

TrabalhoA1

September 14, 2018

```
In [2]: def convertDecimalBinary(decimal):
        binary=[]
        integerDecimal=int(decimal)
        fractionalDecimal=float(decimal)-int(decimal)
        integerBinary=[]
        fractionalBinary=[]
        counter=10

        while(integerDecimal//2>=1):
            integerBinary.append(integerDecimal%2)
            integerDecimal//=2
        if integerDecimal!=0: integerBinary.append(1)
        else: integerBinary.append(0)
        integerBinary.reverse()
        if (fractionalDecimal==0): fractionalBinary.append(0)

        while (counter>0 and fractionalDecimal!=0):
            fractionalDecimal*=2
            fractionalBinary.append(int(fractionalDecimal))
            fractionalDecimal=float(fractionalDecimal) - int(fractionalDecimal)
            counter-=1
        binary=integerBinary + [","] + fractionalBinary
        s1 = [str(i) for i in binary]
        s1=''.join(s1)
        return s1
        binary=convertDecimalBinary(65.25)
        print(binary)
```

1000001,01

```
In [3]: import numpy
        """declara uma array 1D com primeiro elemento igual a 100
        e último igual a 199, incrementando de 1 em 1"""
        arr = numpy.arange(100, 200)
        """declara uma lista de índices"""
        select = [5, 25, 50, 75, -5]
        """printa os elementos correspondentes
```

```

da lista de índices"""
print(arr[select])

"""declara uma array 1D com primeiro elemento igual
a 10 e último igual a 19, incrementando de 1 em 1"""
arr = numpy.arange(10, 20)
"""declara uma lista de variáveis booleanas com cada posição
representando a divisibilidade por 3 do mesmo index da array anterior"""
div_by_3 = arr%3 == 0
print(div_by_3)
"""printa somente os elementos com boolean true
para a divisibilidade por 3"""
print(arr[div_by_3])

"""cria uma array 2D, a partir de uma array 1D de 10 elementos,
com 2 linhas e 5 colunas"""
arr = numpy.arange(10, 20).reshape((2,5))
print(arr)

```

```

[105 125 150 175 195]
[False False  True False False  True False False  True False]
[12 15 18]
[[10 11 12 13 14]
 [15 16 17 18 19]]

```

```

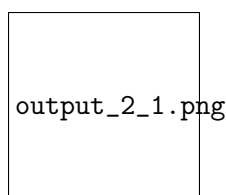
In [5]: import matplotlib.pyplot as plt
import numpy as np
N = 10
y = np.linspace(0, 20, N, endpoint=False)
x1 = np.linspace(0, 5, N, endpoint=False)
x2 = np.linspace(0, 5, N, endpoint=False)
print(x1)
print(x2)
print(y)
plt.plot(x1, y)
plt.plot(x2, y + 1)
plt.show()

```

```

[0.  0.5 1.  1.5 2.  2.5 3.  3.5 4.  4.5]
[0.  0.5 1.  1.5 2.  2.5 3.  3.5 4.  4.5]
[ 0.  2.  4.  6.  8. 10. 12. 14. 16. 18.]

```



```
In [6]: import matplotlib.pyplot as plt
import numpy as np
N = 10
y = np.linspace(0, 20, N, endpoint=False)
x1 = np.linspace(5, 0, N, endpoint=False)
x2 = np.linspace(0, 5, N, endpoint=False)
print(x1)
print(x2)
print(y)
plt.plot(x1, y)
plt.plot(x2, y + 1)
plt.show()
```

```
[5.  4.5 4.  3.5 3.  2.5 2.  1.5 1.  0.5]
[0.  0.5 1.  1.5 2.  2.5 3.  3.5 4.  4.5]
[ 0.  2.  4.  6.  8. 10. 12. 14. 16. 18.]
```

