

Capítulo 6 - Solução

November 20, 2020

```
[1]: # Importando os módulos necessários
import ipywidgets as widget
from ipywidgets import interact, interact_manual
from sympy import symbols, plot
```

1 Atividade 4: Para casa

1.1 Exercício 1: Parábola

Uma parábola tem a sua função dada por:

$$f(x) = ax^2 + bx + c$$

De forma similar à tarefa da equação da reta crie um programa capaz de criar um gráfico de uma equação do segundo grau. Considere as seguintes premissas:

- Os valores de a , b e c devem ser entradas do programa na forma de Sliders;
- Os sliders devem ter os seguintes parâmetros:

Slider	Min	Max	Passo	Val. Inicial
a	-2.0	2.0	0.05	0.25
b	-20.0	20.0	0.5	0.0
c	-45.0	45.0	0.5	0.0

- O título do gráfico deve mostrar a equação da parábola (Considere mostrar a equação corretamente, considerando as situações que a , b e/ou c podem ser zero ou que a e/ou b podem ser 1);
- O usuário poderá alterar a cor do gráfico através de um Widget;
- Os limites do gráfico devem ser $x \in [-100, 100]$ e $f(x) \in [-200, 200]$;
- O usuário pode mostrar ou ocultar o título;
- O usuário pode escolher o algoritmo de plotagem (Uso do algoritmo adaptativo);
- Caso o usuário não escolha o algoritmo adaptativo, o mesmo poderá escolher a quantidade de pontos através de um `FloatLogSlider`:

Características	Valor
base	2
min	64 pontos

Características	Valor
max	4096 pontos

- Você deve utilizar x como variável simbólica;
- O gráfico deve ser plotado com a função `plot()` do pacote `sympy`.

Solução:

```
[2]: ptWidget=widget.FloatLogSlider(
    description='No. Pontos',
    min=6, #Expoente mínimo
    max=12, #Expoente máximo
    value=2**4, #Valor inicial
    base=2, #Base (O número dobra a cada passo)
    step=1, #Passo
    readout_format='d' #Mostrando a saída na forma de inteiro
)
```

```
[3]: @interact(
    a=widget.FloatSlider(min=-2.0,max=2,step=0.05,value=0.25),
    b=(-20.0,20.0,0.5),
    c=(-50.0,50.0,0.5),
    alg=widget.Checkbox(description='Alg. Adaptativo',value=False),
    #Vamos utilizar o FloatLogSlider
    pts=ptWidget,
    cor=widget.ColorPicker(description='Cor:',value='Blue'),
    Título=True
)
def plotando_parabola(a,b,c,alg,pts,Título,cor):
    x = symbols('x')
    y = a*x**2+b*x+c
    ptWidget.disabled=alg # Se o método adaptativo estiver ativado desativa o
    ↪widget

    # Mostrando ou ocultando o título
    t=''
    if Título:
        t += '$y(x)='

    # PARCELA 'a'
    if a!=1 and a!=0:
        t+= '{:.2g}'.format(a)

    # PARCELA 'x^2'
    if a!=0:
        t+= 'x^2'
```

```

# PARCELA 'b'
if a!=0:
    if b!=1 and b!=0:
        t+= '{:+.2g}'.format(b)

    # PARCELA 'x'
    if b!=0:
        t+= 'x'
else:
    if b!=1 and b!=0:
        t+= '{:.2g}'.format(b)

    # PARCELA 'x'
    if b!=0:
        t+= 'x'

# PARCELA 'b'
if c==0:
    if a==0 and b==0:
        t+= '0'
elif a!=0 or b!=0:
    t+= '{:+.2g}'.format(c)
else:
    t+= '{:.2g}'.format(c)

# Finalizando
t += '$'

plot(y,(x,-500,500),
      xlim=(-100,100),
      ylim=(-200,200),
      adaptive=alg,
      title=t,
      nb_of_points=int(pts),
      line_color=cor
)

```

```

interactive(children=(FloatSlider(value=0.25, description='a', max=2.0, min=-2.
↵0, step=0.05), FloatSlider(valu...

```

1.2 Exercício 2: Dissecando uma string

Faça um programa que tenha como entrada uma caixa de texto (o Widget Textarea). O programa deve analisar o conteúdo da caixa de texto e verificar se:

- O texto só tem espaços;
- O texto está vazio;
- O texto é numérico;

- O texto é alfabético;
- O texto é alfanumérico;
- O texto pode ser utilizado como um identificador padrão do Python;
- O texto está em maiúsculas;
- O texto está na forma de título;
- O texto está em minúsculas.

Você deve pesquisar sobre o Widget `Textarea` na documentação.

A análise do texto deve ser feita em tempo real.

```
[4]: @interact(
    str=widget.Textarea(
        value='',
        placeholder='Digite algo',
        description='Texto:',
        disabled=False
    ))
def dissecando_string(str):
    saida = '0 texto '
    count = 0
    if str.isspace():
        if count > 0:
            saida += ', '
        count += 1
        saida += 'só tem espaços'
    if str=='':
        if count > 0:
            saida += ', '
        count += 1
        saida += 'está vazio'
    if str.isnumeric():
        if count > 0:
            saida += ', '
        count += 1
        saida += 'é numérico'
    if str.isalpha():
        if count > 0:
            saida += ', '
        count += 1
        saida += 'é alfabético'
    if str.isalnum():
        if count > 0:
            saida += ', '
        count += 1
        saida += 'é alfanumérico'
    if str.isidentifier():
        if count > 0:
```

```

        saida += ', '
    count += 1
    saida += 'pode ser utilizado como identificador padrão do python'
if str.isupper():
    if count > 0:
        saida += ', '
    count += 1
    saida += 'está em caixa alta'
if str.islower():
    if count > 0:
        saida += ', '
    count += 1
    saida += 'está todo em minúsculo'
if str.istitle():
    if count > 0:
        saida += ', '
    count += 1
    saida += 'está na forma de título'
if count==0:
    saida += 'não é de nenhum padrão conhecido'
saida += '.'
print(saida)

```

```

interactive(children=(Textarea(value='', description='Texto:',
    ↪placeholder='Digite algo'), Output()), _dom_cla...

```

1.3 Exercício 3: Contando dígitos

Crie um programa capaz de contar a quantidade de dígitos de um número inteiro.

- Use um widget de campo de texto como entrada;
- O programa deve aceitar qualquer número inteiro (Positivos e negativos).

[5]: *#Existem muitas formas de resolver este problema. Segue uma delas:*

```

import math
@interact_manual(Número=widget.IntText())
def contarDigito(Número):
    Negativo=Número<0
    if Negativo:
        Número*=-1
    if(Número>0):
        Quantidade = math.floor(math.log10(Número)+1)
    else:
        Quantidade = 1
    print("Número de dígitos : {}".format(Quantidade))

```

```

interactive(children=(IntText(value=0, description='Número'),
    ↪Button(description='Run Interact', style=ButtonS...

```

1.4 Exercício 4: Divisores de um número

Crie um programa capaz de encontrar todos os divisores de um número.

- Use um widget de campo de texto como entrada;
- O programa deve aceitar somente números inteiros entre 0 e 500.

O usuário deve clicar num botão para fazer a análise do número (Use o `interact_manual`).

```
[6]: @interact_manual(Número=widget.BoundedIntText(min=0,max=500))
def contar_divisores(Número):
    Negativo=Número<0
    if Negativo:
        Número*=-1
    if(Número>0):
        divisor = 1
        print('0 número {} pode ser dividido por:'.format(Número))
        while (divisor<= Número/2):
            if (Número % divisor == 0): # Se o resto for 0 ele é divisível
                print('→ {}'.format(divisor*(-1)**int(Negativo)))
                divisor+=1 # Essa expressão tem o mesmo valor que divisor = divisor
        print('→ {}'.format(Número*(-1)**int(Negativo)))
    else:
        print("0 número 0 não tem divisores.")
```

```
interactive(children=(BoundedIntText(value=0, description='Número', max=500),
↳Button(description='Run Interact...
```

1.5 Exercício 5: Formulário

Crie um programa que capaz de fazer a inscrição de um determinado usuário. O formulário deve ter como entrada:

- Nome (Só aceita se a string for na forma de título e não for na forma alfanumérica)
- Nome do usuário (Só aceita se aceita se a string for um identificador padrão do Python)
- Data de nascimento (Só aceita se o usuário for maior de 18 anos) (Utilize o widget `DatePicker`)

Para aceitar a inscrição o usuário deve clicar num botão (Use o `interact_manual`).

Caso a inscrição for inválida o programa deve indicar a causa do erro. Caso a inscrição seja válida o programa deve imprimir:

Inscrição validada com sucesso!

Use o módulo `datetime` e a função abaixo para te ajudar a construir o seu programa:

```
from datetime import date

def calcularIdade(dataNascimento):
    """
```

```

Esta função retorna a sua idade de acordo com a data de nascimento
A função retorna 'Erro' caso a data de nascimento não for definida
"""
try:
    hoje = date.today() #Captura a data de hoje

    # Calcula a sua idade baseando-se na:

    ## Diferença dos anos
    idade = hoje.year - dataNascimento.year

    ## Diferença entre os meses de aniversário
    if (hoje.month, hoje.day) < (dataNascimento.month, dataNascimento.day):
        idade -= 1

    return idade #Retorna a sua idade como valor de saída
except AttributeError:
    return "Erro" #Em caso de erro retorna 'Erro'

```

```

[7]: from datetime import date

def calcularIdade(dataNascimento):
    """
    Esta função retorna a sua idade de acordo com a data de nascimento
    A função retorna 'Erro' caso a data de nascimento não for definida
    """
    try:
        hoje = date.today() #Captura a data de hoje

        # Calcula a sua idade baseando-se na:

        ## Diferença dos anos
        idade = hoje.year - dataNascimento.year

        ## Diferença entre os meses de aniversário
        if (hoje.month, hoje.day) < (dataNascimento.month, dataNascimento.day):
            idade -= 1

        return idade #Retorna a sua idade como valor de saída
    except AttributeError:
        return "Erro" #Em caso de erro retorna 'Erro'

@interact_manual(
    Nome = widget.Text(placeholder='Nome completo',description='Nome:'),
    Usuário = widget.Text(placeholder='Nome do seu usuário'),
    dataNascimento = widget.DatePicker(description='Data de Nascimento')
)

```

```

def inscrição(Nome,Usuário,dataNascimento):
    Validação = True
    erro = 'Inscrição inválida! Corrija os erros abaixo:\n'
    if not(Nome.istitle() and not(Nome.isalnum())):
        Validação = False
        erro += '→ Seu nome deve está na forma de título.\n'
    if not(Usuário.isidentifier()):
        Validação = False
        erro += '→ O nome do seu usuário deve ser um identificador do Python.\n'

    idade = calcularIdade(dataNascimento)

    if idade == 'Erro':
        erro += "→ Preencha a sua idade."
        Validação = False
    elif idade < 0:
        erro += '→ Coloque uma data menor que a data atual.'
    elif idade < 18:
        erro += "→ Você tem apenas {} anos. Inscrição só é válida para maiores_
↳de 18 anos.".format(idade)
        Validação = False

    if not(Validação):
        print(erro)
    else:
        print('Inscrição aceita!')

```

```

interactive(children=(Text(value='', description='Nome:', placeholder='Nome_
↳completo'), Text(value='', descrip...

```