向量 lab

在此 notebook 中，你将学习如何绘制二维向量和进行某些向量计算。

具体来说：

1. 绘制二维向量
2. 将二维向量与标量相乘并绘制结果
3. 将两个向量相加并绘制结果

对于此 lab，我们将使用 python 软件包 NumPy 创建向量并计算向量运算。对于绘图部分，我们将使用 python 软件包 Matplotlib。

绘制二维向量

对于此部分，我们将绘制定义如下的向量 $\vec{v}$。

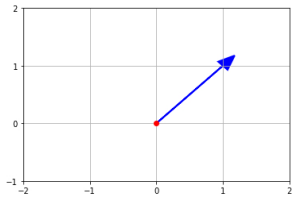
$$\vec{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

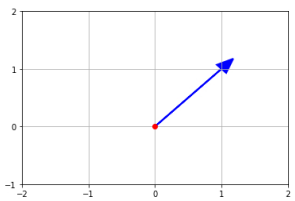
下面简要介绍了绘制向量 $\vec{v}$ 的 Python 代码所包含的部分。

1. 使用 *import* 方法使 NumPy 和 Matlibplot python 软件包可用。
2. 定义向量 $\vec{v}$
3. 使用 Matlibplot 绘制向量 $\vec{v}$
   A. 创建一个变量 ***ax*** 来表示图的坐标轴
   B. 使用 ***ax*** 和 *plot* 方法绘出原点 0,0（红点）
   C. 使用 ***ax*** 和 *arrow* 方法绘出向量 $\vec{v}$：蓝色箭头，起点为 0,0
   D. 确定 x 轴的格式
      a. 使用 *xlim* 方法设置范围
      b. 使用 ***ax*** 和 *set_xticks* 方法设置主要刻度线
   E. 确定 y 轴的格式
      a. 使用 *ylim* 方法设置范围
      b. 使用 ***ax*** 和 *set_yticks* 方法设置主要刻度线
   F. 使用 *grid* 方法创建网格线
   G. 使用 *show* 方法显示图

In [19]:

```
#导入NumPy模块和Matplotlib模块
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

# 定义向量v
v = np.array([1,1])

# 使用Matplotlib将向量v绘制出来 蓝色的箭头 红色的原点
# 创建轴线ax
ax = plt.axes()

# 绘制一个红色的点 在0,0处
ax.plot(0,0,'or')

# 绘制向量v 蓝色的箭头 起点为0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.0, head_width=0.20, head_length=0.25)

# 设置x轴的界限 即-2到2 即X轴整体的长度 而不是里面的刻度 整体界限是-2到2 然后你可以调整里面的刻度
plt.xlim(-2,2)

# 设置x轴的刻度 生成-2到3（左闭右开）的X轴刻度
major_xticks = np.arange(-2, 3)
ax.set_xticks(major_xticks)


# 设置y 轴的界限 -1到2
plt.ylim(-1, 2)

# 设置y 轴的刻度 生成-1到3（左闭右开）的y轴刻度
major_yticks = np.arange(-1, 3)
ax.set_yticks(major_yticks)

# 设置网格线 b代表布尔值/None值 是否有放个 True即有 False即无
plt.grid(b=True, which='major')

# 显示图象
plt.show()
```





使用标量伸缩向量

对于此部分，我们将绘制将向量 $\vec{v}$ 乘以标量 a 后的结果。标量 a 和向量 $\vec{v}$ 的定义如下。

$$a = 3$$

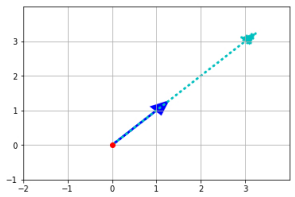
$$\vec{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

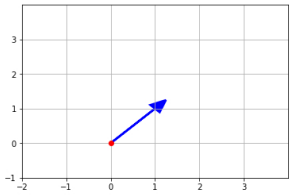
TODO：将向量与标量相乘并绘制结果

对于此部分，你将创建向量 $\vec{av}$，并在图中用青色虚线表示该向量。

1. 在以下代码中使向量 $\vec{v}$ 与标量 a 相乘（请参阅 ***TODO 1.***）。
2. 在以下代码中使用 *ax.arrow(...)* 语句将向量 $\vec{av}$ 添加到图中（请参阅 ***"TODO 2."***）。在 *ax.arrow(...)* 语句中添加 *linestyle = 'dotted'* 并将 *color* 设为 *'c'*，使向量 $\vec{av}$ 变成青色虚线向量。

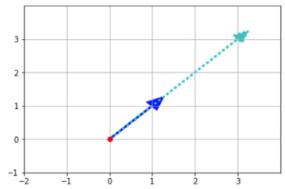
In [9]:
```
# 定义向量v
v = np.array([1,1])

# 定义标量a
a = 3

# TODO 1.: Define vector av - as vector v multiplied by scalar a
av = a * v

# Plots vector v as blue arrow with red dot at origin (0,0) using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# TODO 2.: Plot vector av as dotted (linestyle='dotted') vector of cyan color (color='c')
# using ax.arrow() statement above as template for the plot
'''
起点坐标x,y *向量array 线宽 箭头宽 箭头长 颜色 线形状（默认为实线 可以改成虚线）
'''
ax.arrow(0, 0, *av, color='c',linewidth=2.5, head_width = 0.30, head_length = 0.35, linestyle = 'dotted')


# Sets limit for plot for x-axis
plt.xlim(-2, 4)

# Set major ticks for x-axis
major_xticks = np.arange(-2, 4)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```





伸缩向量的解决方案

以上代码的输出应该与下面的输出相符。



伸缩向量的解决方案视频

你可以在**向量 lab解决方案**部分找到解决方案视频。建议打开另一个浏览器窗口，以便在向量 lab Jupyter Notebook 和此 lab的解决方案视频之间轻松切换。

将两个向量相加

在此部分，我们将绘制将向量 $\vec{w}$ 加到向量 $\vec{v}$ 上的结果。向量 $\vec{v}$ 和 $\vec{w}$ 定义如下。

$$\vec{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

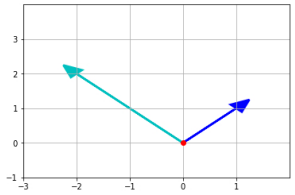
$$\vec{w} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

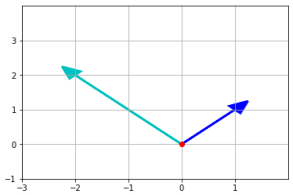
绘制两个向量

以下是显示从原点 (0,0) 开始的向量 $\vec{v}$ 和 $\vec{w}$ 的代码及图。

In [20]:
```
# Define vector v
v = np.array([1,1])

# Define vector w
w = np.array([-2,2])

# Plots vector v(blue arrow) and vector w(cyan arrow) with red dot at origin (0,0)
# using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# Plots vector w as cyan arrow starting at origin 0,0
ax.arrow(0, 0, *w, color='c', linewidth=2.5, head_width=0.30, head_length=0.35)

# Sets limit for plot for x-axis
plt.xlim(-3, 2)

# Set major ticks for x-axis
```

```
major_xticks = np.arange(-3, 2)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```





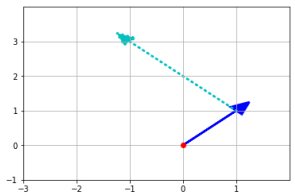
向量相加

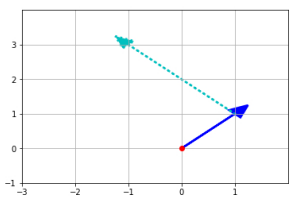
我们在下图中显示了将向量 $\vec{w}$ 加到向量 $\vec{v}$ □上的过程。

绘制向量相加过程

以下是将向量 $\vec{w}$ 加到向量 $\vec{v}$ 上的代码和图。注意，当我们将向量 $\vec{w}$ 加到向量 $\vec{v}$ 上时，向量 $\vec{w}$ 的原点现在变成 (1,1)。此外，我们在 *ax.arrow(...)* 语句中添加了 *linestyle = 'dotted'* 和 *color = 'c'*，使向量 $\vec{w}$ 变成青色虚线向量。

In [21]:
```
# Define vector v
v = np.array([1,1])

# Define vector w
w = np.array([-2,2])

# Plot that graphically shows vector w(dotted cyan arrow) added to vector v(blue arrow)
# using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# Plots vector w as cyan arrow with origin defined by vector v
ax.arrow(v[0], v[1], *w, linestyle='dotted', color='c', linewidth=2.5,
         head_width=0.30, head_length=0.35)

# Sets limit for plot for x-axis
plt.xlim(-3, 2)

# Set major ticks for x-axis
major_xticks = np.arange(-3, 2)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```





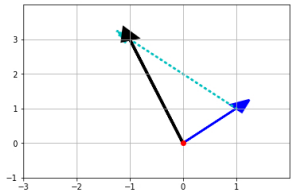
TODO：将两个向量相加并绘制结果

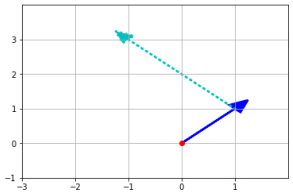
在此部分，你将创建向量 $\vec{vw}$，然后在图中表示成更粗的黑色向量。

1. 在以下代码中通过将向量 $\vec{w}$ 加到向量 $\vec{v}$ 上创建向量 $\vec{vw}$（请参阅 ***TODO 1.:***）。
2. 在以下代码中使用 *ax.arrow(...)* 语句将向量 $\vec{vw}$ 添加到图中（请参阅 ***TODO 2.:***）。在 *ax.arrow(...)* 语句中设置 *linewidth = 3.5* 和 *color = 'k'*，使向量 $\vec{vw}$ 变成更粗的黑色向量。

In [28]:
```
# Define vector v
v = np.array([1,1])
```

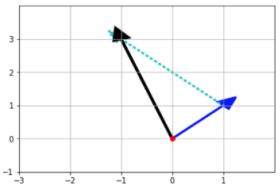
```
# Define vector w
w = np.array([-2,2])

# TODO 1.: Define vector vw by adding vectors v and w
vw = v+w

# Plot that graphically shows vector vw (color='b') - which is the result of
# adding vector w(dotted cyan arrow) to vector v(blue arrow) using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# Plots vector w as cyan arrow with origin defined by vector v
ax.arrow(v[0], v[1], *w, linestyle='dotted', color='c', linewidth=2.5,
         head_width=0.30, head_length=0.35)

# TODO 2.: Plot vector vw as black arrow (color='k') with 3.5 linewidth (linewidth=3.5)
# starting vector v's origin (0,0)
ax.arrow(0, 0, *vw, color='k', linewidth=3.5,
         head_width=0.30, head_length=0.35)

# Sets limit for plot for x-axis
plt.xlim(-3, 2)

# Set major ticks for x-axis
major_xticks = np.arange(-3, 2)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```





将两个向量相加的解决方案

以上代码的输出应该与下面的输出相符。如果你需要帮助或者想要检查你的答案是否正确，请点击此处查看解决方案 notebook。



将两个向量相加的解决方案视频

你可以在**向量 lab解决方案**部分找到解决方案视频。建议打开另一个浏览器窗口，以便在向量 lab Jupyter Notebook 和此 lab 的解决方案视频之间轻松切换。