



PROJETO FRAMEWORK - CELEPAR
CATÁLOGO DE JAVASCRIPT

Julho – 2006

Sumário de Informações do Documento		
Tipo do Documento: Definição		
Título do Documento: Catálogo de JAVASCRIPT		
Estado do Documento: Elaborado		
Responsáveis: Diego Pozzi		
Palavras-Chaves: Componentes, java, reuso, javascript		
Resumo: Descrição dos códigos javascript gerados para o Grupo Framework		
Número de páginas: 19		
Software utilizados:		
Versão	Data	Mudanças
1.0	14/02/05	
1.1	16/06/05	Cleverson Budel
1.2	04/07/06	Revisão documento por Helio S. Piccinatto, Elisabeth hoffmann e Fabio Sgoda

SUMÁRIO

1 CATÁLOGO DE JAVASCRIPT.....	5
1.1 INTRODUÇÃO.....	5
2 ARQUIVOS JAVASCRIPT.....	5
2.1 UTIL.JS.....	5
2.1.1 Função: SoNumero(nro).....	5
2.1.2 Função: DigitaLetra(obj).....	5
2.1.3 Função: DigitaNumero(obj).....	6
2.1.4 Função: TamanhoMax(obj, tamanhoMaximo).....	6
2.1.5 Exemplos de uso:.....	6
2.2 CNPJ.JS.....	6
2.2.1 Função: ValidaCNPJ(campo).....	6
2.2.2 Função: MascaraCNPJ(campo, codTecla).....	6
2.2.3 Função: CNPJsemMascara(campo).....	7
2.2.4 Função: DesmascaraCNPJ(campo).....	7
2.2.5 Exemplos de uso:.....	7
2.3 CPF.JS.....	7
2.3.1 Função: ValidaCPF(campo).....	7
2.3.2 Função: MascaraCPF(campo, codTecla).....	7
2.3.3 Função: CPFsemMascara(campo).....	8
2.3.4 Função: DesmascaraCPF(campo).....	8
2.3.5 Exemplos de uso:.....	8
2.4 PIS.JS.....	8
2.4.1 Função: ValidaPIS(campo).....	8
2.4.2 Função: MascaraPIS(campo, codTecla).....	8
2.4.3 Exemplo de uso:.....	9
2.5 EMAIL.JS.....	9
2.5.1 Função: ValidaEmail(campo).....	9
2.5.2 Exemplo de uso:.....	9
2.6 FONE.JS.....	9
2.6.1 Função: MascaraFone(campo, codTecla).....	9
2.6.2 Função: DesmascaraFone(campo).....	9
2.6.3 Exemplo de uso:.....	10
2.7 DATA.JS.....	10
2.7.1 Função: ValidaData(campo).....	10
2.7.2 Função: MascaraData(campo, codTecla).....	10
2.7.3 Função: DataMaior(dataIni, dataFim).....	10
2.7.4 Função: DataMenor(dataIni, dataFim).....	10
2.7.5 Função: DataIgual(dataIni, dataFim).....	11
2.7.6 Função: GetDia(strData).....	11
2.7.7 Função: GetMes(strData).....	11
2.7.8 Função: GetAno(strData).....	11
2.7.9 Função: DataInversa(strData).....	11
2.7.10 Função: IncrementaData(strData, numDias).....	11
2.7.11 Exemplos de uso.....	11
2.8 FUNCOES.JS.....	12
2.8.1 Função: strZero(campo, tamanhoCampo).....	12
2.8.2 Função: Trim(campo).....	12
2.8.2 Função: Rtrim(campo).....	13
2.8.3 Função: Ltrim(campo).....	13
2.8.4 Exemplos de uso.....	13
2.9 HORA.JS.....	13
2.9.1 Função: MascaraHora(campo, codTecla, segundo).....	13
2.9.2 Função: ValidaHora(campo, tipoHora).....	13
2.9.3 Exemplos de uso.....	14
2.10 NUMERO.JS.....	14
2.10.1 Função: DigitaNumero(campo).....	14
2.10.2 Função: FormatoMoeda(campo).....	14
2.10.3 Exemplos de uso.....	14

2.11	CALENDARPOPUP.JS.....	14
2.11.1	Função: <i>Calendario(campo, botao)</i>	15
2.11.2	Exemplos de uso.....	15
2.12	WINDOWOPEN.JS.....	15
2.12.1	Função: <i>window_open(url, nomeJanela, recursos)</i>	15
2.12.2	Exemplo de uso.....	15
2.13	MAISINFO.JS.....	15
2.13.1	Função: <i>maisinfo(msg)</i>	15
2.13.2	Função: <i>menosinfo()</i>	16
2.13.3	Exemplo de uso.....	16
2.14	DTREE.JS.....	16
2.14.1	Função: <i>dTree(obj)</i>	16
2.14.2	Comando: <i>add(id, pid, name, url, title, target, icon,iconOpen, open)</i>	16
2.14.3	Comando: <i>document.write(menu)</i>	16
2.14.4	Exemplo de uso.....	16
2.15	COOLMENUS4.JS.....	17
2.15.1	Função: <i>makeMenu</i>	17
2.15.2	Exemplo de uso.....	17
2.16	CALENDAR.JS.....	19
2.16.1	Função: <i>addLink(mes, dia, link)</i>	19
2.16.2	Função: <i>montaCalendario(mes, ano)</i>	19
2.16.3	Exemplo de uso.....	19
2.17	PROTOTYPE.JS.....	19

1 CATÁLOGO DE JAVASCRIPT

1.1 Introdução

O catálogo foi criado com a finalidade de auxiliar o desenvolvedor a conhecer as funções javascript que foram padronizados e disponibilizadas para o desenvolvimento.

Boa parte das funções javascript utilizadas no catálogo já existiam na empresa e foram criados por diversos autores. O serviço realizado pelo Grupo Framework foi o de reuni-las, filtrar com boa possibilidade de reuso e padronizá-las, criando testes unitários para validar as suas funcionalidades.

Das funções existentes algumas foram adaptadas para o projeto e outras criadas para facilitar a utilização pelo desenvolvedor.

O catálogo javascript será apresentado obedecendo a divisão por arquivos e funções javascript, mostrando assim as funcionalidades de cada função existente no arquivo.

2 ARQUIVOS JAVASCRIPT

2.1 util.js

Este arquivo contém funções de uso geral que dão suporte ao funcionamento de outros scripts, por isso deverá ser inserido em todas as páginas.

2.1.1 Função: SoNumero(nro)

Descrição: Esta função retorna somente os números de um conjunto de string passado como parâmetro.

@param String.

@return String (somente com números).

2.1.2 Função: DigitaLetra(obj)

Descrição: Esta função permite que sejam digitadas somente letras.
@param componente onde estão sendo digitadas as letras.
@return o componente é recarregado somente com as letras.

2.1.3 Função: DigitaNumero(obj)

Descrição: Esta função permite que sejam digitados somente números.
@param componente onde estão sendo digitados os números.
@return o componente é recarregado somente com os números.

2.1.4 Função: TamanhoMax(obj, tamanhoMaximo)

Descrição: Esta função limita a quantidade de caracteres a serem digitados.
@param componente onde é digitada a string.
@param tamanho máximo permitido para a string do componente.
@return componente recarregado com string truncada com o tamanho máximo.

2.1.5 Exemplos de uso:

- `<input id="dgnum" type="text" value="" onKeyUp="javascript:DigitaNumero (document.getElementById('dgnum'))">`
- `<input id="dgletra" type="text" value="" onKeyUp="javascript:DigitaLetra (document.getElementById('dgletra'))">`
- `<input id="sonum" type="text" value="" onBlur="alert(SoNumero (document.getElementById('sonum').value));">`
- `<input id="tmax" type="text" value="" onBlur="javascript:TamanhoMax (document.getElementById('tmax'),'10')">`

2.2 cnpj.js

Este arquivo contém funções voltadas para tratamento do CNPJ.

2.2.1 Função: ValidaCNPJ(campo)

Descrição: Esta função verifica se o CNPJ digitado é válido.
@param componente destinado a digitação do CNPJ.
@return "true" se CNPJ válido, "false" se CNPJ inválido.

2.2.2 Função: MascaraCNPJ(campo, codTecla)

Descrição: Esta função coloca a máscara do CNPJ no momento em que está sendo digitado.
@param componente destinado a digitação do CNPJ.
@param código da tecla pressionada.
@return CNPJ com a máscara.

2.2.3 Função: CNPJsemMascara(campo)

Descrição: Esta função elimina os caracteres não numéricos da string contida no campo do CNPJ digitado deixando-o sem máscara e com no máximo 14 caracteres.

@param componente destinado a digitação do CNPJ.

@return CNPJ só com os números,ou seja, sem máscara.

2.2.4 Função: DesmascaraCNPJ(campo)

Descrição: Esta função retira caracteres não numéricos do CNPJ.

@param componente destinado a digitação do CNPJ (com a máscara).

@return CNPJ só com os números,ou seja, sem máscara.

2.2.5 Exemplos de uso:

- `<input id="CNPJ" type="text" value="" onKeyUp= "javascript: MascaraCNPJ (document.getElementById('CNPJ'),event)" onBlur= "javascript:if (ValidaCNPJ (document.getElementById('CNPJ'))){alert('OK');} else {alert('inválido')}">`
- `<input id="CNPJ" type="text" value="" onKeyUp="javascript: CNPJsemMascara(document.getElementById('CNPJ'));TamanhoMax (document.getElementById('CNPJ'),18)" onBlur="javascript:if (ValidaCNPJ(document.getElementById('CNPJ'))){ alert('OK');} else {alert('inválido')}">`
- `<input id="CNPJ" type="text" value="" onKeyUP="TamanhoMax (document.getElementById('CNPJ'),18);" onBlur="javascript: DesmascaraCNPJ(document.getElementById('CNPJ'))">`

2.3 cpf.js

Este arquivo contém funções voltadas para tratamento do CPF.

2.3.1 Função: ValidaCPF(campo)

Descrição: Esta função verifica se o CPF é válido.

@param componente onde será digitado o CPF.

@return "true" se CPF válido, "false" se CPF inválido.

2.3.2 Função: MascaraCPF(campo, codTecla)

Descrição: Esta função coloca a máscara do CPF no momento em que está sendo digitado.

@param componente onde será digitado o CPF.

@param código da tecla pressionada.

@return CPF com a máscara.

2.3.3 Função: CPFsemMascara(campo)

Descrição: Esta função elimina os caracteres não numéricos da string contida no campo do CPF digitado deixando-o sem máscara e com no máximo 11 caracteres.

@param componente onde está sendo digitado o CPF.

@return CPF só com os números, ou seja, sem máscara.

2.3.4 Função: DesmascaraCPF(campo)

Descrição: Esta função retira a máscara do CPF (caracteres não numéricos).

@param componente onde será digitado o CPF com a máscara.

@return CPF só com os números, ou seja, sem máscara.

2.3.5 Exemplos de uso:

- `<input id="CPF" type="text" value="" onKeyUp="javascript:MascaraCPF(document.getElementById('CPF'),event)" onBlur="javascript:if (ValidaCPF(document.getElementById('CPF'))){alert('OK');} else {alert('inválido')}">`
- `<input id="CPF" type="text" value="" onKeyUp="javascript:CPFsemMascara(document.getElementById('CPF'));TamanhoMax(document.getElementById('CPF'),11)" onBlur="javascript:if (ValidaCPF(document.getElementById('CPF'))){ alert('OK');} else {alert('inválido')}">`
- `<input id="CPF" type="text" value="" onKeyUp="TamanhoMax(document.getElementById('CPF'),14);" onBlur="javascript:DesmascaraCPF(document.getElementById('CPF'));">`

2.4 pis.js

Este arquivo contém funções voltadas para tratamento do PIS.

2.4.1 Função: ValidaPIS(campo)

Descrição: Função que verifica se o PIS é válido.

@param componente(input type="text") onde encontra-se o PIS.

@return "true" - PIS válido, "false" - PIS inválido.

2.4.2 Função: MascaraPIS(campo, codTecla)

Descrição: Esta função coloca a máscara do PIS no momento em que está sendo digitado.

@param componente onde será digitado o PIS.

@param código da tecla pressionada.

@return PIS com a máscara.

2.4.3 Exemplo de uso:

- ```
<input id="PIS" type="text" value="" onKeyUp="javascript:MaskaraPIS
(document.getElementById('PIS'),event)" onBlur="javascript:if
(ValidaPIS(document.getElementById('PIS'))){alert('OK');} else
{alert('inválido')}">
```

## 2.5 email.js

Este arquivo contém funções voltadas para tratamento do e-mail.

### 2.5.1 Função: ValidaEmail(campo)

Descrição: Esta função verifica se o e-mail é válido.

@param componente onde será digitado o e-mail.

@return "true" se e-mail válido, "false" se e-mail inválido.

### 2.5.2 Exemplo de uso:

- ```
<input type="text" ID="email" value="" onBlur=" if (ValidaEmail
(document.getElementById('email'))) {alert('OK');} else
{alert('inválido');}">
```

2.6 fone.js

Este arquivo contém funções voltadas para tratamento de número de telefone.

2.6.1 Função: MaskaraFone(campo, codTecla)

Descrição: Esta função coloca a máscara do telefone no momento em que está sendo digitado.

@param componente onde será digitado o telefone.

@param código da tecla pressionada.

@return telefone com a máscara.

2.6.2 Função: DesmaskaraFone(campo)

Descrição: Esta função retira a máscara do telefone (caracteres não numéricos).

@param componente onde será digitado o telefone com a máscara.

@return telefone só com os números, ou seja, sem máscara.

2.6.3 Exemplo de uso:

- ```
<input id="FONE" type="text" value="" onKeyUp="javascript:MascaraFone
(document.getElementById('FONE'),event)" onBlur="javascript: if
(document.getElementById('FONE').value != '') {alert(DesmascaraFone
(document.getElementById('FONE').value));}">
```

## 2.7 data.js

Este arquivo contém funções voltadas para tratamento de data.

### 2.7.1 Função: ValidaData(campo)

Descrição: Esta função verifica se a data é válida no formato dd/mm/aaaa.

@param componente onde será digitada a data.

@return true se data válida, false se data inválida.

### 2.7.2 Função: MascaraData(campo, codTecla)

Descrição: Esta função coloca a máscara da data no momento em que está sendo digitada.

@param componente onde será digitada a data,

@param código da tecla pressionada.

@return data com a máscara.

### 2.7.3 Função: DataMaior(dataIni, dataFim)

Descrição: Esta função verifica se a dataIni é maior que a dataFim.

@param valor do componente onde será digitada a data inicial no formato dd/mm/aaaa.

@param valor do componente onde será digitada a data final no formato dd/mm/aaaa.

@return "true" se dataIni for maior que dataFim, "false" se dataIni for menor ou igual a dataFim.

### 2.7.4 Função: DataMenor(dataIni, dataFim)

Descrição: Esta função verifica se a dataIni é menor que a dataFim.

@param valor do componente onde será digitada a dataIni no formato dd/mm/aaaa.

@param valor do componente onde será digitada a dataFim no formato dd/mm/aaaa.

@return "true" se dataIni for menor que dataFim, "false" se dataIni for maior ou igual a dataFim.

### 2.7.5 Função: DataIguar(dataIni, dataFim)

Descrição: Esta função verifica se a dataIni é igual a dataFim.

@param valor do componente onde será digitada a dataIni no formato dd/mm/aaaa.

@param valor do componente onde será digitada a dataFim no formato dd/mm/aaaa.

@return "true" se dataIni for igual a dataFim, "false" se dataIni for diferente da dataFim.

### 2.7.6 Função: GetDia(strData)

Descrição: Esta função retorna o dia de uma determinada data.

@param valor do componente onde está a data digitada no formato dd/mm/aaaa.

@return o dia da data recebida, com dois dígitos.

### 2.7.7 Função: GetMes(strData)

Descrição: Esta função retorna o mês de uma determinada data.

@param valor do componente onde está a data digitada no formato dd/mm/aaaa.

@return o mês da data recebida, com dois dígitos.

### 2.7.8 Função: GetAno(strData)

Descrição: Esta função retorna o ano de uma determinada data.

@param valor do componente onde está a data digitada no formato dd/mm/aaaa.

@return o ano da data recebida, com quatro dígitos.

### 2.7.9 Função: DataInversa(strData)

Descrição: Esta função retorna a data invertida no formato aaaammdd.

@param valor do componente onde está a data digitada no formato dd/mm/aaaa.

@return data invertida no formato aaaammdd.

### 2.7.10 Função: IncrementaData(strData, numDias)

Descrição: Esta função retorna uma determinada data acrescida do número de dias desejado.

@param valor do componente onde está a data a ser acrescentada o número de dias, digitada no formato dd/mm/aaaa.

@param valor do número de dias a acrescentar na data.

@return data acrescida com o número de dias, no formato dd/mm/aaaa.

### 2.7.11 Exemplos de uso.

- `<input type="text" id="Data" value="" onKeyUp="javascript:MascaraData (document.getElementById('Data'),event)" onBlur="if (ValidaData (document.getElementById('Data'))){alert('Data OK!!');} else {alert('Data Inválida!!!');}">`
- `<input type="text" id="DataIni" value="" onKeyUp="javascript:MascaraData(document.getElementById('DataIni'),event)" onBlur="if (ValidaData (document.getElementById('DataIni'))){alert('Data OK!!');} else {alert('Data Inválida!!!');}">`  
`<input type="text" id="DataFim" value="" onKeyUp="javascript:MascaraData(document.getElementById('DataFim'),event)" onBlur="if (ValidaData (document.getElementById('DataFim'))){alert('Data OK!!');} else {alert('Data Inválida!!!');}">`
- `<input type="button" value="compararMaior" onclick="javascript:if (DataMaior(document.getElementById('DataIni').value, document.getElementById('DataFim').value)){alert('Data Ini é Maior');} else {alert('Data Ini não é maior');}">`
- `<input type="button" value="compararMenor" onclick="javascript:if (DataMenor(document.getElementById('DataIni').value, document.getElementById('DataFim').value)){alert('Data Ini é Menor');} else {alert('Data Ini não é menor');}">`
- `<input type="button" value="compararIgual" onclick="javascript:if (DataIgual(document.getElementById('DataIni').value, document.getElementById('DataFim').value)){alert('Data é Igual');} else {alert('Data Não é Igual');}">`
- `<input type="text" id="Dt" value="" onKeyUp=" javascript:MascaraData (document.getElementById('Dt'),event)" onBlur="javascript:alert (GetDia(document.getElementById('Dt').value));">`
- `<input type="text" id="Dt" value="" onKeyUp=" javascript:MascaraData (document.getElementById('Dt'),event)" onBlur="javascript:alert (GetMes(document.getElementById('Dt').value));">`
- `<input type="text" id="Dt" value="" onKeyUp=" javascript:MascaraData (document.getElementById('Dt'),event)" onBlur="javascript:alert (GetAno(document.getElementById('Dt').value));">`
- `<input type="text" id="DtI" value="" onKeyUp="javascript:MascaraData (document.getElementById('DtI'),event)" onBlur="javascript:alert (DataInversa(document.getElementById('DtI').value));">`
- `<input type="text" id="DtA" value="" onKeyUp="javascript:MascaraData (document.getElementById('DtA'),event)" onBlur="javascript:alert (IncrementaData(document.getElementById('DtA').value, '2'));">`

## 2.8 funcoes.js

Este arquivo contém funções de uso geral.

### 2.8.1 Função: strZero(campo, tamanhoCampo)

Descrição: Esta função completa o tamanho de um campo com zeros à esquerda.

@param componente onde encontra-se o número a completar.

@param número total de caracteres do componente.

@return número completado com zeros à esquerda.

### 2.8.2 Função: Trim(campo)

Descrição: Esta função retira os espaços em branco que estão à esquerda e à direita de uma string de um campo.

@param componente onde está a string.

@return string sem os espaços em branco à esquerda e à direita.

### 2.8.2 Função: Rtrim(campo)

Descrição: Esta função retira os espaços em branco que estão à direita de uma string.

@param componente onde está a string.

@return string sem os espaços em branco à direita.

### 2.8.3 Função: Ltrim(campo)

Descrição: Esta função retira os espaços em branco que estão à esquerda de uma string.

@param componente onde está a string.

@return string sem os espaços em branco à esquerda.

### 2.8.4 Exemplos de uso.

- `<input id="n1" type="text" value="" onBlur="javascript:strZero (document.getElementById('n1'),8)">`
- `<input id="n2" type="text" value="" onBlur="javascript:Trim (document.getElementById('n2'))">`
- `<input id="n3" type="text" value="" onBlur="javascript:Ltrim (document.getElementById('n3'))">`
- `<input id="n4" type="text" value="" onBlur="javascript:Rtrim (document.getElementById('n4'))">`

## 2.9 hora.js

Este arquivo contém funções voltadas para tratamento de hora.

### 2.9.1 Função: MascaraHora(campo, codTecla, segundo)

Descrição: Esta função coloca a máscara de hora no momento em que está sendo digitada.

@param componente onde será digitada a hora.

@param código da tecla pressionada durante o preenchimento do campo.

@param se "true" - permite digitar os segundos, caso contrário será digitado somente hora e minuto.

@return hora com a máscara.

### 2.9.2 Função: ValidaHora(campo, tipoHora)

Descrição: Esta função verifica se a hora é válida.

@param componente onde encontra-se a hora digitada.

@param se "true" permite que seja digitado hora no sistema 12 horas (AM/PM) caso contrário será verificado no sistema 24 horas.

@return "true" - hora válida, "false" - hora inválida.

### 2.9.3 Exemplos de uso.

- ```
<input id="HORA" type="text" value="" onKeyUp="MascaraHora
(document.getElementById('HORA'),event,true)" onBlur="if
(document.getElementById('HORA').value != '') {if (ValidaHora
(document.getElementById('HORA'),true)) {alert('hora válida');} else
{alert('Inválida!!!');}}">
```

2.10 numero.js

Este arquivo contém funções voltadas para o tratamento de numero.

2.10.1 Função: DigitaNumero(campo)

Descrição: Esta função permite que somente sejam digitados números.

@param componente onde ocorre a digitação.

@return somente números digitado.

2.10.2 Função: FormatoMoeda(campo)

Descrição: Esta função inclui a máscara de moeda nos números digitados, no formato brasileiro com ponto como separador de milhar e virgula como separador de centavos.

@param componente onde serão digitados os números.

@return numero com a máscara no formato moeda brasileira.

2.10.3 Exemplos de uso.

- ```
<input id="num" type="text" value="" onKeyUp="javascript:DigitaNumero
(document.getElementById('num'))">
```
- ```
<input id="md" type="text" value="" onKeyUp="javascript:FormatoMoeda
(document.getElementById('md'))">
```

2.11 calendarpopup.js

Este arquivo contém funções voltadas para o tratamento de calendário.

2.11.1 Função: Calendario(campo, botao)

Descrição: Esta função cria o calendário em uma nova janela e retorna a data escolhida para um campo texto.

@param componente onde será colocada a data.

@param nome do botão que chama o calendário.

@return data no formato dd/mm/aaaa.

2.11.2 Exemplos de uso.

- ```
<input type="text" id="date2" name="date2" value="" size=25
style="font-weight:normal; Color:#000000" >
<input type="button" name="button" value="..." id="button"
onClick="Calendario(document.getElementById('date2'),'button');" >
```

## 2.12 windowopen.js

Este arquivo contém funções para abrir uma nova janela.

### 2.12.1 Função: window\_open(url, nomeJanela, recursos)

Descrição: Esta função cria uma nova janela.

@param endereço da página a ser aberta.

@param nome para a janela a ser aberta.

@param características da janela.

@return cria uma nova janela com a página indicada.

### 2.12.2 Exemplo de uso.

- ```
<a href="#" onClick="window_open('indice.jsp','help',
'scrollbars=yes,resizable=yes,width=500,height=500')">ajuda</a>
```

2.13 maisinfo.js

Este arquivo contém funções para criar caixa de texto com informações adicionais ao usuário.

2.13.1 Função: maisinfo(msg)

Descrição: Esta função cria uma caixa de mensagem para fornecer mais informações ao usuário.

@param string com a mensagem para o usuário.

@return caixa de mensagem com informação ao usuário.

2.13.2 Função: menosinfo()

Descrição: Esta função apaga a caixa de mensagem da tela.

@param não tem;

@return apaga a caixa de mensagem existente.

2.13.3 Exemplo de uso.

- ``

2.14 dtree.js

Este arquivo contém funções para criar um menu do tipo árvore.

2.14.1 Função: dTree(obj)

Descrição: Esta função cria o menu que será apresentado em formato de árvore.

@param nome do menu.

@return menu com todas os itens acrescentadas pelo comando add.

2.14.2 Comando: add(id, pid, name, url, title, target, icon, iconOpen, open)

Descrição: Este comando acrescenta um item no menu.

@param número de identificação do item.

@param número do item pai ao qual está ligado.

@param descrição (nome) do item.

@param endereço que aparecerá ao clicar no item (**opcional**).

Demais parâmetros opcionais.

@return acrescenta opção no menu.

2.14.3 Comando: document.write(menu)

Descrição: Este comando mostra na página o item de menu criado.

@param nome do menu.

@return menu na página.

2.14.4 Exemplo de uso.

O código apresentado abaixo deverá ser colocado na página onde deseja que apareça o menu:

```
<script type="text/javascript">
  submenu0 = new dTree('submenu0');
```

```

submenu0.add(0,-1,'');
submenu0.add(1,0,'Recurso','');
    submenu0.add(2,1,'Fonte de Recurso',
'/apg2/tabelas/fonteRecurso/frm_inc.jsp');
    submenu0.add(3,1,'Recurso','/apg2/tabelas/recurso/frm_inc.jsp');
submenu0.add(4,0,'Entidade','');

submenu0.add(5,4,'Entidade','/apg2/tabelas/entidade/frm_inc.jsp');
    submenu0.add(6,4,'Tipo de Participação',
'/apg2/tabelas/tipoParticipacao/frm_inc.jsp');
submenu0.add(7,0,'Localização','');
    submenu0.add(8,7,'Grupo de Local',
'/apg2/tabelas/localGrupo/frm_inc.jsp');
    submenu0.add(9,7,'Hierarquia de Grupo',
'/apg2/tabelas/hierarquiaLocalGrupo/frm_inc.jsp');
    submenu0.add(10,7,'Local','/apg2/tabelas/localItem/frm_inc.jsp');
    submenu0.add(11,7,'Hierarquia de Local',
'/apg2/tabelas/hierarquiaLocalItem/frm_inc.jsp');
submenu0.add(12,0,'Critério','/apg2/tabelas/criterio/frm_inc.jsp');
submenu0.add(13,0,'Beneficiário',
'/apg2/tabelas/beneficiario/frm_inc.jsp');
document.write(submenu0);
</script>

```

2.15 coolmenus4.js

Este arquivo contém funções para criar um menu.

2.15.1 Função: makeMenu

Descrição: Esta função cria as opções do menu que será apresentado.

@param número de identificação da opção.

@param número da opção pai ao qual está ligada.

@param descrição da opção.

@param endereço que aparecerá ao clicar na opção (**opcional**).

Demais parâmetros são opcionais.

@return acrescenta opção no menu.

2.15.2 Exemplo de uso.

O código apresentado a abaixo deverá ser colocado na página onde deseja que apareça o menu:

```

<script>
oCMenu.makeMenu('top0','','Cadastros','','')
oCMenu.makeMenu('sub00','top0','Básicos','teste.html')
oCMenu.makeMenu('sub000','sub00','Sistema','teste.html')
oCMenu.makeMenu('sub001','sub00','Grupo','teste.html')
oCMenu.makeMenu('sub002','sub00','Função','teste.html')
oCMenu.makeMenu('sub003','sub00','Usuário','teste.html')
oCMenu.makeMenu('sub01','top0','Complementares','teste.html')

```

```

        oCMenu.makeMenu('sub010','sub01','Ícones do Sistema',
'teste.html')
        oCMenu.makeMenu('sub011','sub01','Operações do Sistema',
'teste.html')
        oCMenu.makeMenu('sub02','top0','Alteração de Senha','teste.html')
        oCMenu.makeMenu('sub020','sub02','Com Confirmação de
Dados', 'teste.html')
        oCMenu.makeMenu('sub021','sub02','Sem Confirmação de
Dados', 'teste.html')
        oCMenu.makeMenu('sub03','top0','Teste','')

oCMenu.makeMenu('sub030','sub03','inicial','pages/inicial.jsp')
        oCMenu.makeMenu('sub031','sub03','form
lista','pages/form.jsp')
        oCMenu.makeMenu('sub032','sub03','msg aviso',
'pages/form_aviso.jsp')
        oCMenu.makeMenu('sub033','sub03','msg erro',
'pages/form_erro.jsp')
        oCMenu.makeMenu('sub034','sub03','msg sucesso',
'pages/form_sucesso.jsp')
        oCMenu.makeMenu('sub035','sub03','abas','pages/abas.jsp')

oCMenu.makeMenu('top1','','Controle','','')
        oCMenu.makeMenu('sub10','top1','Operação do
Sistema','teste.html')
        oCMenu.makeMenu('sub100','sub10','Mensagem de Entrada',
'teste.html')
        oCMenu.makeMenu('sub101','sub10','Registro de Funções
Exec.', 'teste.html')
        oCMenu.makeMenu('sub11','top1','Bloqueios','teste.html')
        oCMenu.makeMenu('sub110','sub11','Grupo','teste.html')
        oCMenu.makeMenu('sub111','sub11','Função','teste.html')
        oCMenu.makeMenu('sub112','sub11','Usuário','teste.html')
        oCMenu.makeMenu('sub12','top1','Restrições','teste.html')
        oCMenu.makeMenu('sub120','sub12','Grupo','teste.html')
        oCMenu.makeMenu('sub121','sub12','Função','teste.html')
        oCMenu.makeMenu('sub122','sub12','Gerais','teste.html')

oCMenu.makeMenu('top2','','Direito','','')
        oCMenu.makeMenu('sub20','top2','Grupo','teste.html')
        oCMenu.makeMenu('sub200','sub20','Acesso ao Sistema',
'teste.html')
        oCMenu.makeMenu('sub201','sub20','Acesso a Função',
'teste.html')
        oCMenu.makeMenu('sub21','top2','Usuário','teste.html')
        oCMenu.makeMenu('sub210','sub21','Acesso ao Grupo',
'teste.html')
        oCMenu.makeMenu('sub211','sub21','Acesso a Função',
'teste.html')
        oCMenu.makeMenu('sub212','sub21','Acesso ao Sistema',
'teste.html')

oCMenu.makeMenu('top3','','Consultas','','')
        oCMenu.makeMenu('sub30','top3','Consulta','teste.html')

oCMenu.makeMenu('top4','','Operações','','')
        oCMenu.makeMenu('sub40','top4','Transferir Desen. - Prod.',

```

```

        'teste.html')

oCMenu.makeMenu('top5','','Sair','teste.html','')
//esta instrucao vai construir o menu
oCMenu.construct()
</script >

```

2.16 calendar.js

Este arquivo contém funções para criar um calendário na página corrente.

2.16.1 Função: addLink(mes, dia, link)

Descrição: Esta função adiciona link nas datas informadas.

@param número referente ao mês – 0=janeiro...

@param número referente ao dia.

@param endereço do link.

@return acrescenta acrescenta link na data informada.

2.16.2 Função: montaCalendario(mes, ano)

Descrição: Gera um calendário na tela corrente a partir do parâmetros informados.

@param número referente ao mês – 0=janeiro...

@param número referente ao ano.

@return gera o calendário.

2.16.3 Exemplo de uso

- `<input type="button" value="Ok" onMouseDown="javascript:montaCalendario (document.formcalend.Mes.value,document.formcalend.Ano.value)" >`

2.17 prototype.js

Detalhes a respeito deste arquivo podem ser encontrados no documento PADRÃO DE UTILIZAÇÃO DE AJAX no capítulo ANEXOS. Este documento pode ser acessado através do site <http://www.frameworkpinhao.pr.gov.br> no menu

Metodologia : Fases : Construção : Material de Apoio : Padrões : AJAX.