Buscando dados em textos (Páginas da Web, Arquivos, Data Scraping)

Moderator: Gio

Post **Reply**

- Print view

Search this topic... Search Advanced search

1 post • Page **1** of **1**

- @

- Quote (./posting.php?mode=quote&f=71&p=317988)
- (javascript:void(0);)
  - @
  - Quote (./posting.php?mode=quote&f=71&p=317988)

17 Mar 2020, 12:59

Bom dia.

Já pensou em escrever um script que te apresente a cotação do dólar atualizada sempre que o abrir? Ou talvez um script onde você possa escrever o nome de um filme e ele te informe todos os dados desse filme? Ou quem sabe uma rotina para extrair dados de produtos através das notas fiscais?

Saiba que tudo isso é mais fácil do que parece e este tutorial tem o objetivo de clarificar como podemos implementar estas ideias a partir do AutoHotkey simplesmente recuperando as informações de dados em formato de texto.

Vamos lá?

1. Dados binários vs Dados de texto

Quando aprendemos sobre a história da programação, somos levados a pensar em dados na sua forma binária, isto é, "zeros e ums". Do ponto de vista da programação em baixo nível, isto faz todo sentido, mas no mundo atual, onde quase toda programação atual é realizada em alto nível, não é mais tão comum vermos dados em sua forma bruta binária.

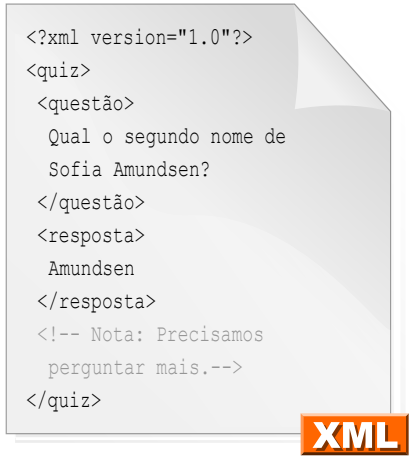
A grande "mágica" da programação está em tornar as coisas cada vez mais fáceis e acessíveis para nós, seres humanos, e isso inclui facilitar as coisas para os próprios programadores. Por este motivo, não faz mais tanto sentido colocar dados em forma binária se isso não for necessário. Assim, muito do nosso trabalho como programadores é atualmente com dados em forma de texto.

Vamos a alguns exemplos:

1. O código-fonte dos seus scripts em AutoHotkey (e de quase qualquer outra linguagem de programação) é escrito em forma de texto.
2. O código-fonte desta página da web, contendo todas as informações da página, é em grande parte escrito em forma de texto (hTml).
3. Os endereços de páginas da web são escritos em forma de texto, e muitos protocolos de comunicação web também (GET e POST).
4. As notas fiscais que as empresas emitem têm suas informações em forma de texto (xml)
5. Muitos dados de telas de programas são armazenados nos bancos de dados em formato de texto.

2. Mas o que são dados de texto?

A grosso modo, são dados onde as sequências binárias podem ser imediatamente traduzidas em caracteres textuais a partir de uma simples tabela comparativa, e quando essa tradução é efetivamente feita, os dados adquirem então novos significados. Um exemplo é o código XML, onde o nome das tags já diz em grande parte para qualquer um que o ler qual o tipo de informações elas contém.



No código XML acima, cujos bytes já foram traduzidos em caracteres textuais através da tabela ASCII, podemos ver o seguinte: a informação Amundsen está "envolvida" entre a "palavra" <resposta> e a "palavra" </resposta>. Isso basicamente nos diz, com um pouco de interpretação, que a informação em questão é a resposta de uma pergunta. Qual pergunta? Provavelmente aquela que está "envolvida" entre as "palavras" <questão> e </questão>. O nome correto dessas "palavras" que estão escritas de forma estruturada seria tags.

Perceba como o uso de dados em forma de texto nos ajudou bastante aqui: se tivéssemos lido os dados em 0s e 1s, nós sequer teríamos percebido quais informações estavam ali, mas como lemos os dados na forma de texto, tudo ficou muito mais fácil entender.

3. E qual a importância de saber trabalhar dados de textos?

No item 2 deste tutorial, vimos como foi fácil entender onde estava a informação "resposta" da pergunta. Então imagine que tenhamos um arquivo XML enorme mais ou menos na mesma formatação que o arquivo acima, mas com muitas perguntas e respostas. Seria muito interessante se pudéssemos criar um script onde haveria uma tabela em que cada linha teria uma pergunta e uma resposta. Também poderíamos criar um script onde selecionaríamos aleatoriamente uma pergunta para o usuário responder e depois mostraríamos a resposta. Outra possibilidade seria que o nosso script exibisse uma pergunta e uma resposta por dia ao usuário do nosso programa sempre que ele logasse a primeira vez no programa (seria algo como uma curiosidade ou um "você sabia" para que o usuário tivesse uma surpresa boa ao abrir o script).

Todas essas ideias podem facilmente ser implementadas através do AutoHotkey. Isso significa que se você encontrar (quase) qualquer informação em forma de texto, pode trabalhar ela da forma que você quiser usando as ferramentas da linguagem. Pense nas possibilidades: tudo o que está escrito na wikipedia ou em qualquer outra página da internet, as consultas do Google, os títulos de vídeos do Youtube, as postagens no facebook, as informações de notas fiscais nos XML e até os posts desse fórum que utilizamos, todas essas informações estão disponíveis em formato de texto.

4. Ok, entendi, estou interessado. Agora o mais importante: Como fazer?

Pense no XML de exemplo que usamos no item 2 (aquele com a pergunta e resposta). Nós fomos capazes de encontrar rapidamente a resposta da pergunta. Como fizemos isso? Ora, fizemos através dos nomes das tags. Com um pouco de atenção, vamos ver que as tags têm elementos bastante distintivos: elas começam com uma estrutura de abertura que sempre funciona assim: tem um

símbolo de < (menor-que), depois tem o nome da tag (que diz qual a informação que ela contém) tal como resposta, e depois disso tem um símbolo de > (menor-que). Daí, logo depois disso é que vem a informação em sí, tal como Amundsen. Depois disso vem uma nova estrutura de fechamento que sempre funciona quase indêntico à estrutura de abertura, com a diferença que tem uma barra antes do nome da tag.

Isso significa que é bastante fácil para qualquer um entender a estrutura dessa linguagem e com isso ler ela quase como se fosse um texto qualquer. Isso acontece porque o XML é de fato uma linguagem estruturada, ou seja, **os símbolos e as posições informam a natureza de cada parte texto e o texto em sí traz logo depois a parte correspondente a esta natureza**.

É exatamente essa natureza da informação textual, de se apresentar em uma forma previsível, que nos permite criar uma rotina de tratamento desses dados em forma de texto.

No AutoHotkey temos vários comandos e funções que trabalham com texto, e cada um deles faz um pequeno trabalho no texto, tais como:
1. cortar a partir (ou até) uma posição no texto (ou seja, eliminar uma parte do texto que esteja em alguma posição específica contada caractere-a-caractere)
2. procurar a posição de uma palavra ou de uma expressão no texto (ou seja, saber em qual posição caractere-a-caractere essa palavra está).
3. verificar se determinada palavra está contida no texto (para saber se de fato aquela palavra está lá e condicionar alguma ação à essa presença ou não-presença)
4. dividir o texto a partir de um caractere (como o caractere separador de linhas ou o de Tabs, por exemplo)
5. isolar caracteres da posição X até a posição Y no texto,
6. Substituir uma palavra no texto por outra
7. etc, etc, etc.

Esses comandos e funções normalmente possuem nomes com String (ou str) ou então RegEx e abaixo você encontra uma lista de vários deles:

SubStr() (https://www.autohotkey.com/docs/commands/SubStr.htm) : Permite que você corte uma parte do texto iniciada na posição X e terminada na posição Y indicada por você. (exemplo: do texto "eu comi abacate", posso isolar a palavra "abacate" cortando da posição 9 até a posição 15 do texto - os espaços contam como caraceteres!).
StringTrimLeft (https://www.autohotkey.com/docs/commands/StringTrimLeft.htm) : Permite que você elimine um número de caracteres indicado por você iniciando na esquerda do texto. (exemplo, do texto "eu comi abacate", posso isolar a palavra "abacate" eliminando os 8 primeiros caracteres do texto).
StringTrimRight (https://www.autohotkey.com/docs/commands/StringTrimLeft.htm) : Permite que você elimine um número de caracteres indicado por você iniciando na direita do texto. (exemplo, do texto "eu comi abacate", posso isolar a expressão "eu comi" eliminando os 8 últimos caracteres do texto).
IfInString (https://www.autohotkey.com/docs/commands/IfInString.htm) : É um condicional que permite que você verifique se determinada frase ou expressão está contida em um texto. (exemplo, posso verificar se o texto "eu comi abacate" contém (ou não contém) a palavra "cebola".)
StringGetPos (https://www.autohotkey.com/docs/commands/StringGetPos.htm) : Permite que você consulte a posição de uma palavra ou expressão no texto (por exemplo, do texto "eu comi abacate", se eu buscar "abacate" vou receber como resposta que o número 9, pois é a partir da 9a posição da esquerda para a direita que inicia a palavra abacate. Se a palavra não existe, o comando também indica).
StringLen (https://www.autohotkey.com/docs/commands/StringLen.htm) : Permite que você consulte o número total de caracteres de um texto. (por exemplo, do texto "eu comi abacate", indicaria 15, pois este é o total de caracteres do texto - os espaços também são caracteres!)
StringReplace (https://www.autohotkey.com/docs/commands/StringReplace.htm) : Permite que você substitua uma palavra ou expressão por outra se a primeira estiver presente no texto (exemplo, posso tentar substituir "abacate" por "cebola", e com isso no texto "eu comi abacate" passaria a conter "eu comi cebola").
StringSplit (https://www.autohotkey.com/docs/commands/StringSplit.htm) : Permite que você divida um texto em várias partes tendo como divisor um caractere ou expressão indicado por vocÊ. (por exemplo, se você indicar o caractere do espaço como divisor, e rodar o comando no texto "eu comi abacate" vai ter como resultado 3 variáveis, onde uma vai conter "eu", outra vai conter "comi" e a outra vai conter "abacate". Essas variáveis estão indexadas pelo comando (seus nomes serão previsíveis a partir de um nome indicado por você) e além disso, haverá uma quarta variável te dizendo quantas partes foram criadas pelo comando.
RegExMatch() (https://www.autohotkey.com/docs/commands/RegExMatch.htm) : Comando avançado que permite buscar uma palavra, frase, número ou expressão complexa em um texto a partir de características dela (por exemplo, uma sequência de números que tenha entre 10 e 20 números (sem letras no meio) OU uma sequência determinada por uma palavra seguida de um número qualquer OU até mesmo algo complexo como: uma sequencia de 1 a 3 números, seguida de um ponto, seguida de 1 a 3 números, seguida de uma vírgula, seguida de 1 a 2 números).
RegExReplace() (https://www.autohotkey.com/docs/commands/RegExReplace.htm) : Comando avançado que permite substituir uma palavra, frase, número ou expressão complexa em um texto.

O pulo do gato é entender como podemos encadenar estes comandos uns com os outros para criar rotinas que isolem as informações desejadas do texto. É assim que criamos o passo-a-passo que sempre resolve os problemas na programação.

Imagine o seguinte: Suponha que eu tenho isolado o código da pergunta do item 2 em uma variável. Como fazer para encontrar exibir somente a pergunta e logo depois somente a resposta ao usuário? Veja no exemplo abaixo:

```
TEXTO_COMPLETO := "<?xml version=" . """ . "1.0" . """ . "?><quiz><questão>Qual o segundo nome de Sofia Amundsen?</questão><resposta>Amundsen</resposta></quiz>

StringGetPos, POSICAO_DA_QUESTAO, TEXTO_COMPLETO, <questão>
CARACTERES_A_CORTAR_ESQUERDA := POSICAO_DA_QUESTAO + 9
StringTrimLeft, TEXTO_CORTADO, TEXTO_COMPLETO, %CARACTERES_A_CORTAR_ESQUERDA%
StringGetPos, POSICAO_FINAL_DA_QUESTAO, TEXTO_CORTADO, </questão>
TEXTO_FINAL_QUESTAO := SubStr(TEXTO_CORTADO, 1, POSICAO_FINAL_DA_QUESTAO)
msgbox % TEXTO_FINAL_QUESTAO
```

Execute o código acima para ver que ele isola a pergunta e depois a resposta corretamente. Depois, Tente entender passo-a-passo o que fizemos no código analisando-o acima. Primeiro, nós tínhamos o texto completo do XML. Depois, usamos um StringGetPos para identificar a posição da expressão <questão>. Sabemos que a expressão <questão> tem um total de 9 caracteres, então cortamos, usando o StringTrimLeft, a partir da esquerda do texto, um total de caracteres correspondente à posição da expressão mais 9. Com isso, já cortamos tudo o que estiver ANTES do texto da questão, e agora só nos falta cortar o que estiver DEPOIS. Para fazer isso, nós usamos de novo o StringGetPos, mas dessa vez buscando a posição da expressão </questão> (perceba a barra antes do nome, é isso que identifica uma tag de fechamento no XML). Uma vez encontrada a posição da expressão de fechamento, usamos então a função SubStr() para pegar apenas o texto que esteja ANTES dessa tag. Com isso conseguimos então isolar o texto da pergunta e o exibimos ao usuário em uma caixa de mensagem.

Depois, tudo que tivemos que fazer foi repetir a rotina para isolar dessa vez a informação da resposta, mas veja que é basicamente o mesmo esquema: a única diferença é que como a expressão <resposta> tem 10 caracteres, vamos cortar do texto a posição da expressão + 10 caracteres para isolar corretamente a resposta na esquerda.

Viu como é fácil!? Com apenas 6 linhas de código já conseguimos isolar uma informação de um texto!

5. Caso concreto: recuperando informações em texto da internet

No passo anterior, focamos em trabalhar as informações do texto. Mas como podemos conseguir essas informações? Ora, de várias formas: No caso do XML de uma nota fiscal, por exemplo, este arquivo pode ser baixado diretamente dos servidores da receita ou solicitado do emissor da nota. As vezes também podemos nos deparar com arquivos de texto estruturados criados por programas feitos por terceiros (tipo o programa que roda em um escritório). Para ler arquivos de texto e colocar o texto em variáveis, usamos o comando FileRead (https://www.autohotkey.com/docs/commands/FileRead.htm) . Se o texto estiver em uma página da web, podemos baixar o código-fonte dessa página usando o comando URLDownloadToFile (https://www.autohotkey.com/docs/commands/URLDownloadToFile.htm) e depois ler o conteúdo de texto usando o FileRead.

O fato é que uma vez encontrado um acesso aos dados em forma de texto, podemos trabalhá-los com um script para que tudo fique conforme precisamos.

Assim, vamos ver um caso específico onde vamos obter dados em texto de uma página da internet, depois vamos extrair uma informação útil e em seguida vamos apresentar ao usuário: a cotação do dólar.

Veja como isso é simples de fazer com os comandos RegEx():

```
URLDownLoadToFile, https://www.google.com/search?q=dolar+real, %A_ScriptDir%/DADOS_TEXTO.txt
FileRead, CODIGO_FONTE_PAGINA, %A_ScriptDir%/DADOS_TEXTO.txt

RegExMatch(CODIGO_FONTE_PAGINA, "\d.\d\d Real brasileiro",CODIGO_FILTRADO)

If (CODIGO_FILTRADO = "")
{
```

```
Msgbox, 0x10, Erro, Não foi possível encontrar a informação no código fonte da página.
```

O código acima fez o seguinte: salvou o código-fonte de uma página da web obtida a partir da consulta google "dolar real". Se você fizer essa consulta no próprio Google em seu navageador, vai ver que em um determinado lugar da página, aparece escrito o valor do dólar em reais. É essa informação que nós queremos isolar. Para isolar ela, o que temos que fazer é analisar o código-fonte da página, para entender onde (e como) esta informação está inserida nele.

No caso deste exemplo, observei que em algum lugar do texto do código-fonte sempre aparece a seguinte expressão: um número, seguido de um ponto, seguido de dois números, seguido de um espaço, seguido de "Real brasileiro". Essa informação é exatamente a informação que buscamos.

Então, após nos certificarmos de que outra expressão com as mesmas características não aparece em outros lugares do código-fonte, podemos configurar um comando RegExMatch() simples para isolá-la dentro da variável VALOR_FINAL. Com isso, nós tiramos somente a informação desejada de dentro do enorme texto que é o código-fonte completo da página.

Em seguida tudo o que resta é criar uma frase para exibir essa informação ao usuário, e para isso usamos as variáveis embutidas para recriar a data e hora da consutla, além da cotação obtida, e com isso tive como resultado final neste momento a expressão Cotação do Dólar em 17/03/2020 12:50: R$ 5.02, que é o que exibiremos ao usuário.

Considerando também que a busca do Google se mantém atualizada, podemos garantir que se o script for executado em outros dias, teremos ainda assim a cotação certa do dia em que o script foi executado.

6. Conclusão

Se você seguiu este tutorial do começo ao fim, deve ter aprendido (ou revisto) um monte de coisas. Primeiro, sabe que grande parte da informação útil está disponível para nós em formato de texto, seja em páginas da web ou em arquivos de texto. Depois, viu que é relativamente fácil isolar as informações desejadas dentro de um texto ou código-textual e com isso enriquecer o seu script com várias informações para o usuário. Tenho certeza que após ler este tutorial você teve muitas ideias interessantes para scripts, então que tal fazer algum agora e compartilhar conosco? Seria um excelente exercício!

Se tiver algum dúvda sobre este assunto, sinta-se livre para postar abaixo ou criar um tópico na seção "Ajuda e Suporte Geral". E vamos em frente

Top
Post **Reply**
• Print view
1 post • Page **1** of **1**
Return to “Tutoriais”
Jump to
• AutoHotkey Foundation
• About This Community
• Forum Issues
• AutoHotkey (v2, current version)
• Ask for Help (v2)
• Gaming
• Scripts and Functions (v2)
• Gaming
• Tutorials (v2)
• Tips and Tricks
• Wish List
• Suggestions on Documentation Improvements
• Bug Reports
• AutoHotkey Development
• AutoHotkey_H

- Ask for Help
- Development
- Editors
- Adventure IDE
- Old Topics
- AHK Studio
- Notepad++
- Pulovers Macro Creator
- SciTE4AutoHotkey
- Visual Studio Code
- Announcements
- General Discussion
- AutoHotkey (v1.1 and older)
- Ask for Help (v1)
- Gaming Help (v1)
- Scripts and Functions (v1)
- Gaming Scripts (v1)
- Tutorials (v1)
- Tips and Tricks (v1)
- General
- Other Utilities & Resources
- Other Programming Languages
- C/C++
- ASM
- C#
- Off-topic Discussion
- RPA
- Other languages
- Looking for Volunteers in other languages
- Deutsch (German)
- Ich brauche Hilfe
- Spiele
- Skripte und Funktionen
- Tutorials
- Tooltime
- Allgemeines
- 中文 (Chinese)
- 请求帮助
- 脚本函数
- 教程资料
- 相关工具
- 其他
- Español (Spanish)
- Pedir Ayuda
- Automatización de Juegos
- Scripts y Funciones
- Tutoriales
- Otras Utilidades y Recursos
- General
- Русский (Russian)
- Помощь

- Скрипты для Игр
- Скрипты и библиотеки
- Статьи и руководства
- Прочие ресурсы и ПО.
- Свободное общение
- Français (French)
- J'ai besoin d'aide
- Scripts et Fonctions
- Tutoriels
- Autres Utilitaires et Ressources
- Hors Sujet
- Português (Portuguese)
- Ajuda e Suporte Geral
- Scripts e Funções
- Tutoriais
- Outras Ferramentas e Recursos
- Outros Assuntos
- 한국어 (Korean)

Who is online

Users browsing this forum: No registered users and 1 guest