C:\Users\PRTS\Documents\GitHub [master +3 ~0 -0 !]> get-help about\_hash\_tables

TÓPICO

about\_Hash\_Tables

DESCRIÇÃO RESUMIDA

Descreve como criar, usar e classificar tabelas de hash no Windows

PowerShell.

DESCRIÇÃO LONGA

Uma tabela de hash, também conhecida como um dicionário ou matriz

associativa, é uma estrutura de dados compacta que armazena um ou

mais pares nome/valor. Por exemplo, uma tabela de hash pode conter

uma série de nomes e IDs de funcionário, nomes de computadores e

endereços IP ou IDs e texto de mensagem.

As tabelas de hash são frequentemente usadas porque são muito

eficientes para localizar e recuperar dados. Você pode usar

tabelas de hash para armazenar listas e criar propriedades

calculadas no Windows PowerShell. E o Windows PowerShell tem um

cmdlet, ConvertFrom-StringData, que converte cadeias de

caracteres em uma tabela de hash.

Criando tabelas de hash

Os itens em uma tabela de hash são organizados em pares

nome/valor, como:

Msg1="Digite sua senha."

Msg2="O parâmetro de caminho é obrigatório."

Msg3="O alias de Get-Command é gcm."

Os valores são mapeados ou associados aos nomes para que o Windows

PowerShell retorne o valor, quando você enviar o nome.

No Windows PowerShell, a sintaxe de uma tabela de hash é como segue:

@{ <nome> = <valor>; [<nome> = <valor> ] ...}

Quando você criar uma tabela de hash, siga estas diretrizes:

- Inicie a tabela de hash com um sinal de arroba (@).

- Coloque a tabela de hash entre chaves ({}).

- Digite um ou mais pares nome-valor para o conteúdo da

tabela de hash.

- Use um sinal de igual (=) para separar cada nome de seu valor.

- Use um ponto-e-vírgula (;) para separar os pares nome/valor.

- Se um nome ou valor contém espaços, inclua-o entre em aspas.

Por exemplo, uma tabela de hash das mensagens anteriores do

usuário se parece com:

@{

Msg1="Digite sua senha.";

Msg2="O parâmetro é obrigatório.";

Msg3="O alias de Get-Command é gcm."; }

Para usar uma tabela de hash em scripts e comandos, salve-a em uma

variável. O valor da variável é um objeto de tabela de hash

(System.Collections.Hashtable) e cada nome nos pares nome/valor é uma

propriedade do objeto de tabela de hash.

Os comandos a seguir salvam a tabela de hash de mensagem de

usuário na variável $a e usa o método de ponto para exibir os

valores.

C:\PS> $a = @{

>> Msg1="Digite sua senha.";

>> Msg2="O parâmetro de caminho é obrigatório.";

>> Msg3="O alias de Get-Command é gcm.";

>> }

C:\PS> $a

Name Value

---- -----

Msg1 Digite sua senha.

Msg3 O alias de Get-Command é gcm.

Msg2 O parâmetro de caminho é obrigatório.

C:\PS> $a.Msg1

Digite sua senha.

As tabelas de hash não são limitadas a um tipo de dados. Você

pode digitar qualquer tipo de dados em uma tabela de hash e

combinar tipos de dados em uma única tabela de hash. Por

exemplo, você pode compilar uma tabela de hash que contém um

inteiro, uma chamada para um cmdlet e uma cadeia de caracteres.

Classificando tabelas de hash

Para classificar a tabela de hash alfabeticamente por chaves ou

valores, use o método GetEnumerator de tabelas de hash para obter as

chaves e valores na tabela de hash e então use o cmdlet Sort-Object para

classificá-los.

Por exemplo, o comando a seguir classifica a tabela de hash em $a

alfabeticamente por meio de chaves.

C:\PS> $a.getenumerator() | sort-object - chave de propriedade

Name Value

---- -----

Msg1 Digite sua senha.

Msg2 O parâmetro de caminho é obrigatório.

Msg3 O alias de Get-Command é gcm.

O comando a seguir usa o mesmo método para classificar os

valores de hash em ordem decrescente.

C:\PS> $a.getenumerator() | sort-object -property value

-descending

Name Value

---- -----

Msg2 O parâmetro de caminho é obrigatório.

Msg3 O alias de Get-Command é gcm.

Msg1 Digite sua senha.

ConvertFrom-StringData

O cmdlet ConvertFrom-StringData converte uma cadeia de

caracteres ou uma cadeia de caracteres here de pares nome/valor

em uma tabela de hash. Você pode usar o cmdlet ConvertFrom-Strin

gData com segurança na seção Dados de um script e com o cmdlet

Import-LocalizedData para exibir mensagens para o usuário na

cultura da interface do usuário atual.

As cadeias de caracteres here são especialmente úteis quando os

valores na tabela de hash incluem aspas. (Para obter mais

informações sobre cadeias de caracteres here, consulte

about\_Quoting\_Rules.)

O exemplo a seguir mostra como criar uma cadeia de caracteres

here das mensagens do usuário no exemplo anterior e como usar

ConvertFrom-StringData para convertê-las de cadeia de

caracteres para tabela hash.

O comando a seguir cria uma cadeia de caracteres here dos pares

nome/valor e salva-a na variável $string.

C:\PS> $string = @"

Msg1="Digite sua senha."

Msg2="O parâmetro de caminho é obrigatório."

Msg3="O alias de Get-Command é gcm."

"@

Este comando usa o cmdlet ConvertFrom-StringData para converter a

cadeia de caracteres here em uma tabela de hash.

C:\PS> convertfrom-stringdata $string

Name Value

---- -----

Msg3 "O alias de Get-Command é gcm."

Msg2 "O parâmetro de caminho é obrigatório."

Msg1 "Digite sua senha."

CONSULTE TAMBÉM

about\_Arrays

about\_Quoting\_Rules

about\_Script\_Internationalization

ConvertFrom-StringData

Import-LocalizedData

C:\Users\PRTS\Documents\GitHub [master +3 ~0 -0 !]>