Pluralize — *Smooth CoffeeScript*

This literate program is *interactive* in its HTML form. Edit a CoffeeScript segment to try it.

How do you add 's' to a word when there is more than one of something?

It is often seen in JavaScript with the ternary operator ?: but that is hard-coding at the expense of future maintenance and internationalization.

First some preliminaries. Let's `show` the results whether we are running in a server or a web client. Something like `show = if exports`? `then console.log else alert`. And wrap the different methods in a `test` function so we can see if they work.

```
test = (points) ->
```

Readable ways

The simplest readable way

```
  show ”0: You got #{points} point#{if points > 1 then 's' else ''}”
```

Encapsulate in a function, future i18n will be easier

```
  pluralIf = (stem, cond) -> stem + (if cond then 's' else '')
  show ”1: You got #{points} #{pluralIf 'point', points > 1}”
```

Encapsulate in a function, handle negative numbers

```
  pluralUnless = (stem, cond) -> stem + (unless cond then 's' else '')
  show ”2: You got #{points} #{pluralUnless 'point', -2 < points < 2}”
```

Encapsulate in a String method, could collide with another definition

```
  String::pluralIf ?= (cond) -> this + (if cond then 's' else '')
  show ”3: You got #{points} #{'point'.pluralIf points > 1}”
```

Tricks

Use the existential operator to catch the singular undefined

```
  show ”4: You got #{points} point#{('s' if points > 1) ? ''}”
```

Convert the condition to a number and do an array lookup

```
  show ”5: You got #{points} point#{['','s'][+(points > 1)]}”
```

Use the inverted condition to ask for a letter.

`true => charAt 1 => '' — false => charAt 0 => 's'`

```
  show ”6: You got #{points} point#{'s'.charAt points <= 1}”
```

Create an array whose length is determined by the condition. It consists of empty elements so join it with 's'

```
  show ”7: You got #{points} point#{Array(1+(points>1)).join 's'}”
```

Concatenate to an empty array, undefined does nothing

```
  show ”8: You got #{points} point#{[].concat ('s' if points > 1)}”
```

Contributed

Create an array with either 's' or undefined — from satyr

```
show "9: You got #{points} point#{['s' if points > 1]}"
```

Test

Run a test for a small range of numbers

```
test n for n in [-3..3]
```

Output

```
1    0: You got -3 point
2    1: You got -3 point
3    2: You got -3 points
4    3: You got -3 point
5    4: You got -3 point
6    5: You got -3 point
7    6: You got -3 point
8    7: You got -3 point
9    8: You got -3 point
10   9: You got -3 point
11   0: You got -2 point
12   1: You got -2 point
13   2: You got -2 points
14   3: You got -2 point
15   4: You got -2 point
16   5: You got -2 point
17   6: You got -2 point
18   7: You got -2 point
19   8: You got -2 point
20   9: You got -2 point
21   0: You got -1 point
22   1: You got -1 point
23   2: You got -1 point
24   3: You got -1 point
25   4: You got -1 point
26   5: You got -1 point
27   6: You got -1 point
28   7: You got -1 point
29   8: You got -1 point
30   9: You got -1 point
31   0: You got 0 point
32   1: You got 0 point
33   2: You got 0 point
34   3: You got 0 point
35   4: You got 0 point
36   5: You got 0 point
37   6: You got 0 point
38   7: You got 0 point
39   8: You got 0 point
40   9: You got 0 point
41   0: You got 1 point
42   1: You got 1 point
43   2: You got 1 point
44   3: You got 1 point
45   4: You got 1 point
46   5: You got 1 point
47   6: You got 1 point
48   7: You got 1 point
49   8: You got 1 point
```

```
50  9: You got 1 point
51  0: You got 2 points
52  1: You got 2 points
53  2: You got 2 points
54  3: You got 2 points
55  4: You got 2 points
56  5: You got 2 points
57  6: You got 2 points
58  7: You got 2 points
59  8: You got 2 points
60  9: You got 2 points
61  0: You got 3 points
62  1: You got 3 points
63  2: You got 3 points
64  3: You got 3 points
65  4: You got 3 points
66  5: You got 3 points
67  6: You got 3 points
68  7: You got 3 points
69  8: You got 3 points
70  9: You got 3 points
```

JavaScript

```
1   (function() {
2     var n, show, showDocument, test;
3
4     show = console.log;
5
6     showDocument = function(doc, width, height) {
7       return show(doc);
8     };
9
10    test = function(points) {
11      var pluralIf, pluralUnless, _base, _ref, _ref2;
12      show("0: You got " + points + " point" + (points > 1 ? 's' : ''));
13      pluralIf = function(stem, cond) {
14        return stem + (cond ? 's' : '');
15      };
16      show("1: You got " + points + " " + (pluralIf('point', points > 1)));
17      pluralUnless = function(stem, cond) {
18        return stem + (!cond ? 's' : '');
19      };
20      show("2: You got " + points + " " + (pluralUnless('point', (-2 < points && points < 2))));
21      if ((_ref = (_base = String.prototype).pluralIf) == null) {
22        _base.pluralIf = function(cond) {
23          return this + (cond ? 's' : '');
24        };
25      }
26      show("3: You got " + points + " " + ('point'.pluralIf(points > 1)));
27      show("4: You got " + points + " point" + ((_ref2 = (points > 1 ? 's' : void 0)) != null ? _ref2 : ''));
28      show("5: You got " + points + " point" + ['', 's'][+(points > 1)]);
29      show("6: You got " + points + " point" + ('s'.charAt(points <= 1)));
30      show("7: You got " + points + " point" + (Array(1 + (points > 1)).join('s')));
31      show("8: You got " + points + " point" + ([].concat((points > 1 ? 's' : void 0))));
32      return show("9: You got " + points + " point" + [points > 1 ? 's' : void 0]);
33    };
34
35    for (n = -3; n <= 3; n++) {
36      test(n);
37    }
38
39  }).call(this);
```

Formats CoffeeScript Markdown PDF HTML