**Unidade 3: Iniciação à programação em Python** | **Lição 1: Funções e Ciclos**

Nesta primeira lição da Unidade 3, aplicará o conhecimento de algoritmos e da linguagem Python para:

- Procurar soluções de uma equação $f(x) = 0$.
- Resolver um problema de otimização.

**Objetivos:**

- Utilizar uma função em linguagem Python.
- Implementar o ciclo **While**.

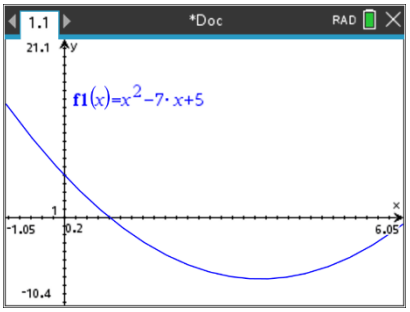
---

**Método da Bisseção**

Considere a função *f* definida em $[-2, 3]$ por $f(x) = x^2 - 7x + 5$.

Utilize a calculadora para traçar uma curva $C_f$ representação gráfica da função $f$.

Iremos resolver a equação $f(x) = 0$ com um programa em Python, que vai escrever, correspondente a um algoritmo conhecido por "método da bisseção".




**A bissecção?!**

Para se compreender melhor o que é a bisseção, considere-se uma pequena tarefa:

"***procurar uma palavra num volumoso dicionário com 1024 páginas***".

Uma estratégia possível, e com base no método da bissecção, é:

- Abrir o dicionário ao meio e a palavra não está lá, mas está antes, ou seja, está nas primeiras 512 páginas.
- Abrir, agora, a meio da 1ª metade e a palavra não está lá, mas está depois, ou seja está entre as páginas 257 e 512.
- Abrir, mais uma vez a meio, mas depois das páginas onde a palavra se encontra, e assim sucessivamente.

De cada vez que abrir o dicionário, o número de páginas que falta examinar é dividido por 2.

Assim, neste dicionário com 1024 páginas, encontrará a palavra no máximo após 10 vezes que efetuar a abertura do dicionário, pois $1024/(2^{10}) = 1$.

**ALGORÍTMO:**

Enquanto $b - a > erro$ faz:

  m $\leftarrow \frac{a+b}{2}$

  Se $f(m)$ e $f(a)$ têm sinais contrários

  então

    $b \leftarrow m$

  senão

    $a \leftarrow m$

Fim Enquanto

**OBSERVAÇÕES:**

*[a, b]* - extremos do intervalo inicial;

$f$ - função a estudar, contínua em [a, b];

$m$ - considera-se o valor central do intervalo [a, b];

Se f(m) e f(a) têm sinais contrários, então procura-se a solução de $f(x) = 0$ no intervalo [a, m]

senão no intervalo [m, b] ;

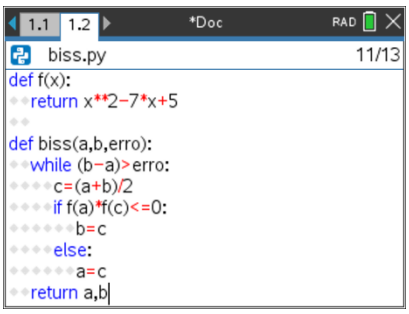
Volta a calcular-se o valor central do novo intervalo até que os extremos distem menos que o erro indicado à partida.

## IMPLEMENTAÇÃO DO ALGORITMO:

- Enquadrar entre dois números inteiros, a e b, a solução $x_0$ da equação $f(x) = 0$ com uma precisão definida (erro máximo), que designaremos de "erro".
- Verificar que $f(a) \times f(b) < 0$ .
- Calcular $f(a) \times f\left(\frac{a+b}{2}\right)$ e $f\left(\frac{a+b}{2}\right) \times f(b)$ .
- Concluir se $x_0$ pertence ao intervalo $\left[a, \frac{a+b}{2}\right]$ ou ao intervalo $\left[\frac{a+b}{2}, b\right]$.

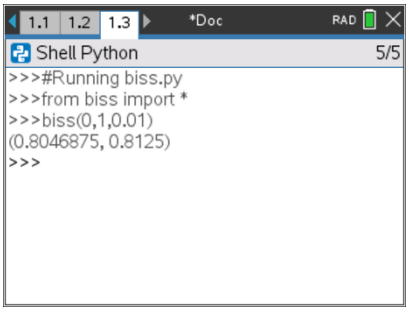
Abra uma nova página da aplicação TI-Python com o editor de programas, designe-o por BISS.

- Introduza as diferentes instruções de código no programa, pode encontrá-las no módulo **Planos integrados** (clicar na tecla menu depois **4: Planos integrados**).



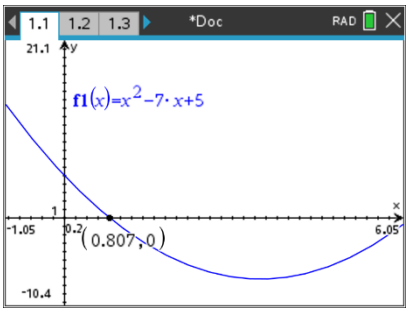

- Os operadores relacionais e lógicos podem obter-se diretamente pressionando menu depois **4: Planos integrados** e por fim **3: Ops**.
- $[a, b]$ representa o intervalo de estudo, "erro" o erro máximo admissível.

Execute o programa, clicando simultaneamente nas teclas ctrl e R, e no interpretador (Shell Python) aplique a função BISS() ao intervalo [0, 1] e com erro máximo de 0.01.




Observe o resultado obtido. Teste os resultados apresentados pelo seu programa recorrendo à resolução gráfica da equação na aplicação Gráficos da TI-Nspire CX II.

Vá refinando a procura para encontrar uma solução tão próxima quanto a obtida pela resolução gráfica.



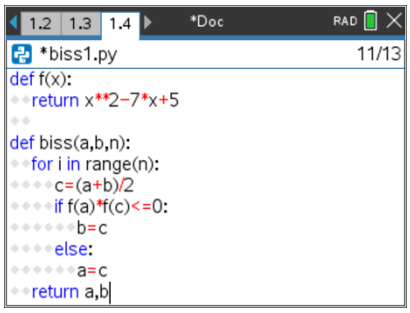

O valor exato da solução $x_0$ no intervalo [0, 1] é dado por: $x_0 = \frac{7-\sqrt{29}}{2}$.

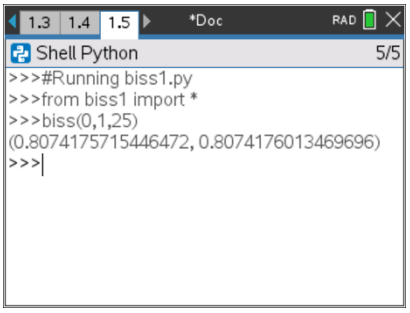
## POSSÍVEIS EXTENSÕES:

### a) Processo Iterativo

Em alternativa a procurar a solução em função de um erro dado, fazê-lo a partir de um número n de iterações.

- Comece por fazer uma cópia do programa BISS() e designe-o por BISS1, aliás nome sugerido por defeito. Para duplicar um programa, pressione a tecla menu, em seguida na opção **1: Ações** e por fim seleciona a opção **3: Criar cópia …** . Surgirá uma janela para inserir o nome.
- Altere no código do programa o ciclo WHILE pelo ciclo FOR adequado, conforme se pode observar no ecrã ao lado.
- O restante código do programa mantem-se igual. Execute o programa e função, compare o resultado obtido com o anterior.




### b) Processo Recursivo

Construa um programa em que o processo de procura do intervalo onde se encontra a solução recorra recursivamente a uma mesma função.

Observe a proposta de programa ao lado e construa o programa em TI-Python, experimente-o.

**PROGRAMA**

```
def biss(a,b,erro) :
    if (b-a)<=erro :
        return a,b
    else :
            c=(a+b)/2
            if f(a)*f(b)<=0 :
                return biss(a,c,erro)
            else :
                return biss(c,b,erro)
```

**NOTA:**

Para mais informações sobre recursividade deve consultar a Lição 3 desta Unidade 3 do projeto “10 Minutos de Código”.