In [1]:
```
doc_trump = "Mr. Trump became president after winning the political election. Thoug



doc_election = "President Trump says Putin had no political interference is the ele



doc_putin = "Post elections, Vladimir Putin became President of Russia. President P



documents = [doc_trump, doc_election, doc_putin]
```

In [5]:
```
from sklearn.feature_extraction.text import CountVectorizer

import pandas as pd

count_vect = CountVectorizer(stop_words='english')

count_vect = CountVectorizer()

sparse_matrix = count_vect.fit_transform(documents)
doc_term_matrix = sparse_matrix.todense()

df = pd.DataFrame(doc_term_matrix,

                  columns=count_vect.get_feature_names_out(),

                  index=['doc_trump', 'doc_election', 'doc_putin'])

df
```

Out[5]:

	after	as	became	by	career	claimed	do	earlier	election	elections	...
doc_trump	1	0	1	0	0	0	0	0	1	0	...
doc_election	0	0	0	1	0	1	1	0	2	0	...
doc_putin	0	1	1	0	1	0	0	1	0	1	...

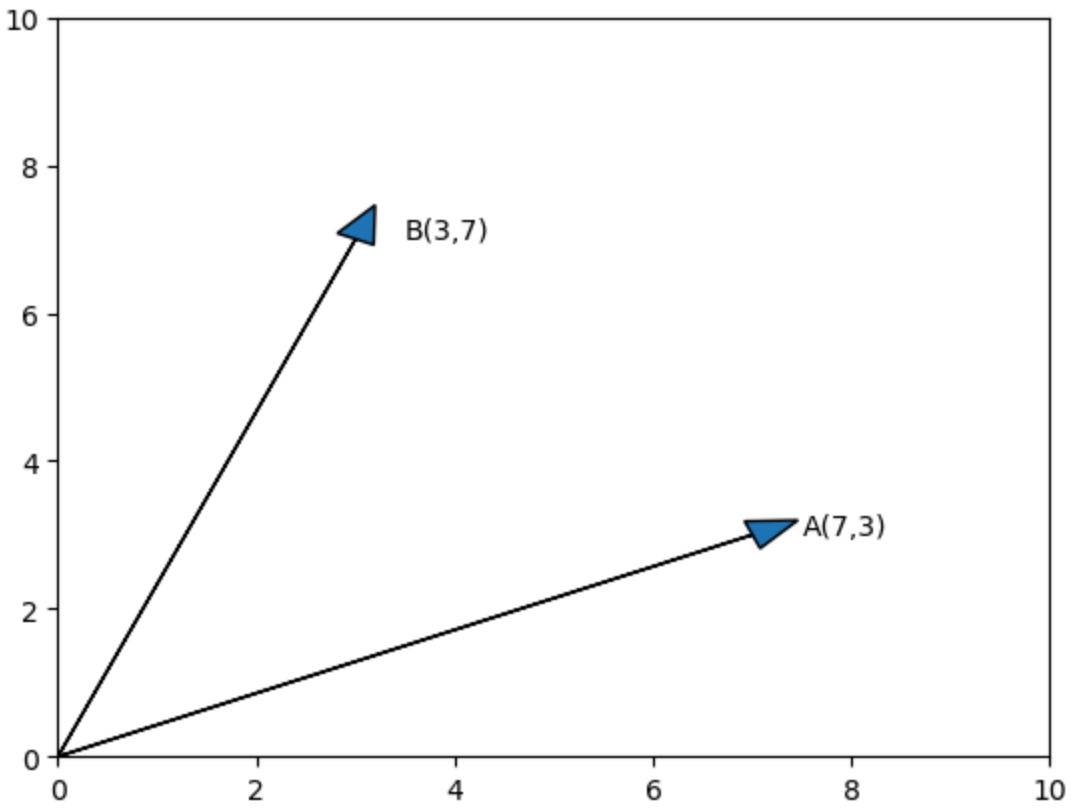
3 rows × 48 columns

In [7]:
```
from sklearn.metrics.pairwise import  cosine_similarity
print(cosine_similarity(df,df))
```

```
[[1.         0.51480485 0.38890873]
 [0.51480485 1.         0.38829014]
 [0.38890873 0.38829014 1.        ]]
```

In [15]:
```
import numpy as np

import matplotlib.pyplot as plt
```

```
A = np.array([7,3])

B = np.array([3,7])

ax = plt.axes()

ax.arrow(0.0, 0.0, A[0], A[1], head_width=0.4, head_length=0.5)

plt.annotate(f"A({A[0]},{A[1]})", xy=(A[0], A[1]),xytext=(A[0]+0.5, A[1]))

ax.arrow(0.0, 0.0, B[0], B[1], head_width=0.4, head_length=0.5)

plt.annotate(f"B({B[0]},{B[1]})", xy=(B[0], B[1]),xytext=(B[0]+0.5, B[1]))

plt.xlim(0,10)

plt.ylim(0,10)

plt.show()

plt.close()
```



In [13]:
```
cos_sim=np.dot(A,B)/(np.linalg.norm(A)*np.linalg.norm(B))

print (f"Cosine Similarity between A and B:{cos_sim}")

print (f"Cosine Distance between A and B:{1-cos_sim}")
```

```
Cosine Similarity between A and B:0.7241379310344827
Cosine Distance between A and B:0.27586206896551735
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: