</code>	X
--	---

function foo(arg)	X
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
 1 function foo(arg) return arg ** arg end	 X
 2 function foo(arg)	 X
3 -- second line	X
4 -- third line	X
5 -- fourth line	X
6 -- fifth line	X
7 return arg ** arg	X
8 end	X
 1 function foo(arg) return arg ** arg end	 X
 10 function foo(arg)	 X
11 -- second line	X
12 -- third line	X
13 -- fourth line	X
14 -- fifth line	X
15 return arg ** arg	X
16 end	X
 1 function foo(arg) return arg ** arg end	 X
 2 function foo(arg)	 X
3 -- second line	X
4 -- third line	X
5 -- fourth line	X
6 -- fifth line	X
7 return arg ** arg	X
8 end	X
 1 function foo(arg) return arg ** arg end	 X
 10 function foo(arg)	 X
11 -- second line	X
12 -- third line	X
13 -- fourth line	X
14 -- fifth line	X
15 return arg ** arg	X
16 end	X

```

function foo(arg) return arg ** arg end      X 1

function foo(arg)                             X 2
  -- second line                             X 3
  -- third line                              X 4
  -- fourth line                             X 5
  -- fifth line                              X 6
  return arg ** arg                          X 7
end                                            X 8

function foo(arg) return arg ** arg end      X 1

function foo(arg)                             X 10
  -- second line                             X 11
  -- third line                              X 12
  -- fourth line                             X 13
  -- fifth line                              X 14
  return arg ** arg                          X 15
end                                            X 16

function foo(arg) return arg ** arg end      X 1

function foo(arg)                             X 2
  -- second line                             X 3
  -- third line                              X 4
  -- fourth line                             X 5
  -- fifth line                              X 6
  return arg ** arg                          X 7
end                                            X 8

function foo(arg) return arg ** arg end      X 1

function foo(arg)                             X 10
  -- second line                             X 11
  -- third line                              X 12
  -- fourth line                             X 13
  -- fifth line                              X 14
  return arg ** arg                          X 15
end                                            X 16

WHAT: function foo(arg) return arg ** arg end  X

```

WHAT: <code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
WHAT: <code>function foo(arg) return arg ** arg end</code>	X
WHAT: <code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
WHAT: <code>function foo(arg) return arg ** arg end</code>	X
WHAT: <code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
WHAT: <code>function foo(arg) return arg ** arg end</code>	X
WHAT: <code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
WHAT: <code>1 function foo(arg) return arg ** arg end</code>	X
WHAT: <code>function foo(arg)</code>	X
<code>3 -- second line</code>	X
<code>4 -- third line</code>	X
<code>5 -- fourth line</code>	X
<code>6 -- fifth line</code>	X
<code>7 return arg ** arg</code>	X
<code>8 end</code>	X

```

WHAT: 1 function foo(arg) return arg ** arg end      X

WHAT: function foo(arg)                             X
11  -- second line                                  X
12  -- third line                                    X
13  -- fourth line                                   X
14  -- fifth line                                    X
15  return arg ** arg                                X
16 end                                               X

WHAT: 1 function foo(arg) return arg ** arg end      X

WHAT: function foo(arg)                             X
3   -- second line                                  X
4   -- third line                                    X
5   -- fourth line                                   X
6   -- fifth line                                    X
7   return arg ** arg                                X
8 end                                               X

WHAT: 1 function foo(arg) return arg ** arg end      X

WHAT: function foo(arg)                             X
11  -- second line                                  X
12  -- third line                                    X
13  -- fourth line                                   X
14  -- fifth line                                    X
15  return arg ** arg                                X
16 end                                               X

WHAT: function foo(arg) return arg ** arg end      X 1

WHAT: function foo(arg)                             X 2
    -- second line                                  X 3
    -- third line                                    X 4
    -- fourth line                                   X 5
    -- fifth line                                    X 6
    return arg ** arg                                X 7
end                                                  X 8

WHAT: function foo(arg) return arg ** arg end      X 1

```

WHAT: function foo(arg)	X 10
-- second line	X 11
-- third line	X 12
-- fourth line	X 13
-- fifth line	X 14
return arg ** arg	X 15
end	X 16
WHAT: function foo(arg) return arg ** arg end	X 1
WHAT: function foo(arg)	X 2
-- second line	X 3
-- third line	X 4
-- fourth line	X 5
-- fifth line	X 6
return arg ** arg	X 7
end	X 8
WHAT: function foo(arg) return arg ** arg end	X 1
WHAT: function foo(arg)	X 10
-- second line	X 11
-- third line	X 12
-- fourth line	X 13
-- fifth line	X 14
return arg ** arg	X 15
end	X 16
function foo(arg) return arg ** arg end	X : WHAT
function foo(arg)	X : WHAT
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
function foo(arg) return arg ** arg end	X : WHAT
function foo(arg)	X : WHAT
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X

<code>function foo(arg) return arg ** arg end</code>	X : WHAT
<code>function foo(arg)</code>	X : WHAT
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
<code>function foo(arg) return arg ** arg end</code>	X : WHAT
<code>function foo(arg)</code>	X : WHAT
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
<code>1 function foo(arg) return arg ** arg end</code>	X : WHAT
<code>2 function foo(arg)</code>	X : WHAT
<code>3 -- second line</code>	X
<code>4 -- third line</code>	X
<code>5 -- fourth line</code>	X
<code>6 -- fifth line</code>	X
<code>7 return arg ** arg</code>	X
<code>8 end</code>	X
<code>1 function foo(arg) return arg ** arg end</code>	X : WHAT
<code>10 function foo(arg)</code>	X : WHAT
<code>11 -- second line</code>	X
<code>12 -- third line</code>	X
<code>13 -- fourth line</code>	X
<code>14 -- fifth line</code>	X
<code>15 return arg ** arg</code>	X
<code>16 end</code>	X
<code>1 function foo(arg) return arg ** arg end</code>	X : WHAT

2 function foo(arg)	X : WHAT
3 -- second line	X
4 -- third line	X
5 -- fourth line	X
6 -- fifth line	X
7 return arg ** arg	X
8 end	X
1 function foo(arg) return arg ** arg end	X : WHAT
10 function foo(arg)	X : WHAT
11 -- second line	X
12 -- third line	X
13 -- fourth line	X
14 -- fifth line	X
15 return arg ** arg	X
16 end	X
function foo(arg) return arg ** arg end	X 1: WHAT
function foo(arg)	X : WHAT
-- second line	X 3
-- third line	X 4
-- fourth line	X 5
-- fifth line	X 6
return arg ** arg	X 7
end	X 8
function foo(arg) return arg ** arg end	X 1: WHAT
function foo(arg)	X : WHAT
-- second line	X 11
-- third line	X 12
-- fourth line	X 13
-- fifth line	X 14
return arg ** arg	X 15
end	X 16
function foo(arg) return arg ** arg end	X 1: WHAT
function foo(arg)	X : WHAT
-- second line	X 3
-- third line	X 4
-- fourth line	X 5
-- fifth line	X 6
return arg ** arg	X 7
end	X 8

function foo(arg) return arg ** arg end	X 1: WHAT
function foo(arg)	X : WHAT
-- second line	X 11
-- third line	X 12
-- fourth line	X 13
-- fifth line	X 14
return arg ** arg	X 15
end	X 16
function foo(arg) return arg ** arg end	X
function foo(arg)	X
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
function foo(arg) return arg ** arg end	X
function foo(arg)	X
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
function foo(arg) return arg ** arg end	X
function foo(arg)	X
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
function foo(arg) return arg ** arg end	X

function foo(arg)	X
-- second line	X
-- third line	X
-- fourth line	X
-- fifth line	X
return arg ** arg	X
end	X
WHAT: 1 function foo(arg) return arg ** arg end	X
WHAT: function foo(arg)	X
3 -- second line	X
4 -- third line	X
5 -- fourth line	X
6 -- fifth line	X
7 return arg ** arg	X
8 end	X
WHAT: 1 function foo(arg) return arg ** arg end	X
WHAT: function foo(arg)	X
11 -- second line	X
12 -- third line	X
13 -- fourth line	X
14 -- fifth line	X
15 return arg ** arg	X
16 end	X
WHAT: 1 function foo(arg) return arg ** arg end	X
WHAT: function foo(arg)	X
3 -- second line	X
4 -- third line	X
5 -- fourth line	X
6 -- fifth line	X
7 return arg ** arg	X
8 end	X
WHAT: 1 function foo(arg) return arg ** arg end	X
WHAT: function foo(arg)	X
11 -- second line	X
12 -- third line	X
13 -- fourth line	X
14 -- fifth line	X
15 return arg ** arg	X
16 end	X

<code>function foo(arg) return arg ** arg end</code>	X 1: WHAT
 <code>function foo(arg)</code>	 X : WHAT
<code>-- second line</code>	X 3
<code>-- third line</code>	X 4
<code>-- fourth line</code>	X 5
<code>-- fifth line</code>	X 6
<code>return arg ** arg</code>	X 7
<code>end</code>	X 8
 <code>function foo(arg) return arg ** arg end</code>	 X 1: WHAT
 <code>function foo(arg)</code>	 X : WHAT
<code>-- second line</code>	X 11
<code>-- third line</code>	X 12
<code>-- fourth line</code>	X 13
<code>-- fifth line</code>	X 14
<code>return arg ** arg</code>	X 15
<code>end</code>	X 16
 <code>function foo(arg) return arg ** arg end</code>	 X 1: WHAT
 <code>function foo(arg)</code>	 X : WHAT
<code>-- second line</code>	X 3
<code>-- third line</code>	X 4
<code>-- fourth line</code>	X 5
<code>-- fifth line</code>	X 6
<code>return arg ** arg</code>	X 7
<code>end</code>	X 8
 <code>function foo(arg) return arg ** arg end</code>	 X 1: WHAT
 <code>function foo(arg)</code>	 X : WHAT
<code>-- second line</code>	X 11
<code>-- third line</code>	X 12
<code>-- fourth line</code>	X 13
<code>-- fifth line</code>	X 14
<code>return arg ** arg</code>	X 15
<code>end</code>	X 16
 <code>function foo(arg) return arg ** arg end</code>	 X

<code>function foo(arg)</code>	X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
 <code>function foo(arg) return arg ** arg end</code>	 X
 <code>function foo(arg)</code>	 X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
 <code>function foo(arg) return arg ** arg end</code>	 X
 <code>function foo(arg)</code>	 X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
 <code>function foo(arg) return arg ** arg end</code>	 X
 <code>function foo(arg)</code>	 X
<code>-- second line</code>	X
<code>-- third line</code>	X
<code>-- fourth line</code>	X
<code>-- fifth line</code>	X
<code>return arg ** arg</code>	X
<code>end</code>	X
 <code>1 function foo(arg) return arg ** arg end</code>	 X : WHAT
 <code>2 function foo(arg)</code>	 X : WHAT
<code>3 -- second line</code>	X
<code>4 -- third line</code>	X
<code>5 -- fourth line</code>	X
<code>6 -- fifth line</code>	X
<code>7 return arg ** arg</code>	X
<code>8 end</code>	X

```

1 function foo(arg) return arg ** arg end      X : WHAT

10 function foo(arg)                           X : WHAT
11   -- second line                           X
12   -- third line                            X
13   -- fourth line                           X
14   -- fifth line                            X
15   return arg ** arg                        X
16 end                                         X

1 function foo(arg) return arg ** arg end      X : WHAT

2 function foo(arg)                           X : WHAT
3   -- second line                           X
4   -- third line                            X
5   -- fourth line                           X
6   -- fifth line                            X
7   return arg ** arg                        X
8 end                                         X

1 function foo(arg) return arg ** arg end      X : WHAT

10 function foo(arg)                           X : WHAT
11   -- second line                           X
12   -- third line                            X
13   -- fourth line                           X
14   -- fifth line                            X
15   return arg ** arg                        X
16 end                                         X

WHAT: function foo(arg) return arg ** arg end  X 1

WHAT: function foo(arg)                       X 2
      -- second line                         X 3
      -- third line                         X 4
      -- fourth line                       X 5
      -- fifth line                       X 6
      return arg ** arg                     X 7
      end                                   X 8

WHAT: function foo(arg) return arg ** arg end  X 1

```

```

WHAT: function foo(arg)                X 10
      -- second line                   X 11
      -- third line                    X 12
      -- fourth line                   X 13
      -- fifth line                    X 14
      return arg ** arg                 X 15
    end                                 X 16

```

```

WHAT: function foo(arg) return arg ** arg end X 1

```

```

WHAT: function foo(arg)                X 2
      -- second line                   X 3
      -- third line                    X 4
      -- fourth line                   X 5
      -- fifth line                    X 6
      return arg ** arg                 X 7
    end                                 X 8

```

```

WHAT: function foo(arg) return arg ** arg end X 1

```

```

WHAT: function foo(arg)                X 10
      -- second line                   X 11
      -- third line                    X 12
      -- fourth line                   X 13
      -- fifth line                    X 14
      return arg ** arg                 X 15
    end                                 X 16

```

9 cache

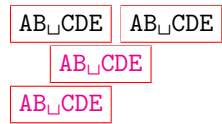
ABCDE

10 \CDRExport

A: 1	X
2 2	X
3 3	X
B,C,D: 1 B,C,D: 1	X
B,C: 1 B,C: 2	X
D,B: 1 D,B: 3	X
C,D: 1 C,D: 4	X
B: 1 B: 5	X
C: 1 C: 6	X
D: 1 D: 7	X

11 Engine

11.1 Code



11.2 Block

D: def function(x):	X
2 '''Square'''	X
3 return x * x	X
D: def function(x):	X
2 '''Square'''	X
3 return x * x	X
D: def function(x):	X
2 '''Square'''	X
3 return x * x	X

11.3 POC

1. Lorem.

2.



3.



4.



5. Epsum.

abcde



11.3.1 POC A

abcde

POC A

```
def function(x):  
    '''Square'''  
    return x * x
```

abcde

```
def function(x):  
    '''Square'''  
    return x * x
```

abcde

11.3.2 POC B

abcde

POCB

def function(x):	X
'''Square'''	X
return x * x	X

abcde

11.3.3 DEBUG

abcde

DEBUG

D:	def function(x):	X
2	'''Square'''	X
3	return x * x	X

abcde

DEBUG

D:	def function(x):	X
2	'''Square'''	X
3	return x * x	X

abcde

DEBUG		
D:	<code>def function(x):</code>	X
2	<code> '''Square'''</code>	X
3	<code> return x * x</code>	X

abcde

DEBUG		
D:	<code>def function(x):</code>	X
2	<code> '''Square'''</code>	X
3	<code> return x * x</code>	X

11.4 In progress

abcde

ABCDE

abcdeabcde

WHAT:	<code>def function foo(x):</code>	X
2	<code> '''Square function'''</code>	X
3	<code> return x * x</code>	X

abcde

WHAT:	<code>def function foo(x):</code>	X
2	<code> '''Square function'''</code>	X
3	<code> return x * x</code>	X

abcde

WHAT:	<code>def function foo(x):</code>	X
2	<code> '''Square function'''</code>	X
3	<code> return x * x</code>	X

abcde