

Laboratório - Crie um jogo simples com Python IDLE

Objetivos

Parte 1: Iniciar o VirtualBox e inserir a VM I2IoT Server

Parte 2: Criar um jogo simples com Python IDLE

Parte 3: IDLE for Python

Histórico

Python, uma linguagem de programação, permite declarações mais simples. O Python é muito fácil de usar, poderoso e versátil. Essa linguagem se tornou a linguagem de escolha para muitos desenvolvedores de IoT. Um dos principais motivos para a popularidade do Python é a comunidade de desenvolvedores. Os desenvolvedores do Python criaram e disponibilizaram muitos módulos específicos que podem ser importados em qualquer programa para dar ao programa, imediatamente, funcionalidades adicionais.

Cenário

Neste laboratório, você criará um jogo simples usando o Python IDLE.

Recursos necessários

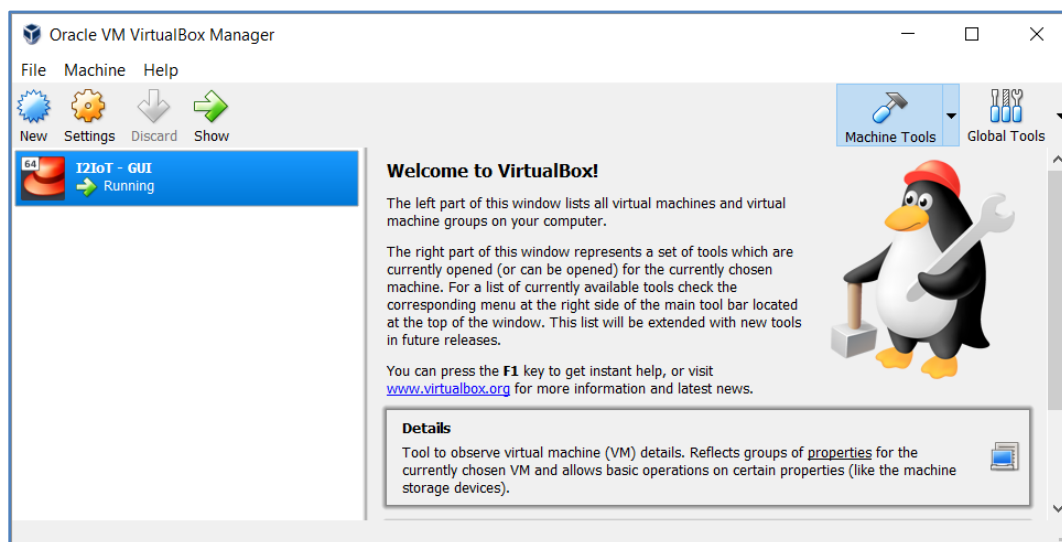
- Um computador pessoal moderno com RAM suficiente e com acesso à Internet.
- VirtualBox com I2IoT Server instalado.

Parte 1: Inicie o VirtualBox e entre na VM I2IoT Server

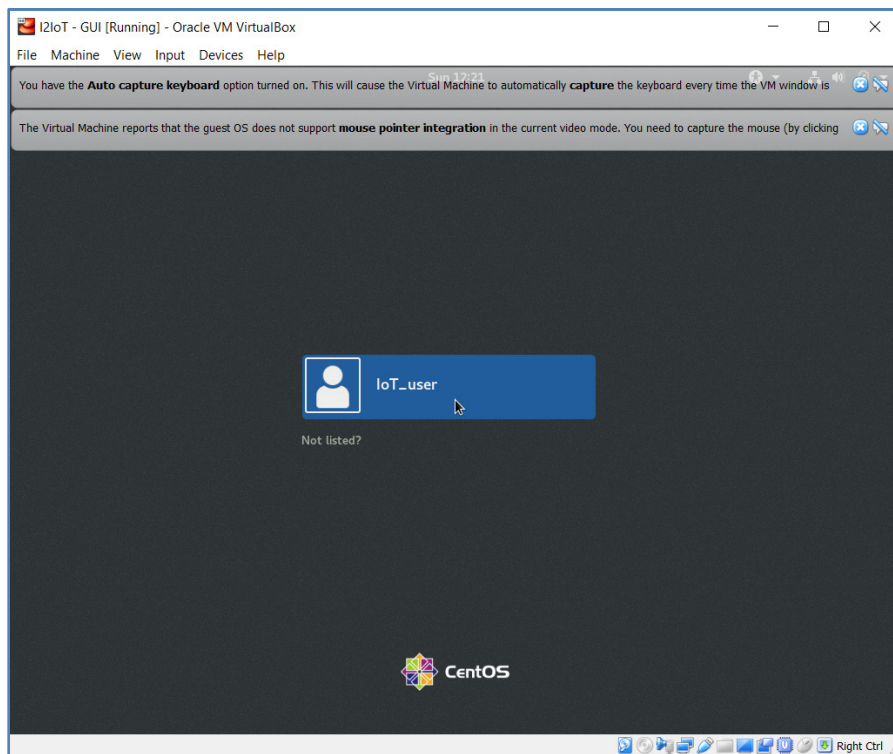
Na Parte 1, você inicia o software de virtualização VirtualBox e faz login na VM I2IoT Server.

Etapa 1: Inicie o VirtualBox.

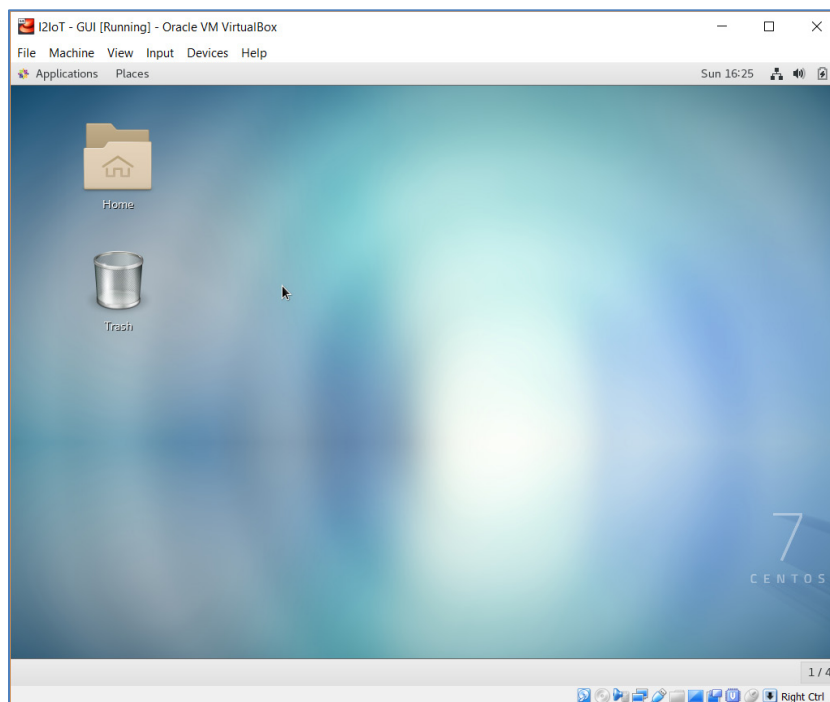
- Depois da instalação do VirtualBox (veja o laboratório 2.1.2.a), o ícone do VirtualBox deverá aparecer no ambiente de trabalho. Clique no ícone para iniciar o VirtualBox.



- b. Clique em **I2IoT – GUI** no painel à esquerda para iniciar a VM.



- c. O nome de usuário padrão é IoT_user, sem senha. Clique na barra azul **IoT_user** no meio da tela para fazer login na VM.



Etapa 2: Navegue até o diretório do Documentos do usuário

- a. Para acessar a command line interface, clique em **Aplicativo** na barra de menu e escolha **Terminal**.

```
IoT_user@localhost:~  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$
```

- b. Use o comando **pwd** para exibir o diretório atual.

```
IoT_user@localhost:~  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ pwd  
/home/IoT_user  
[IoT_user@localhost ~]$
```

- c. Use o comando **ls** para mostrar a lista de conteúdo no diretório atual. Use o comando **ls** com a opção **-l** para exibir informações detalhadas sobre o conteúdo.

```
IoT_user@localhost:~  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ pwd  
/home/IoT_user  
[IoT_user@localhost ~]$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
[IoT_user@localhost ~]$ ls -l  
total 0  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Desktop  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Documents  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 15:24 Downloads  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Music  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Pictures  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Public  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Templates  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Videos
```

- d. Use o comando **cd Documents** para mudar o diretório para o diretório /home/IoT_user/Documents. Verifique usando o comando **pwd**.

```
IoT_user@localhost:~/Documents  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
[IoT_user@localhost ~]$ cd Documents  
[IoT_user@localhost Documents]$ pwd  
/home/IoT_user/Documents  
[IoT_user@localhost Documents]$
```

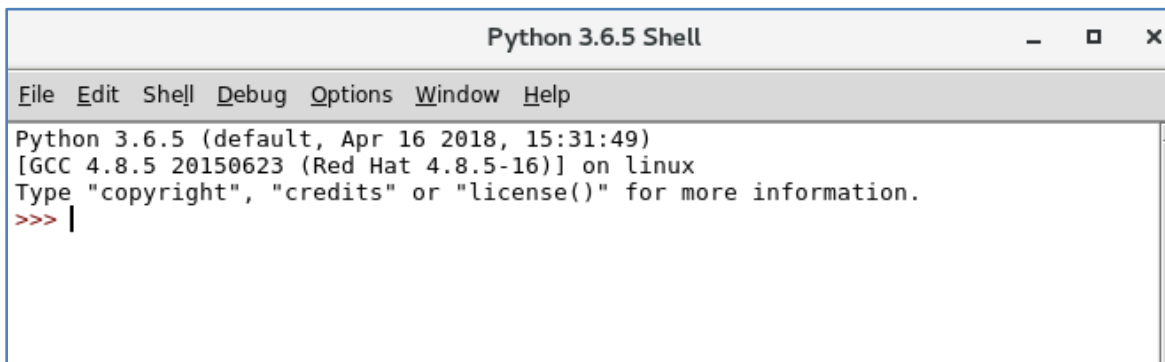
Parte 2: Crie um jogo simples com o Python IDLE

Na Parte 2, você criará um jogo simples. O objetivo do jogo é encontrar um número escolhido por um jogador entre 0 e 1024 usando o método de bissecção.

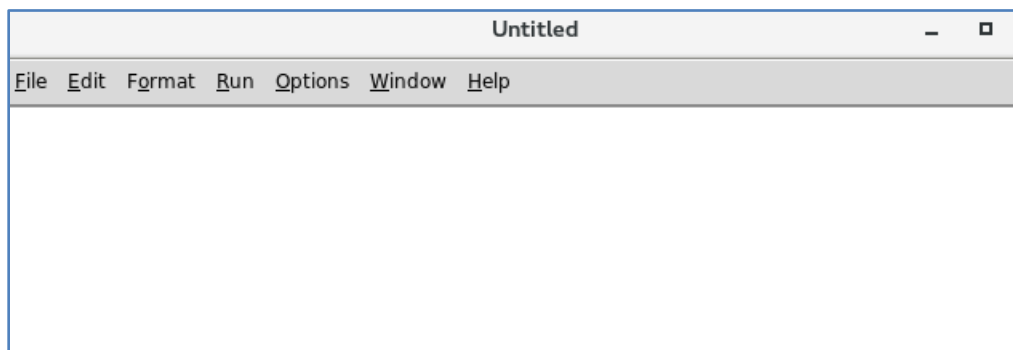
Etapa 1: Inicie o IDLE.

- Use o comando **idle3** para iniciar o IDLE. Por padrão, ele começa na janela Shell do Python ou do interpretador interativo. Você já está familiarizado com o interpretador interativo.

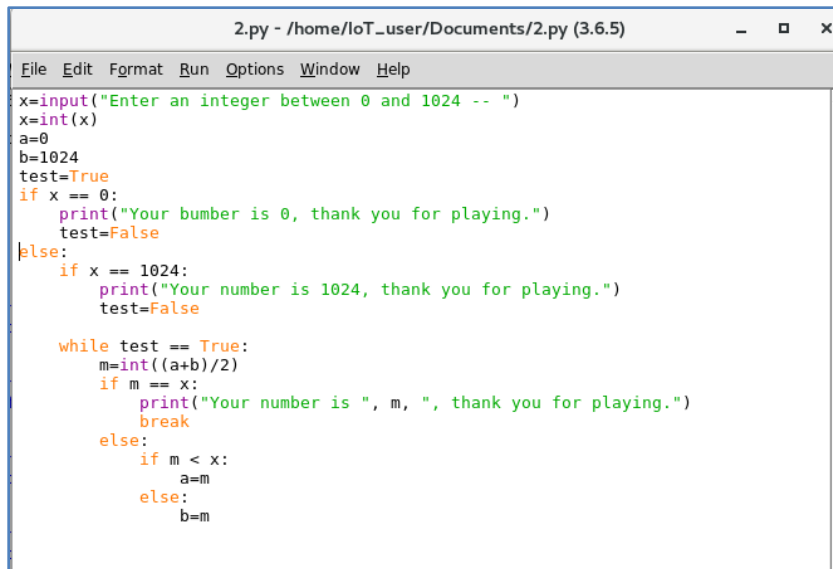
```
[IoT_user@stueverj-vm2 Documents]$ idle3
```



- Clique em **Arquivo** -> **Novo arquivo** para abrir um novo script Python (sem título).



- Digite os códigos no script e observe que os códigos são codificados por cores com abertura e fechamento de parêntesis correspondentes.



```
2.py - /home/IoT_user/Documents/2.py (3.6.5)
File Edit Format Run Options Window Help
x=input("Enter an integer between 0 and 1024 -- ")
x=int(x)
a=0
b=1024
test=True
if x == 0:
    print("Your bumber is 0, thank you for playing.")
    test=False
else:
    if x == 1024:
        print("Your number is 1024, thank you for playing.")
        test=False
    while test == True:
        m=int((a+b)/2)
        if m == x:
            print("Your number is ", m, ", thank you for playing.")
            break
        else:
            if m < x:
                a=m
            else:
                b=m
```

- d. Clique em **Arquivo** -> **Salvar** e salve o script atual como 2.py no diretório atual. Clique no botão **Salvar**.
- e. Clique em **Executar** -> **Executar módulo** (ou pressione F5). A janela de shell exibirá o resultado.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 102463
Your number is 63 , thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 --0
Your bumber is 0, thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 1024
Your number is 1024, thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 72
Your number is 72 , thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 12
Your number is 12 , thank you for playing.
>>>
Ln: 30 Col: 4
```

- f. Solucione o problema, se ocorrer um erro durante a avaliação de sintaxe do código.

Reflexão

Como capturar se o jogador inserir um número além do intervalo de 0 a 1024?

Como capturar se o jogador inserir um número flutuante?
