

Atividade01_Final

September 14, 2018

```
In [16]: import numpy as numpy
import matplotlib.pyplot as plt

#-----

arr = numpy.arange(100, 200)
select = [5, 25, 50, 75, -5]

#irá gerar valores aleatórios ente 100 e 200 com especificações do select
print(arr[select])
[105, 125, 150, 175, 195]
arr = numpy.arange(10, 20) #numeros aleatórios de 10 a 20
print(arr)

#se o resto de cada elemento do arr, dividido por 3, da resto zero
div_by_3 = arr%3 == 0
print(div_by_3)
[ False,  False,  True,  False,  False,  True,  False,  False,  True,  False]

#Os valores booleanos são usados como indices, imprimindo apenas os valores verdadeiros
print(arr[div_by_3])
[12, 15, 18]
arr = numpy.arange(10, 20) . reshape((2,5))
[[10, 11, 12, 13, 14],
 [15, 16, 17, 18, 19]]

#-----

def converter(n): #função para converter para binário
    inteira = int(n) #apenas a parte inteira do numero
    frac = float(n) - int(n) #apenas a parte fracionada
    binarioint = []
    binariofrac = []

    while(inteira != 0 ): #enquanto a parte inteira for diferente de 0
        a = inteira % 2
        binarioint.append(a) #adiciona na array o resto da divisão
        inteira = int(inteira/2) #novo valor de "inteira" é a divisão inteira do valor
```

```

binarioint.reverse()

while(frac != 0.0 ): #enquanto a parte fracionada for diferente de 0.0
    a = frac * 2 #multiplicação da parte fracionada por dois
    b = int(a) #b é apenas a parte inteira da multiplicação
    c = float(a) - int(a) #c é apenas a parte fracionada da multiplicação
    binariofrac.append(b) #a parte inteira(b) é armazenada na array
    frac = c #a parte fracionada agora é c

conver_fin = binarioint + ['.'] + binariofrac #concatena os dois array
conver_fin = [str(i) for i in conver_fin]
conver_fin = ''.join(conver_fin) #todos os elementos são postos juntos
return(conver_fin)

print("Coversão do número para binário: ",converter(45.5))

#-----

#comandos para a criação das retas
plt.plot([1,2,3,4]) #reta azul, com as coordenadas de x
plt.plot([2,3,4,5]) #reta laranja, com as coordenadas de x

plt.show()

plt.plot([5,6,7,8]) #reta azul, com as coordenadas de x
plt.plot([8,7,6,5]) #reta laranja, com as coordenadas de x

plt.show()

[105 125 150 175 195]
[10 11 12 13 14 15 16 17 18 19]
[False False  True False False  True False False  True False]
[12 15 18]
Coversão do número para binário:  101101.1

```

