

The background is a blurred image of a computer screen. On the left, there is a sidebar with several icons and text labels, some of which are highlighted in orange. The main area of the screen displays lines of code in a dark-themed editor. The code is written in a syntax-highlighted language, likely Ruby, and includes comments in Portuguese. The text is out of focus, but some words like 'require', 'File.expand_path', 'abort', 'spec_helper', 'rspec/rails', 'capybara/rspec', and 'capybara/rails' are visible. The overall tone is professional and technical.

DESENVOLVIMENTO COM AUTOMAÇÃO ROBÓTICA DE PROCESSOS - RPA

Texto base

4

Usando Python

Osvaldo Kotaro Takai & Ana Cristina dos Santos

Resumo

O objetivo deste artigo é aprender a criar Bots que consigam executar Scripts escritos na Linguagem Python.

4.1. Introdução

O framework Automation 360 nem sempre possui todas as ações necessárias para a criação de bots. Um exemplo disso é a ausência do operador que retorna o resto da divisão inteira (MOD, ou %). Nesses casos, o framework permite chamar um script Python, ou VBScript, previamente construído, para cobrir essas faltas.

O bot desta semana irá chamar um script Python que simplesmente retorna a data e hora do computador ao invés da implementação do operador MOD, por possibilitar o uso de uma nova ação: **String: Dividir**.

No final deste artigo será apresentado o script que possibilitará implementar o operador MOD.

4.2. Instalação do Python no computador local

Para que o bot possa executar scripts Python, o computador local deve ter o interpretador Python instalado. Para verificar se o Python já está instalado no computador, os seguintes passos devem ser seguidos:

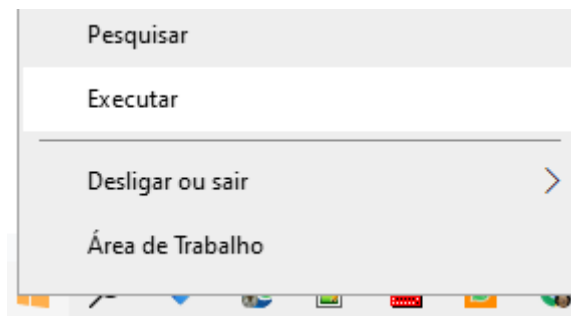
1. Abrir o **Prompt de Comandos do Windows**. Para isso basta clicar com o **botão direito do mouse** no menu iniciar do Windows (**Figura 4.1**), selecionar a opção **executar** (**Figura 4.2**) e digitar “cmd” (**Figura 4.3**). Ao final a tela de Prompt será exibida (**Figura 4.4**).

Figura 4.1: Botão iniciar do Windows



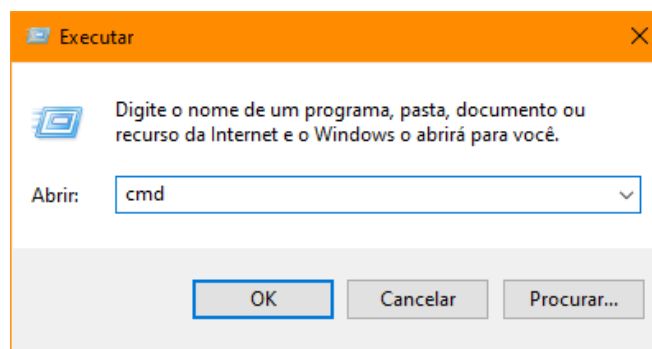
Fonte: do autor, 2021

Figura 4.2: Opção Executar



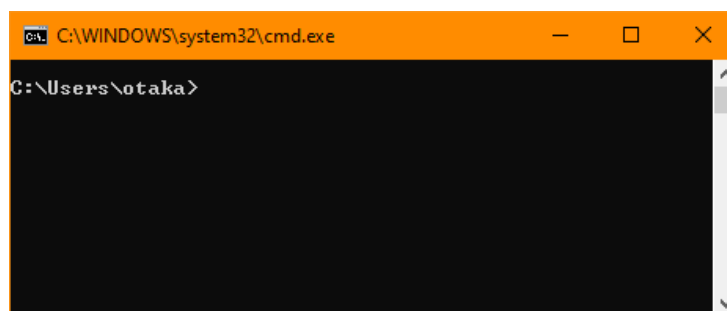
Fonte: do autor, 2021

Figura 4.3: Tela de Executar do Windows



Fonte: do autor, 2021

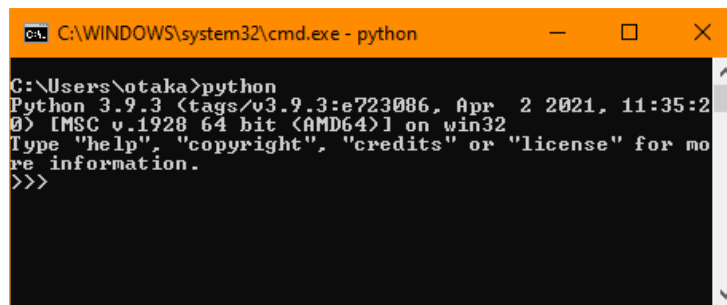
Figura 4.4: Tela de Prompt de Comandos do Windows



Fonte: do autor, 2021

2. Na tela de Prompt do Windows, digitar “python”. Se o console de comandos do python for exibido (**Figura 4.5**), então isso significa que o python está instalado no computador. Caso contrário, será necessário instalar o Python. Pode-se seguir as instruções do Python Brasil para realizar a instalação (PYTHON BRASIL, 2021). Após a instalação, é recomendável retornar ao passo 1 e verificar novamente se o Python está instalado no computador.

Figura 4.5: Console do Python



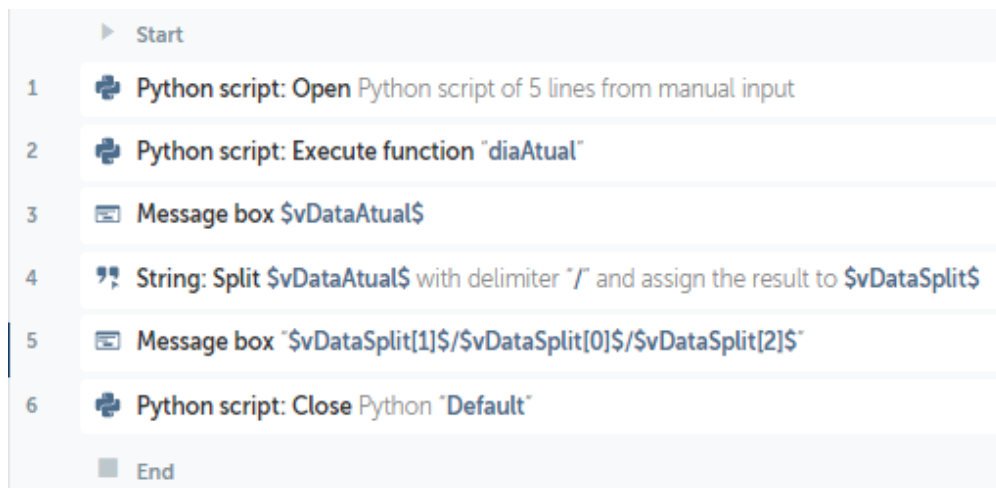
```
C:\WINDOWS\system32\cmd.exe - python
C:\Users\otaka>python
Python 3.9.3 (tags/v3.9.3:eb723086, Apr 2 2021, 11:35:20)
[MSC v.1928 64 bit <AMD64>] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
```

Fonte: do autor, 2021

4.3. A lista de ações do bot da semana

As ações do bot desta semana são bem simples como pode ser observado na **Figura 4.6**.

Figura 4.6 - Lista de ações do bot



Fonte: do autor, 2021

Um novo bot deve ser criado e as ações acima arrastadas em sua lista. As configurações de cada ação serão apresentadas nas subseções a seguir.

4.3.1 Script Python: Open

A configuração desta ação deve seguir de acordo com a **Figura 4.7**. O primeiro campo “Python session” define um nome para o Script Python. Isso permite que vários scripts python sejam definidos num mesmo bot.

O segundo item da configuração permite definir a origem do script python. Se a primeira opção fosse selecionada a ação iria obter o script python de um arquivo que seria especificado no campo “Python file”. Entretanto, este não é o caso para este bot, uma vez que o script que será utilizado é bastante simples.

Portanto a opção é “Manual input”, onde o script em Python foi inserido.

Figura 4.7: Configuração da ação Script Python: Open

Python script: Open Redefinir Aplicar

Opens a Python script

Python session

Python file

Procurar...

☒ Manual input

Enter script here

```

1 from datetime import datetime
2
3 def diaAtual():
4     return datetime.now().strftime("%m/%d/%Y")
5

```

Python runtime version

☐ 2

☒ 3

Fonte: do autor, 2021

O script Python que deve ser escrito neste campo precisa ser compatível com o interpretador instalado no computador local que, nesta demonstração apresentada na **Figura 4.5**, é da versão 3.

A linha 1 faz a importação da função **datetime** da biblioteca de datas do Python. Essa função é utilizada pela função **diaAtual()**, na linha 4, onde é retornada a data atual no formato string contendo o mês, dia e ano separados por barras (“/”).

No último item, “Python runtime version”, foi selecionada a opção 3 para indicar a versão do Python instalada no computador local.

4.3.2 Python script: Execute function

Esta ação permite chamar a função Python definida na ação anterior: **Script Python: Open**. A vinculação desta ação para a anterior se dá pelo nome da sessão que, no caso, foi definida como “Default”, conforme pode ser observado na **Figura 4.8**.

Figura 4.8 - Configuração da ação Python script: Execute function

Python script: Execute function Redefinir Aplicar

Executes a Python function

Python session
Default

Enter name of function to be executed (opcional)
diaAtual

Arguments to the function (opcional)
Supports 0 or 1 argument

Assign the output to variable (opcional)
vDataAtual - String

Fonte: do autor, 2021

O próximo campo, permite indicar o nome da função Python que será chamada. No caso deste bot, a função será **diaAtual** sem os parênteses. Como esta função não possui nenhum parâmetro, então nada foi especificado no campo “Arguments to the function (optional)”.

O último campo permite especificar a variável que receberá o retorno da função diaAtual. Como o valor de retorno é uma string, então a variável vDataAtual foi criada como uma string.

4.3.3 Message box

Esta ação exibirá a data exatamente como ela foi retornada pela função diaAtual, como pode ser vista na **Figura 4.9**. Ou seja, esta ação exibirá a data com o **mês**, seguido do **dia** e **ano**.

Figura 4.9: Configuração da ação Message box

Message box

Displays a message box

Enter the message box window title

06 - Usando Python

Enter the message to display

\$vDataAtual\$

Scrollbar after lines

30

☐ Close message box after

Fonte: do autor, 2021

4.3.4 String: Split

O script Python foi construído para que retornasse o mês antes do dia propositalmente, para motivar o uso desta ação. **String: Split** permite fatiar a String com base num delimitador. Neste caso, o delimitador utilizado é o barra ("/"), conforme pode ser observado na **Figura 4.10**.

Além disso, foi especificado que o delimitador não é sensível ao case: "Not case sensitive", isto é, que a ação não considerará letras maiúsculas e minúsculas como sendo diferente.

A opção "All possible" de "Split into substrings" indica que se deseja obter todas as substring possíveis que, no caso, são três: "mês", "dia" e "ano". Se a opção "Only" tivesse sido escolhida, a ação abriria um novo campo para indicar quantas substring se deseja retornar; por exemplo, se nesse campo fosse indicado 2, então a ação retornaria "mês" e "dia/ano".

Figura 4.10: Configuração da ação String: Split

String: Split Redefinir Aplicar

Splits the source string into multiple strings using a delimiter.

Source string

Delimiter

Delimiter is
☐ Case sensitive
☒ Not case sensitive

Split into substrings
☒ All possible
☐ Only

Assign the output to list variable

Fonte: do autor, 2021

Por último, a variável vDataSplit do tipo **List** foi criada para receber as substrings fatiadas (**Figura 4.11**).

Figura 4.11: Configuração da variável vDataSplit

Edit variable Cancel Apply

Type

Subtype

Name

Max characters = 50

Description (optional)

Max characters = 255

☐ Use as input
☐ Use as output
☒ Constant (read-only)

⚠ Default value
 This list is empty

4.3.5 Message box

Por fim, a data no formato dia/mês/ano deve ser exibida. Para tanto, a configuração apresentada na **Figura 4.12** deve ser seguida.

Figura 4.12: Configuração da ação Message box

Message box [Redefinir] [Aplicar]

Displays a message box

Enter the message box window title

06 - Usando Python

Enter the message to display

\$vDataSplit[1]\$/ \$vDataSplit[0]\$/ \$vDataSplit[2]\$/

Scrollbar after lines

30

☐ Close message box after

Seconds

30

Fonte: do autor, 2021

A novidade aqui se encontra na forma como serão disponibilizados os elementos da lista `vDataSplit`. A lista inicia no índice zero; assim, o mês encontra-se na posição 0, o dia na posição 1 e ano na posição 2. Logo, as posições 0 e 1 devem ser invertidas para a exibição no formato desejado:

- `$vDataSplit[1]$/`: será exibido o ano.
- `/`: será exibida a barra.
- `$vDataSplit[0]$/`: será exibido o mês.
- `/`: será exibida a barra.
- `$vDataSplit[2]$/`: será exibido o ano.

4.3.6 Python script: Close

A sessão “Default” criada pela ação Python script Execute function precisa ser finalizada. Isso é feito por esta ação.

4.3.7 Salvar e executar

Se não houver erros, basta salvar e executar o bot.

4.4. Considerações finais

Se o objetivo fosse simplesmente de criar um bot para exibir a data local, o bot não precisaria ter feito uso de script Python, já que o Automation 360 fornece variáveis de sistema prontas que podem ser consultadas para esse fim. O objetivo aqui foi puramente didático para exemplificar como o framework faz para executar scripts escritos na linguagem Python.

Para atender ao compromisso assumido no início deste artigo, segue o script em Python que pode ser utilizado para implementar o operador MOD ou % que obtém o resto de uma divisão entre dois números inteiros, já que o framework Automation 360 ainda não fornece esse operador.

```
def mod(args):  
    a, b = args.split(",")  
    return int(a) % int(b)
```

O operador mod recebe apenas um argumento, pois o Automation 360 não permite passar mais de um argumento para uma função Python. Assim, para chamar essa função, os operandos devem ser passados como uma única string separados por um vírgula.

Como um desafio, tente fazer um bot que leia uma data de nascimento do usuário e retorne a sua idade.

Referências

AUTOMATION ANYWHERE AUTOMATION 360. **Using dictionary variable for email properties.** Disponível em: <<https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/commands/cloud-using-email-properties.html>>. Acesso em: 21 jul. 2021.

AUTOMATION ANYWHERE COMMUNITY EDITION. **Formulário para obtenção de acesso à versão community edition do automation anywhere gratuita.** São José – EUA. Disponível em: <<https://www.automationanywhere.com/products/enterprise/community-edition>>. Acesso em: 21 jun. 2021.

AUTOMATION ANYWHERE UNIVERSITY. **Introdução ao automation anywhere.** São José – EUA. Disponível em: <<https://apeople.automationanywhere.com/s/getting-started>>. Acesso em: 21 jun. 2021a.

AUTOMATION ANYWHERE UNIVERSITY. **Trilhas de aprendizagem.** São José – EUA. Disponível em: <<https://university.automationanywhere.com/training/rpa-learning-trails/>>. Acesso em: 21 jun. 2021b.

AUTOMATION ANYWHERE UNIVERSITY. **Email server setting.** Disponível em: <<https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/commands/cloud-configuring-mail-server.html>>. Acesso em: 23 jul. 2021c.

BANIN, S. L. **Python 3: conceitos e aplicações: uma abordagem didática.** São Paulo: Érica, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536530253/>>. Acesso em: 21 jun. 2021.

CHICONI, N. **O que é ASCII, UNICODE e UTF-8.** CCM, 2020. Disponível em: <<https://br.ccm.net/faq/9956-o-que-e-ascii-unicode-e-utf-8>>. Acesso em: 21 jul. 2021.

CHANDRA, R. V.; VARANASI, B. S. **Python requests essentials:** learn how to integrate your applications seamlessly with web services using python requests. Packt Publishing, 2015.

DIGICERT. **The ultimate guide:** what is SSL, TLS and HTTPS? Disponível em: <<https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>>. Acesso em: 23 jul. 2021.

ELMAN, J.; LAVIN, M. **Django essencial** - usando REST, websockets e backbone. São Paulo: Novatec, 2015.

GOOGLE. **Ajuda do administrador do google workspace:** controle o acesso a apps menos seguros. Disponível em: <<https://support.google.com/a/answer/6260879?hl=pt-BR>>. Acesso em: 21 jul. 2021.

JARMUL, K.; LAWSON, R. **Python web scraping**. 2. ed. Birmingham: Packt Publishing, 2017.

LOPES, M. D. e LIMA, W. R. **Análise do índice de massa corporal de funcionários de uma instituição de ensino superior**; EFDeportes.com, Revista Digital. Buenos Aires, ano 18, n. 181, jun. 2013. Disponível em: <<https://www.efdeportes.com/efd181/analise-do-indice-de-massa-corporal-de-funcionarios.htm>>. Acesso em: 21 jul. 2021.

MICROSOFT. **O que são IMAP e POP?** Disponível em: <<https://support.microsoft.com/pt-br/office/o-que-s%C3%A3o-imap-e-pop-ca2c5799-49f9-4079-aeef-ddca85d5b1c9>>. Acesso em: 23 jul. 2021.

PYTHON BRASIL. **Instalando o python 3 no windows**. Disponível em: <<https://python.org.br/instalacao-windows/>>. Acesso em: 24 jul. 2021.

WDG AUTOMATION – AN IBM COMPANY. **7 pilares essenciais para projetos de RPA bem-sucedidos**. São Paulo: Newsletter WDG. Disponível em: <<https://www.wdgautomation.com/7-pilares-essenciais-para-projetos-de-rpa-bem-sucedidos/>>. Acesso em: 21 jun. 2021.