

# Using a Vector Class

Copyright (c) 2017, 2019 Tor Olav Kristensen, <http://subcube.com> (<http://subcube.com>).

<https://github.com/t-o-k/scikit-vectors> (<https://github.com/t-o-k/scikit-vectors>).

Use of this source code is governed by a BSD-license that can be found in the LICENSE file.

```
In [2]: 1 from skvectors import create_class_Vector
```

```
In [3]: 1 # Create a 3-dimensional vector class
2
3 VC = create_class_Vector('VC', 'abc')
4
5 # Explicit alternative:
6 # VC = \
7 #     create_class_Vector(
8 #         name = 'VC',
9 #         component_names = [ 'a', 'b', 'c' ],
10 #         brackets = [ '<', '>' ],
11 #         sep = ', ',
12 #         cnull = 0,
13 #         cunit = 1,
14 #         functions = None
15 #     )
```

```
In [4]: 1 # Null value for vector components in the class
2 VC.component_null()
```

Out[4]: 0

```
In [5]: 1 # Unit value for vector components in the class
2 VC.component_unit()
```

Out[5]: 1

```
In [7]: 1 # Basis vectors in class
        2 VC.basis_a(), VC.basis_b(), VC.basis_c()
```

```
Out[7]: (VC(a=1, b=0, c=0), VC(a=0, b=1, c=0), VC(a=0, b=0, c=1))
```

```
In [9]: 1 # Create a vector with all the components set to the cnull value
        2 VC.zero()
```

```
Out[9]: VC(a=0, b=0, c=0)
```

```
In [10]: 1 # Create a vector with all the components set to the cunit value
         2 VC.one()
```

```
Out[10]: VC(a=1, b=1, c=1)
```

```
In [11]: 1 # Null value for vector components
         2 v = VC(7, -8, 9)
         3 v.cnull
```

```
Out[11]: 0
```

```
In [12]: 1 # Unit value for vector components
         2 v = VC(7, -8, 9)
         3 v.cunit
```

```
Out[12]: 1
```

```
In [13]: 1 # Sum of component values in vector
         2 v = VC(-3, 4, 5)
         3 v.csum
```

```
Out[13]: 6
```

```
In [14]: 1 # Product of component values in vector
         2 v = VC(-3, 4, 5)
         3 v.cprod
```

```
Out[14]: -60
```

```
In [15]: 1 # Check if vector is zero vector
         2 v = VC.zero()
         3 v.is_zero_vector()
```

Out[15]: True

```
In [16]: 1 # Check if vector is zero vector
         2 v = VC(0, 1e-14, 0)
         3 v.is_zero_vector()
```

Out[16]: False

```
In [17]: 1 # Check if vector is not zero vector
         2 v = VC(0, 0, 0)
         3 bool(v)
```

Out[17]: False

```
In [18]: 1 # Check if vector is not zero vector
         2 v = VC(0, 1e-14, 0)
         3 bool(v)
```

Out[18]: True

```
In [19]: 1 # Create a vector from the sum of vectors
         2 VC.sum_of_vectors([ ])
```

Out[19]: VC(a=0, b=0, c=0)

```
In [27]: 1 # Create a vector from the sum of vectors
         2 vectors = [ VC(-1, 2, 3), VC(-2, -2, 2), VC(4, 0, 5) ]
         3 VC.sum_of_vectors(vectors)
```

Out[27]: VC(a=1, b=0, c=10)

```
In [28]: 1 # Create a vector from the sum of vectors
         2 vectors = [ VC(-1, 2, 3), VC(-2, -2, 2), VC(4, 0, 5) ]
         3 VC.sum_of_vectors(v for v in vectors)
```

Out[28]: VC(a=1, b=0, c=10)

```
In [22]: 1 # Create a vector from the sum of vectors and scalars
        2 VC.sum_of_vectors([ VC(-1, 2, 3), 100, VC(-2, -2, 2), 8000 ])
```

Out[22]: VC(a=8097, b=8100, c=8105)

```
In [20]: 1 # Create a vector from the product of vectors
        2 VC.prod_of_vectors([ ])
```

Out[20]: VC(a=1, b=1, c=1)

```
In [21]: 1 # Create a vector from the product of vectors
        2 vectors = [ VC(-1, 2, 3), VC(-2, -2, 2), VC(4, 0, 5) ]
        3 VC.prod_of_vectors(vectors)
```

Out[21]: VC(a=8, b=0, c=30)

```
In [22]: 1 # Create a vector from the product of vectors
        2 vectors = [ VC(-1, 2, 3), VC(-2, -2, 2), VC(4, 0, 5) ]
        3 VC.prod_of_vectors(v for v in vectors)
```

Out[22]: VC(a=8, b=0, c=30)

```
In [23]: 1 # Create a vector from the product of vectors and scalars
        2 VC.prod_of_vectors([ VC(-1, 2, 3), -1/2, VC(-2, -2, 2), 10 ])
```

Out[23]: VC(a=-10.0, b=20.0, c=-30.0)

```
In [29]: 1 # Create vectors by applying the math methods floor, ceil and trunc to vector components
        2 from math import floor, ceil, trunc
        3 v = VC(-2.8, 3.3, 5.9)
        4 ceil(v), floor(v), trunc(v)
```

Out[29]: (VC(a=-2, b=4, c=6), VC(a=-3, b=3, c=5), VC(a=-2, b=3, c=5))

```
In [ ]: 1
```