Using a Simple Vector Class

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

In [1]:
```
from skvectors import create_class_Simple_Vector
```

In [2]:
```
# Create a 3-dimensional simple vector class

# The first argument is a string with the name of the class
# to be created.

# The number of elements in the iterable given as the second
# argument determines the number of dimensions for the class.

SVC = create_class_Simple_Vector('VC', 'IJK')

# Explicit alternative:
# SVC = \
#     create_class_Simple_Vector(
#         name = 'SVC',
#         component_names = [ 'I', 'J', 'K' ],
#         brackets = [ '<', '>' ],
#         sep = ', '
#     )
```

In [3]:
```
# Apply abs to the I-component
v = SVC(-2, 3, -4)
v.c_abs_I()
```

Out[3]:
```
VC(I=2, J=3, K=-4)
```

In [4]:
```
# Apply unary minus to the K-component
v = SVC(2, 3, 4)
v.c_neg_K()
```

Out[4]: VC(I=2, J=3, K=-4)

In [5]:
```
# Apply unary minus to all components except the K-component
v = SVC(2, 3, 4)
v.c_neg_bar_K()
```

Out[5]: VC(I=-2, J=-3, K=4)

In [6]:
```
# Apply unary plus to the J-component and the K-component
v = SVC(2, 3, 4)
v.c_pos_J_K()
```

Out[6]: VC(I=2, J=3, K=4)

In [7]:
```
# Add 100 to the K-component
v = SVC(2, 3, 4)
v.c_add_K(100)
```

Out[7]: VC(I=2, J=3, K=104)

In [8]:
```
# Add 100 in-place to the K-component
v = SVC(2, 3, 4)
v.c_iadd_K(100)
v
```

Out[8]: VC(I=2, J=3, K=104)

In [9]:
```
# Subtract 3 from the J-component
v = SVC(2, 3, 4)
v.c_sub_J(3)
```

Out[9]: VC(I=2, J=0, K=4)

In [10]:
```
# Subtract 3 in-place from the J-component
v = SVC(2, 3, 4)
v.c_isub_J(3)
v
```

Out[10]: VC(I=2, J=0, K=4)

In [11]:
```
# Multiply all components except none by 8
v = SVC(2, 3, 4)
v.c_mul_bar(8)
```

Out[11]: VC(I=16, J=24, K=32)

In [12]:
```
# Multiply in-place all components except none by 8
v = SVC(2, 3, 4)
v.c_imul_bar(8)
v
```

Out[12]: VC(I=16, J=24, K=32)

In [13]:
```
# Raise the I-component to the power of 10
v = SVC(2, 3, 4)
v.c_pow_I(10)
```

Out[13]: VC(I=1024, J=3, K=4)

In [14]:
```
# Raise in-place the I-component to the power of 10
v = SVC(2, 3, 4)
v.c_ipow_I(10)
v
```

Out[14]: VC(I=1024, J=3, K=4)

In [15]:
```
# True divide none of the components by 0
v = SVC(2, 3, 4)
v.c_truediv(0)
```

Out[15]: VC(I=2, J=3, K=4)

In [16]:
```
# True divide in-place all of the components by 10
v = SVC(2, 3, 4)
v.c_itruediv_bar(10)
v
```

Out[16]: VC(I=0.2, J=0.3, K=0.4)

In [17]:
```
# Floor divide of all of the components by 2
v = SVC(2, 3, 4)
v.c_floordiv_I_J_K(2)
```

Out[17]: VC(I=1, J=1, K=2)

In [18]:
```
# Floor divide in-place all of the components by 2
v = SVC(2, 3, 4)
v.c_ifloordiv_I_J_K(2)
v
```

Out[18]: VC(I=1, J=1, K=2)

In [19]:
```
# Modulus of all of the components by 2
v = SVC(2, 3, 4)
v.c_mod_I_J_K(2)
```

Out[19]: VC(I=0, J=1, K=0)

In [20]:
```
# Modulus in-place of all of the components by 2
v = SVC(2, 3, 4)
v.c_imod_I_J_K(2)
v
```

Out[20]: VC(I=0, J=1, K=0)

In [21]:
```
# Multiply the K-component by 100
v = SVC(2, 4, 6)
v.c_mul_K(100)
```

Out[21]: VC(I=2, J=4, K=600)

In [22]:
```
# Multiply in-place the K-component by 100
v = SVC(2, 4, 6)
v.c_imul_K(100)
v
```

Out[22]: VC(I=2, J=4, K=600)

In [23]:
```
# Apply several operations to the components
v = SVC(2, 3, 4)
f = v.c_mul_K
f(10).c_add_bar(88).c_mul_I_J(88).c_sub_bar_J_K(100000).c_neg_K()
```

Out[23]: VC(I=-92080, J=8008, K=-128)

In [24]:
```
# Round components to 3 decimals
v = SVC(2.22222, 4.444444, 6.6666666)
round(v, ndigits=3)
```

Out[24]: VC(I=2.222, J=4.444, K=6.667)

In [25]:
```
# Round components to integer value
v = SVC(2.22222, 4.444444, 6.6666666)
round(v)
```

Out[25]: VC(I=2.0, J=4.0, K=7.0)

In [26]:
```
# Round component values
v = SVC(-55555555.5, -33333333.3, 55555555.5)
round(v, -4)
```

Out[26]: VC(I=-55560000.0, J=-33330000.0, K=55560000.0)

In [27]:
```
# Apply unary minus to vector
v = SVC(-3, 4, 5)
-v
```

Out[27]: VC(I=3, J=-4, K=-5)

In [28]:
```
# Apply unary plus to vector
v = SVC(-3, 4, 5)
+v
```

Out[28]: VC(I=-3, J=4, K=5)

In [29]:
```
# Addition of vectors
v = SVC(-3, 4, 5)
v + SVC(1, 1, -1)
```

Out[29]: VC(I=-2, J=5, K=4)

In [30]:
```
# In-place addition of vectors
v = SVC(-3, 4, 5)
v += SVC(1, 1, -1)
v
```

Out[30]: VC(I=-2, J=5, K=4)

In [31]:
```
# Subtraction of vectors
v = SVC(-3, 4, 5)
v - SVC(1, 1, -1)
```

Out[31]: VC(I=-4, J=3, K=6)

In [32]:
```
# In-place subtraction of vectors
v = SVC(-3, 4, 5)
v -= SVC(1, 1, -1)
v
```

Out[32]: VC(I=-4, J=3, K=6)

In [33]:
```
# Multiplication of vectors
v = SVC(-1, 2, 3)
v * SVC(2, 0, -2)
```

Out[33]: VC(I=-2, J=0, K=-6)

In [34]:
```
# In-place multiplication of vectors
v = SVC(-1, 2, 3)
v *= SVC(2, 0, -2)
v
```

Out[34]: VC(I=-2, J=0, K=-6)

In [35]:
```
# Multiplication of vector and scalar
v = SVC(-1, 2, 3)
2 * v, v * 2
```

Out[35]: (VC(I=-2, J=4, K=6), VC(I=-2, J=4, K=6))

In [36]:
```
# In-place multiplication of vector and scalar
v = SVC(-1, 2, 3)
v *= 2
v
```

Out[36]: VC(I=-2, J=4, K=6)

In [37]:
```
# True division of vectors
v = SVC(-3, 4, 6)
v / SVC(2, -2, 2)
```

Out[37]: VC(I=-1.5, J=-2.0, K=3.0)

In [38]:
```
# In-place true division of vectors
v = SVC(-3, 4, 6)
v /= SVC(2, -2, 2)
v
```

Out[38]: VC(I=-1.5, J=-2.0, K=3.0)

In [39]:
```
# True division of vector and scalar
v = SVC(-3, 4, 6)
v / 6
```

Out[39]: VC(I=-0.5, J=0.6666666666666666, K=1.0)

In [40]:
```
# In-place true division of vector and scalar
v = SVC(-3, 4, 6)
v /= 2
v
```

Out[40]: VC(I=-1.5, J=2.0, K=3.0)

In [41]:
```
# Vector to the power of vector
v = SVC(-3, 4, 6)
v**SVC(2, -2, 2)
```

Out[41]: VC(I=9, J=0.0625, K=36)

In [42]:
```
# In-place vector to the power of vector
v = SVC(-3, 4, 6)
v **= SVC(2, -2, 2)
v
```

Out[42]: VC(I=9, J=0.0625, K=36)

In [43]:
```
# Vector to the power of scalar
v = SVC(-3, 5, 6)
v**2
```

Out[43]: VC(I=9, J=25, K=36)

In [44]:
```
# In-place vector to the power of scalar
v = SVC(-3, 5, 6)
v **= 2
v
```

Out[44]: VC(I=9, J=25, K=36)

In [45]:
```
# Floor division of vectors
v = SVC(-3, 5, 6)
v // SVC(2, -2, 2)
```

Out[45]: VC(I=-2, J=-3, K=3)

In [46]:
```
# In-place floor division of vectors
v = SVC(-3, 5, 6)
v //= SVC(2, -2, 2)
v
```

Out[46]: VC(I=-2, J=-3, K=3)

In [47]:
```
# Floor division of vector and scalar
v = SVC(-3, 5, 6)
v // 2
```

Out[47]: VC(I=-2, J=2, K=3)

In [48]:
```
# In-place floor division of vector and scalar
v = SVC(-3, 5, 6)
v //= 2
v
```

Out[48]: VC(I=-2, J=2, K=3)

In [49]:
```
# Vector modulus vector
u = SVC(-3, 5, 6)
w = SVC(2, -2, 2)
u % w
```

Out[49]: VC(I=1, J=-1, K=0)

In [50]:
```
# In-place vector modulus vector
v = SVC(-3, 5, 6)
w = SVC(2, -2, 2)
v %= w
v
```

Out[50]: VC(I=1, J=-1, K=0)

In [51]:
```
# Modulus of vector and scalar
v = SVC(-3, 5, 6)
v % 2
```

Out[51]: VC(I=1, J=1, K=0)

In [52]:
```
# In-place modulus of vector and scalar
v = SVC(-3, 5, 6)
v %= 2
v
```

Out[52]: VC(I=1, J=1, K=0)

In [ ]:
```

```