



TMPL

Template Manager Program Language

Descrição do projeto computacional:

Gerenciador de templates para diferentes formatos de arquivos.

Os templates são voltados para formatos de arquivos baseados em texto como HTML, “.txt”, arquivos de código fonte, documentos e outros. O gerenciador interpreta os arquivos de templates, solicita ao usuário as informações necessárias e gera o arquivo final.

O template é previamente desenvolvido por alguém que entenda a linguagem e a sintaxe, depois pode ser utilizado facilmente por qualquer usuário para gerar arquivos baseados nos templates.

Sintaxe do gerenciador de template

Uma breve descrição da sintaxe dos códigos utilizados nos arquivos de templates do gerenciador de de templates.

Estrutura básica do template

O template é basicamente um arquivo de texto do bloco de notas “.txt” (pode ser com outro nome de extensão de arquivo além de “.txt”). Facilmente você pode criar um arquivo do tipo pelos próprios programas presentes no seu sistema operacional.

Observação: O arquivo de texto deve usar a codificação ANSI para evitar problemas com caracteres especiais do idioma português.

Para editar o arquivo de template você vai precisar usar basicamente um editor de bloco de notas “padrão” de sua preferência (qualquer bloco de notas que acompanha o sistema operacional a princípio deve funcionar).

Visão Geral da estrutura do código

O código do template possui três partes principais que o programa irá interpretar de formas diferentes.

As três partes são:

A 1ª parte: No início do arquivo de texto o programa irá ler os parâmetros do template. Os parâmetros servem para que o programa saiba como lidar com template. Nele haverá informações sobre como interagir com o usuário e como gerar o arquivo final de acordo com as informações fornecidas pelo usuário.

Exemplos de informações contidas nos parâmetros:

- O formato de arquivo (extensão de arquivo, exemplo: “.cpp”, “.txt”, “.html”, etc..) .
- Se o arquivo final deve ser convertido para a codificação UTF-8.
- Quantidade de tipos de inserções de texto ou de dados a serem inseridas no template ao ser gerado o arquivo final.
- Informações sobre opções predefinidas.
- Textos que serão mostrados ao usuário nos diálogos.
- Nome de identificadores das posições que haverá a inserção de informações.

A 2ª parte: Na segunda parte o programa identificará onde começa e onde termina as informações que serão utilizadas para gerar o arquivo final.

Essas informações seriam a princípio texto, para serem utilizadas em formatos que se baseiam em texto, como por exemplo o HTML.

No meio das informações haverá identificadores que serão substituídos por informações selecionadas pelo usuário.

A 3ª parte: Depois da segunda parte o programa ignora o que vem a seguir, isso torna a terceira parte útil para inserir comentários para quem estiver lendo e editando o arquivo de template.

Documentação dos parâmetros da 1ª parte:

Parâmetro de formato:

No parâmetro de formato a pessoa que está desenvolvendo o template fornecerá ao programa o tipo de arquivo que será gerado (extensão de arquivo).

Por exemplo, se o template está sendo feito para gerar um arquivo de texto, é importante você indicar que será um arquivo txt (".txt") para que o windows e os programas entendam de que tipo de arquivo se trata. Para por exemplo, você no seu navegador de arquivos clicar duas vezes sobre o arquivo e automaticamente abrir ele no bloco de notas do windows.

A sintaxe desse parâmetro para um arquivo de texto (".txt") seria assim:

```
<formato>txt</formato>
```

Para outros formatos possíveis seria assim:

Para HTML (".html"):

```
<formato>html</formato>
```

Códigos em C++ (".cpp"):

```
<formato>cpp</formato>
```

Códigos em Python (".py"):

<formato>py</formato>

E assim por diante...

Parâmetro de codificação:

É opcional e serve para alterar a codificação do texto do arquivo final. Por padrão o programa utiliza ANSI porém alguns formatos de arquivo baseados em texto utilizam UTF-8, principalmente por questões de acentuação e padrões do mercado. Por isso, alguns formatos precisam que o arquivo final seja salvo em UTF-8 para que seja compatível.

Exemplo do parâmetro de codificação em UTF-8:

<codificacao>utf8</codificacao>

Observação: o programa vai ler UTF-8 se escrito desta forma: “utf8”

Parâmetro do número de variáveis:

Caso você queira que o programa solicite ao usuário mais de uma informação a ser inserida no arquivo final, você precisa informar a quantidade de informações (chamadas de variáveis pelo programa) que serão solicitadas.

Caso você não informe ele identificará automaticamente como somente uma.

O parâmetro para utilizar duas variáveis, como exemplo seria assim:

```
<numero:variaveis>2</numero:variaveis>
```

Para três variáveis seria assim:

```
<numero:variaveis>3</numero:variaveis>
```

E assim por diante...

Parâmetro de diálogo de variáveis:

Informa ao programa o texto que será mostrado ao usuário quando for solicitadas as variáveis.

Exemplo, diálogo da variável 1:

```
<1:dialogo>Nome do aluno</1:dialogo>
```

Parâmetros de predefinições

Definem respostas pré selecionadas para que o usuário escolha.

Parâmetros de número de predefinições:

Caso não seja fornecido, será 0 (nenhuma predefinição). Define o número de opções predefinidas para a respectiva variável.

Exemplo, 4 predefinições para a variável 2:

<2:predefinicoes>4</2:predefinicoes>

Parâmetros de diálogo de predefinição:

Informa o texto que representará a respectiva opção predefinida da respectiva variável.

Exemplo, nesse caso o diálogo da opção 1 da variável 2:

<2:1:dialogo>Sexo masculino</2:1:dialogo>

Parâmetros de opção de predefinição:

Define o valor a ser inserido na respectiva opção da respectiva variável.

Exemplo, nesse caso a opção 1 da variável 2:

<2:1:opcao>Homem</2:1:opcao>

Parâmetro de nome de variáveis:

O parâmetro de nome de variável é utilizado para definir os nomes das variáveis de acordo com o respectivo número de cada uma. Essa informação de nome será utilizada pelo programa na hora de solicitar informações ao usuário (caso não esteja definido um diálogo para a variável) e quando o programa procurar por “identificadores de inserção” dentro das informações do arquivo base que gerará o arquivo final.

Exemplo de um parâmetro de nome da variável número 1:

<1:variavel>Nome da variavel 1</1:variavel>

Exemplo de um parâmetro de nome da variável número 3:

<3:variavel>Nome terceira variavel</3:variavel>

Exemplo de um parâmetro de nome da variável número 5:

<5:variavel>Quinta</5:variavel>

Caso não seja fornecido o nome de variável, o programa gerará automaticamente um nome padrão baseado no nome da variável.

Exemplo:

“Variavel 1”;

“Variavel 3”;

“Variavel 8”;

Parâmetro de tag de variáveis:

Define o identificador “tag” da respectiva variável. Se não definido será “número:variavel”;

Exemplo, tag da variavel:

<2:tag>texto</2:tag>

O programa procurará um identificador assim:

<texto>nome_variavel</texto>

Documentação da 2ª parte:

Detecção do início e fim das informações que serão utilizadas como base para o arquivo final:

Quando o programa identificar o seguinte trecho ele interromperá a leitura de parâmetros e passa a ler os dados base para o arquivo final.

Identificador do início do arquivo base:

<arquivo>inicio</arquivo>

Quando o programa identificar o seguinte trecho ele ignorará a linha atual e interromperá a leitura da base do arquivo.

Inserção de informações:

O programa pegará as as informações base do arquivo novo e procurará por identificadores de variáveis no meio das informações. Quando ele encontrar ele substituirá o trecho que contém o identificador pela informação fornecida ou selecionada pelo usuário e salvará tudo no arquivo final.

Exemplo de identificador, com a tag “texto” e o nome da variável “nome”:

<texto>nome</texto>

Exemplos de template:

Pergunta o nome e dá as boas vindas:

```
<formato>txt</formato>
```

```
<numero:variaveis>1</numero:variaveis>
```

```
<1:tag>texto</1:tag>
```

```
<1:variavel>Nome</1:variavel>
```

```
<arquivo>inicio</arquivo>
```

```
<texto>Nome</texto>, seja bem vindo!
```

```
<arquivo>fim</arquivo>
```

comentário

Seleciona o sexo da pessoa, pergunta o nome e gera um pequeno formulário em texto:

```
<formato>txt</formato>
```

```
<numero:variaveis>2</numero:variaveis>
```

```
<1:tag>info</1:tag>
```

```
<1:variavel>sexo</1:variavel>
```

```
<1:dialogo>Sexo da pessoa</1:dialogo>
```

**<1:predefinicoes>2</1:predefinicoes>
<1:1:dialogo>Sexo masculino</1:1:dialogo>
<1:1:opcao>Masculino</1:1:opcao>
<1:2:dialogo>Sexo feminino</1:2:dialogo>
<1:2:opcao>Feminino</1:2:opcao>**

**<2:tag>texto</2:tag>
<2:variavel>nome</2:variavel>
<2:dialogo>Nome da pessoa</2:dialogo>**

**<arquivo>inicio</arquivo>
Sexo: <info>sexo</info>
Nome: <texto>nome</texto>
<arquivo>fim</arquivo>**

comentário