
PHP-Core

Versão v0.6.2-beta

01 jan., 2021

1	Instalação	3
1.1	Constantes e Funções Globais	3
1.1.1	Constantes	3
1.1.2	Array	4
1.1.2.1	array_check_required_keys	4
1.1.2.2	array_in_ci	4
1.1.2.3	array_is_assoc	4
1.1.3	Date	5
1.1.3.1	date_to_first_month_day	5
1.1.3.2	date_to_last_month_day	5
1.1.4	Diretórios e Arquivos	5
1.1.4.1	dir_chmod_r	5
1.1.4.2	dir_convert_to_utf8	6
1.1.4.3	dir_copy	6
1.1.4.4	dir_deltree	6
1.1.4.5	dir_scan_w	7
1.1.4.6	file_convert_to_utf8	7
1.1.5	Strings	7
1.1.5.1	mb_str_contains	7
1.1.5.2	mb_str_ends_with	7
1.1.5.3	mb_str_insert	8
1.1.5.4	mb_str_last_pos	8
1.1.5.5	mb_str_limit_chars	8
1.1.5.6	mb_str_limit_words	9
1.1.5.7	mb_str_pattern_match	9
1.1.5.8	mb_str_pos_all	9
1.1.5.9	mb_str_preserve_chars	9
1.1.5.10	mb_str_remove	10
1.1.5.11	mb_str_remove_chars	10
1.1.5.12	mb_str_remove_glyphs	10
1.1.5.13	mb_str_remove_invisible_chars	10
1.1.5.14	mb_str_replace_once	11
1.1.5.15	mb_str_starts_with	11
1.1.5.16	mb_str_to_valid_filename	11
1.1.5.17	mb_str_uc_first	12
1.1.5.18	mb_str_uc_names	12

1.1.5.19	mb_str_uc_names_ptbr	12
1.1.5.20	mb_str_uc_words	13
1.1.5.21	mb_substr_replace	13
1.1.6	Números	13
1.1.6.1	numeric_decimal_part	13
1.1.6.2	numeric_integer_part	14
1.1.6.3	numeric_is_even	14
1.1.6.4	numeric_is_odd	14
1.1.6.5	numeric_mod	14
1.1.7	Misc	15
1.1.7.1	html_print	15
1.1.7.2	object_clone	15
1.1.7.3	object_convert_values_to_html_entities	15
1.1.7.4	redirect	16
1.1.7.5	to_system_path	16
1.1.7.6	var_initi_set	16
1.1.7.7	var_is_defined	17
1.1.8	WeekDate	17
1.1.8.1	weekdate_count_weeks	17
1.1.8.2	weekdate_get_first_week	17
1.1.8.3	weekdate_get_last_week	18
1.1.8.4	weekdate_to_array	18
1.1.8.5	weekdate_to_datetime	18
2	Namespaces	19
2.1	AeonDigital	19
2.1.1	Namespaces	19
2.1.1.1	Collection	19
2.1.1.1.1	Classes	19
2.1.1.1.1.1	BasicCollection	19
2.1.1.1.1.2	Properties	20
2.1.1.1.1.3	Methods	20
2.1.1.1.1.4	Collection	24
2.1.1.1.1.5	Methods	24
2.1.1.1.1.6	TypeList	26
2.1.1.1.1.7	Properties	26
2.1.1.1.1.8	Methods	26
2.1.1.2	SimpleType	27
2.1.1.2.1	Namespaces	27
2.1.1.2.1.1	Abstracts	27
2.1.1.2.1.2	Classes	27
2.1.1.2.1.3	aBasic	27
2.1.1.2.1.4	Methods	27
2.1.1.2.1.5	aFloat	28
2.1.1.2.1.6	Methods	28
2.1.1.2.1.7	aInt	29
2.1.1.2.1.8	Methods	29
2.1.1.2.2	Classes	30
2.1.1.2.2.1	stBool	30
2.1.1.2.2.2	Methods	30
2.1.1.2.2.3	stByte	31
2.1.1.2.2.4	Methods	31
2.1.1.2.2.5	stDateTime	31
2.1.1.2.2.6	Methods	32

2.1.1.2.2.7	stDouble	33
2.1.1.2.2.8	Methods	33
2.1.1.2.2.9	stFloat	33
2.1.1.2.2.10	Methods	33
2.1.1.2.2.11	stInt	34
2.1.1.2.2.12	Methods	34
2.1.1.2.2.13	stLong	34
2.1.1.2.2.14	Methods	35
2.1.1.2.2.15	stReal	35
2.1.1.2.2.16	Methods	35
2.1.1.2.2.17	stShort	36
2.1.1.2.2.18	Methods	36
2.1.1.2.2.19	stString	36
2.1.1.2.2.20	Methods	37
2.1.1.3	Tools	37
2.1.1.3.1	Classes	37
2.1.1.3.1.1	Image	37
2.1.1.3.1.2	Methods	38
2.1.1.3.1.3	JSON	40
2.1.1.3.1.4	Methods	40
2.1.1.3.1.5	MinifyCSS	41
2.1.1.3.1.6	Methods	41
2.1.1.3.1.7	MinifyJS	42
2.1.1.3.1.8	Methods	42
2.1.1.3.1.9	Zip	43
2.1.1.3.1.10	Methods	43
2.1.1.4	Traits	44
2.1.1.4.1	Traits	44
2.1.1.4.1.1	MainCheckArgumentException	44
2.1.1.4.1.2	Properties	44
2.1.1.4.1.3	Methods	44
2.1.2	Classes	44
2.1.2.1	BObject	44
2.1.2.1.1	Methods	45
2.1.2.2	Realtype	45
2.1.2.2.1	Properties	45
2.1.2.2.2	Methods	45
2.1.2.3	Tools	52
2.1.2.3.1	Methods	53

PHP Namespace Index	59
----------------------------	-----------

Índice	61
---------------	-----------

PHP-Core

Constantes, Funções Globais e Classes de uso geral.

Instale em seu projeto usando o composer:

Via terminal

```
composer require aeondigital/phpcore
```

Via composer.json

```
"require": {  
    "aeondigital/phpcore": "dev-master"  
}
```

1.1 Constantes e Funções Globais

1.1.1 Constantes

As seguintes constantes serão disponibilizadas em todos os seus projetos sempre que esta biblioteca estiver presente:

constant undefined

Valor de um objeto não definido.

Há momentos onde se deseja que uma variável ou propriedade esteja com o status de **não definida**, e, para estes casos tal constante deve ser usada.

Importante notar que se for verificado o tipo, será retornado **string** portanto é importante que seu uso seja feito de forma controlada.

Sua grafia está especialmente escrita em **lowercase** pois, por sua concepção, estima-se que, se estivesse no core do PHP, seu nível hierárquico seria equivalente à `null` o que faria ela entrar nas recomendações PSR junto com as mesmas regras que definem que `null`, `true` e `false` devem ser escritas em **lowercase**.

constant DS

Separador de diretório conforme o S/O.

Apenas uma forma menor para se referir à constante `DIRECTORY_SEPARATOR`.

1.1.2 Array

1.1.2.1 `array_check_required_keys`

`array_check_required_keys` (*\$keys*, *\$array*)

Verifica se as chaves definidas como obrigatórias de um `Array Associativo` estão realmente presentes.

Parameters

- `< array >` **\$keys** Coleção com o nome das chaves obrigatórias.
- `< array >` **\$array** `Array associativo` que será verificado.

Returns `< array >` Retorna um `array` contendo o nome de cada um dos itens que **NÃO** foram definidos. Ou seja, se retornar um `array` vazio, significa que todas as chaves foram definidas.

1.1.2.2 `array_in_ci`

`array_in_ci` (*\$needle*, *\$haystack*)

Versão `case-insensitive` para o método `in_array()`.

Parameters

- `< string >` **\$needle** Valor que será procurado.
- `< array >` **\$haystack** `Array` onde o valor será procurado.

Returns `< bool >` Retornará `true` se o valor de `$needle` for encontrado em um dos valores contidos no `array $haystack`.

1.1.2.3 `array_is_assoc`

`array_is_assoc` (*\$o*)

Verifica se o objeto passado é um `Array Associativo`.

Parameters

- < mixed > **\$o** Objeto que será testado.

Returns < bool > Retornará `true` se `$o` for mesmo um `Array Associativo`. Um array vazio retornará `false`.

1.1.3 Date

1.1.3.1 date_to_first_month_day

date_to_first_month_day (*\$date*)

Retorna um objeto `DateTime` setado para o primeiro dia do mês. Hora minuto e segundo serão definidos como **00:00:00**.

Parameters

- < DateTime > **\$date** Data original.

Returns < \DateTime >

1.1.3.2 date_to_last_month_day

date_to_last_month_day (*\$date*)

Retorna um objeto `DateTime` setado para o último dia do mês. Hora minuto e segundo serão definidos como **23:59:59**.

Parameters

- < DateTime > **\$date** Data original.

Returns < \DateTime >

1.1.4 Diretórios e Arquivos

1.1.4.1 dir_chmod_r

dir_chmod_r (*\$absoluteSystemPathToDir*, *\$permissions*)

Efetua alteração nas permissões de um diretório e em todos seus itens filhos.

Parameters

- < string > **\$absoluteSystemPathToDir** Diretório que terá suas alterações alteradas.
- < int > **\$permissions** As permissões que serão setadas.

Returns < bool > Retornará `true` se **TODOS** os itens alvo tiverem suas permissões alteradas.

1.1.4.2 `dir_convert_to_utf8`

`dir_convert_to_utf8` (*\$absoluteSystemPaths*, *\$validExtensions*)

Converte todos os arquivos alvo para o encode **UTF-8**.

Se algum diretório for alvo desta ação, todos os seus arquivos filhos (incluindo subdiretórios) serão também convertidos.

Ocorrendo qualquer falha durante o processamento das conversões, o processamento parará imediatamente.

Parameters

- < array > **`$absoluteSystemPaths`** Caminhos para os recursos que serão convertidos. Podem ser apontados diretórios ou arquivos.
- < array > **`$validExtensions`** Coleção de extensões válidas para executar a conversão.

Returns < bool > Retornará `true` se **TODOS** os recursos indicados em `$absoluteSystemPaths` existirem e forem corretamente convertidos.

1.1.4.3 `dir_copy`

`dir_copy` (*\$absoluteSystemPathToDir_source*, *\$absoluteSystemPathToDir_dest*, *\$permissions=700*)

Copia todo o conteúdo de um diretório para outro local.

Parameters

- < string > **`$absoluteSystemPathToDir_source`** Caminho para o diretório de conteúdo que será copiado.
- < string > **`$absoluteSystemPathToDir_dest`** Caminho para o diretório de destino.
- < int > **`$permissions`** As permissões que devem ser setadas no novo diretório.

Returns < bool > Retornará `true` se a cópia do conteúdo ocorrer sem erros.

1.1.4.4 `dir_deltree`

`dir_deltree` (*\$absoluteSystemPathToDir*)

Remove um diretório e todo seu conteúdo.

Parameters

- < string > **`$absoluteSystemPathToDir`** Diretório que será excluído.

Returns < bool > Retornará `true` se o diretório alvo for excluído.

1.1.4.5 `dir_scan_w`

`dir_scan_w` (*\$absoluteSystemPathToDir*)

Retorna a listagem do conteúdo do diretório alvo já ordenado adequadamente conforme o padrão Windows.

Parameters

- < string > **`$absoluteSystemPathToDir`** Diretório que será listado.

Returns < array > Lista de diretórios e arquivos encontrados no local indicado.

1.1.4.6 `file_convert_to_utf8`

`file_convert_to_utf8` (*\$absoluteSystemPathToFile*)

Converte o encode de um arquivo para **UTF-8**.

O processo consiste em resgatar todo o conteúdo e verificar caracter à caracter convertendo aqueles que não forem unicode e então remontando todo o arquivo.

Parameters

- < string > **`$absoluteSystemPathToFile`** Caminho para o arquivo que será convertido.

Returns < bool > Retornará `true` caso o documento seja convertido sem falhas.

1.1.5 Strings

1.1.5.1 `mb_str_contains`

`mb_str_contains` (*\$str*, *\$search*)

Indica se o valor de entrada possui *\$search* em alguma posição.

Parameters

- < string > **`$str`** Valor de entrada.
- < string > **`$search`** Valor a ser procurado.

Returns < bool >

1.1.5.2 `mb_str_ends_with`

`mb_str_ends_with` (*\$haystack*, *\$needle*)

Identifica se a *string* passada termina com a sentença indicada.

Esta função é multi-byte.

Parameters

- < string > **\$haystack** String original.
- < string > **\$needle** Caracteres finais.

Returns < bool > Retornará `true` se `$haystack` é uma `string` que termina com `$needle`.

1.1.5.3 mb_str_insert

mb_str_insert (*\$str*, *\$insert*, *\$pos*)

Insere a nova `string` na posição indicada em `$pos`.

Parameters

- < string > **\$str** Valor de entrada.
- < string > **\$insert** String que será adicionada.
- < int > **\$pos** Posição em que a `string` será adicionada. Se negativo, irá adicionar a nova `string` na respectiva posição contando a partir do final da `string` original.

Returns < string >

1.1.5.4 mb_str_last_pos

mb_str_last_pos (*\$str*, *\$search*, *\$start=0*)

Encontra a última ocorrência de `$search` em `$str`.

Parameters

- < string > **\$str** Valor de entrada.
- < string > **\$search** Valor que deve ser procurado.
- < int > **\$start** Posição a partir da qual a pesquisa deve ser feita.

Returns < false | int > Retornará `false` caso a `string` não exista.

1.1.5.5 mb_str_limit_chars

mb_str_limit_chars (*\$str*, *\$max*, *\$etc=""*;))

Limita uma sentença à um número máximo de caracteres.

Parameters

- < string > **\$str** String original.
- < int > **\$max** Número máximo de caracteres.
- < ?string > **\$etc** Será adicionado ao final da sentença, se, o número total de caracteres iniciais for maior que `$max`.

Returns < string >

1.1.5.6 mb_str_limit_words

mb_str_limit_words (*\$str*, *\$max*, *\$etc=""*;)

Limita a sentença à um número máximo de palavras.

Parameters

- < string > **\$str** String original.
- < int > **\$max** Número máximo de palavras.
- < ?string > **\$etc** Será adicionado ao final da sentença, se, o número total de palavras iniciais for maior que \$max.

Returns < string >

1.1.5.7 mb_str_pattern_match

mb_str_pattern_match (*\$str*, *\$regex*)

Verifica se a string corresponde à expressão regular definida.

Parameters

- < string > **\$regex** RegExp que será usada para validar a string.

Returns < bool >

1.1.5.8 mb_str_pos_all

mb_str_pos_all (*\$haystack*, *\$needle*)

Encontra todas as ocorrências de uma *string* e retorna um *array* com as respectivas posições.

Parameters

- < string > **\$needle** Valor que será procurado.

Returns < false | array > Retornará *false* caso nenhuma ocorrência seja encontrada ou um *array* contendo a posição de cada ocorrência de *\$needle* encontrada.

1.1.5.9 mb_str_preserve_chars

mb_str_preserve_chars (*\$str*, *\$preserve*)

Mantém apenas os caracteres que estão na coleção de válidos.

Parameters

- < string > **\$str** String original.
- < string > **\$preserve** Coleção de caracteres que serão preservados.

Returns < string >

1.1.5.10 mb_str_remove

mb_str_remove (*\$str*, *\$start*, *\$length=null*)

Remove uma cadeia de caracteres dentro dos limites indicados.

Parameters

- < string > **\$str** String original.
- < int > **\$start** Posição inicial para inserir a nova string.
- < ?int > **\$length** Tamanho da porção da string original que será substituída.

Returns < string >

1.1.5.11 mb_str_remove_chars

mb_str_remove_chars (*\$str*, *\$remove*)

Remove da string todas as ocorrências da cadeia de caracteres informado.

Parameters

- < string > **\$str** String original.
- < string > **\$remove** Coleção de caracteres que serão excluídos..

Returns < string >

1.1.5.12 mb_str_remove_glyphs

mb_str_remove_glyphs (*\$str*)

Transforma todos caracteres com glifos para seu equivalente sem glifos.

Parameters

- < string > **\$str** String original.

Returns < string >

1.1.5.13 mb_str_remove_invisible_chars

mb_str_remove_invisible_chars (*\$str*)

Remove da string todas as ocorrências de caracteres não visíveis.

Parameters

- `< string > $str` String original.

Returns < string >

1.1.5.14 mb_str_replace_once

mb_str_replace_once (*\$search*, *\$replace*, *\$subject*)

Substitui apenas a primeira ocorrência de `$search` em `$subject` pelo valor correspondente em `$replace`.

Parameters

- `< string > $search` Valor que será substituído na `string`.
- `< string > $replace` Valor que será adicionado no lugar de `$search`.
- `< string > $subject` Valor original que será substituído.

Returns < string > Nova string modificada.

1.1.5.15 mb_str_starts_with

```
mb_str_starts_with($haystack, $needle)
```

Identifica se a `string` passada inicia com a sentença indicada.

Parameters

- `< string > $haystack` String original.
- `< string > $needle` Caracteres iniciais.

Returns <bool> Retornará true se \$haystack é uma string que inicia com \$needle.

1.1.5.16 mb_str_to_valid_filename

```
mb_str_to_valid_filename($str)
```

Verifica a string original passada e remove dela qualquer caracter que a torne um nome de arquivo invalido (tanto em sistemas Unix quando em Windows).

Espaços serão removidos do início e do final da string.

Caracteres não visíveis serão excluídos.

Caracteres com glifos serão substituídos por suas versões simples.

Os caracteres "\", "/", "*", "?", "<", ">", "|", " ", " ", ":", " " serão convertidos em "_".

Espaços serão convertidos em "_".

Parameters

- `<string> $str` String original.

Returns < string >

1.1.5.17 mb_str_uc_first

mb_str_uc_first (*\$string*)

Converte para maiúscula o primeiro caractere de uma *string*.

Este método é equivalente ao `ucfirst()` porém, suporta multi-byte.

Parameters

- <string> **\$string** String que será alterada.

Returns <string> Nova *string* modificada.

1.1.5.18 mb_str_uc_names

mb_str_uc_names (*\$str*, *\$locale*="", *\$ignore*=[])

Converte para maiúsculas o primeiro caractere de cada parte de uma *string* que representa um nome próprio. Todos os demais caracteres da *string* passada serão revertidos para minúsculas.

Parameters

- <string> **\$str** String que será alterada.
- <string> **\$locale** Locale que deve ser usado.
- <array> **\$ignore** Se definido, deve ser um array de *strings* contendo palavras que devem escapar da transformação.

Returns <string> Nova *string* modificada.

1.1.5.19 mb_str_uc_names_ptbr

mb_str_uc_names_ptbr (*\$string*)

Converte para maiúsculas o primeiro caractere de cada parte de uma *string* que representa um nome próprio. Todos os demais caracteres da *string* passada serão revertidos para minúsculas.

Artigos usados nos nomes próprios típicos do locale **pt-BR** serão protegidos de terem seus caracteres alterados.

Parameters

- <string> **\$string** String que será alterada.

Returns <string> Nova *string* modificada.

1.1.5.20 mb_str_uc_words

mb_str_uc_words (*\$string*, *\$ignore=[]*)

Converte para maiúsculas o primeiro caractere de cada palavra.

Este método é equivalente ao `ucwords()` porém, suporta multi-byte.

Parameters

- <string> **\$string** String que será alterada.
- <array> **\$ignore** Se definido, deve ser um array de strings contendo palavras que devem escapar da transformação.

Returns <string> Nova string modificada.

1.1.5.21 mb_substr_replace

mb_substr_replace (*\$string*, *\$replacement*, *\$start*, *\$length=null*, *\$encoding=null*)

Substitui o texto em uma parte da string por outro.

Este método é equivalente ao `substr_replace()` porém, suporta multi-byte.

Parameters

- <string> **\$string** String original.
- <string> **\$replacement** String que será adicionada.
- <int> **\$start** Posição inicial para inserir a nova string.
- <?int> **\$length** Tamanho da porção da string original que será substituída.
- <?string> **\$encoding** Quando usado indica que codificação a string original está usando.

Returns <string> Nova string com o novo valor.

1.1.6 Números

1.1.6.1 numeric_decimal_part

numeric_decimal_part (*\$n*, *\$l=2*)

Retorna unicamente a parte decimal de um numeral.

Por questões internas referentes a forma como os numerais de ponto flutuantes funcionam, a maior precisão possível de ser encontrada é a de números de até 15 dígitos, independente do local onde está o ponto decimal.

Parameters

- < int | float > **\$n** Valor numérico de entrada.
- < int > **\$l** Tamanho da parte decimal a ser retornada. Se não for informado, será usado o valor 2.

Returns < ?float > Retornará `null` caso o valor de entrada não seja numérico.

1.1.6.2 numeric_integer_part

numeric_integer_part (*\$n*)

Retorna unicamente a parte inteira de um numeral.

Parameters

- < int | float > **\$n** Valor numérico de entrada.

Returns < ?int > Retornará `null` caso o valor de entrada não seja numérico.

1.1.6.3 numeric_is_even

numeric_is_even (*\$n*)

Identifica se o valor é par.

Parameters

- < int | float > **\$n** Valor numérico de entrada.

Returns < ?bool > Retornará `null` caso o valor indicado não seja numérico.

1.1.6.4 numeric_is_odd

numeric_is_odd (*\$n*)

Identifica se o valor é ímpar.

Parameters

- < int | float > **\$n** Valor numérico de entrada.

Returns < ?bool > Retornará `null` caso o valor indicado não seja numérico.

1.1.6.5 numeric_mod

numeric_mod (*\$n*, *\$div*)

Retorna o resto da divisão entre o valor atual e o divisor passado.

Parameters

- < int | float > **\$n** Valor numérico de entrada.

- < int | float > **\$div** Divisor.

Returns < ?int > Retornará `null` caso um dos valores passados não seja numérico.

1.1.7 Misc

1.1.7.1 `html_print`

`html_print` (*\$obj*, *\$o=null*, *\$w=null*, *\$h=null*)

Imprime na tela o valor de *\$obj* dentro de uma tag `<pre>`, facilitando assim a leitura quando necessário o debug.

Parameters

- < mixed > **\$obj** Objeto que será **impresso**.
- < bool > **\$o** Indica se a tag `<pre>` deve receber a propriedade css `overflow:auto`.
- < string > **\$w** Valor da propriedade css `width` para definir a largura do objeto `<pre>`.
- < string > **\$h** Valor da propriedade css `height` para definir a altura do objeto `<pre>`.

Returns < void >

1.1.7.2 `object_clone`

`object_clone` (*\$obj*)

Efetua a clonagem de um objeto.

Parameters

- < mixed > **\$obj** Objeto que será clonado.

Returns < mixed >

1.1.7.3 `object_convert_values_to_html_entities`

`object_convert_values_to_html_entities` (*\$o*)

Converte todo conteúdo de um `Array Associativo` ou objeto `\stdClass` em um valor do tipo `string` que pode ser apresentado normalmente em um documento HTML.

Parameters

- < mixed > **\$o** Objeto original, que será convertido.

Returns < mixed > Retornará o mesmo tipo de objeto inicialmente passado em *\$o*, mas com todos seus valores convertidos para `strings` representáveis em um documento HTML.

1.1.7.4 redirect

redirect (*\$url*, *\$statusCode="HTTP/1.1 302 Found"*;)

Redireciona o UA para a URL indicada.

Esta ação interrompe o script imediatamente após o redirecionamento.

Parameters

- < string > **\$url** URL para onde o UA será redirecionado.
- < string > **\$statusCode** Status `Http` que deve ser informado no header. O status padrão **HTTP/1.1 302 Found** indica que o redirecionamento é esperado ou que faz parte do fluxo padrão da aplicação.

Returns < void >

1.1.7.5 to_system_path

to_system_path (*\$systemPath*)

Corrige um caminho para um diretório ou arquivo interno ajustando os separadores de diretório e eliminando duplicação dos mesmos. Qualquer `\\` ou `/` no final do caminho será removida.

Parameters

- < ?string > **\$systemPath** Caminho que será ajustado.

Returns < ?string > Caminho corrigido.

1.1.7.6 var_initi_set

var_initi_set (&*\$o*, *\$d=null*, *\$u=false*)

Se a variável indicada não estiver definida, irá iniciá-la com o valor padrão passado.

Parameters

- < mixed > **\$o** Objeto a ser iniciado.
- < mixed > **\$d** Valor padrão a ser definido.
- < bool > **\$u** Se `true` testa apenas se o valor é `undefined`. Se `false` testa usando `var_is_defined()`.

Returns < void >

1.1.7.7 var_is_defined

var_is_defined (&\$o, \$k=null)

Verifica se a variável indicada está definida.

Uma variável é considerada definida quando NÃO FOR um array vazio ou um objeto stdClass vazio, ou, quando tratar-se de um tipo escalar, for diferente de null, undefined e ''.

Parameters

- < mixed > **\$o** Objeto que será testado.
- < ?string > **\$k** Quando indicado, \$o deve ser um array ou um objeto stdClass e \$k será a chave cuja existência e valor será verificado.

Returns < bool > Retornará true SE a variável ESTA definida E SE seu valor é diferente de null, undefined e ''. Objetos do tipo array e stdClass retornarão true SE não forem vazios quando \$k não for definido.

1.1.8 WeekDate

1.1.8.1 weekdate_count_weeks

weekdate_count_weeks (\$year)

Calcula a quantidade de semanas que o ano informado tem.

Parameters

- < int > **\$year** Ano.

Returns < int >

1.1.8.2 weekdate_get_first_week

weekdate_get_first_week (\$year)

Retorna um objeto DateTime referente ao primeiro dia da primeira semana do ano informado.

Parameters

- < int > **\$year** Ano.

Returns < \DateTime >

1.1.8.3 `weekdate_get_last_week`

`weekdate_get_last_week` (*\$year*)

Retorna um objeto `DateTime` referente ao último dia da última semana do ano informado.

Parameters

- `< int > $year` Ano.

Returns `< \DateTime >`

1.1.8.4 `weekdate_to_array`

`weekdate_to_array` (*\$week*)

A partir de uma `string` de data no formato **week** ou de um objeto `\DateTime` retorna um `array` associativo com os dados brutos da data especificada.

O array terá as seguintes chaves:

```
““ php
    $arr = [
        "year" => int,
        "week" => int,
        "day" => int
    ];
““
```

Parameters

- `< ?string | \DateTime > $week` String ou objeto `DateTime`.

Returns `< ?array >` Retornará `null` se não for possível identificar os componentes da data.

1.1.8.5 `weekdate_to_datetime`

`weekdate_to_datetime` (*\$week*)

Retorna um `DateTime` conforme parametros passados.

Parameters

Returns `< ?\DateTime >`

2.1 AeonDigital

AeonDigital

2.1.1 Namespaces

2.1.1.1 Collection

AeonDigitalCollection

2.1.1.1.1 Classes

2.1.1.1.1.1 BasicCollection

class AeonDigital\Collection\BasicCollection

Implementa a interface iBasicCollection.

Esta classe traz componentes que permitem implementar quaisquer das interfaces do namespace AeonDigital\Interfaces\Collection no entanto APENAS iBasicCollection está devidamente implementada, restando às classes concretas que herdem esta se especializarem em cada uma destas capacidades.

A partir desta classe as seguintes interfaces podem ser implementadas imediatamente:

- iProtectedCollection
- iAppendOnlyCollection
- iReadOnlyCollection
- iCaseInsensitiveCollection

Na documentação de cada método ou propriedade desta classe vem especificado de que forma a implementação de uma destas interfaces deve alterar o comportamento do mesmo.

Parent *AeonDigital\BObject*

Implements *AeonDigital\Interfaces\Collection\iBasicCollection*

2.1.1.1.1.2 Properties

2.1.1.1.1.3 Methods

public *AeonDigital\Collection\BasicCollection::isProtected()*

Indica se a coleção implementa a interface *iProtectedCollection*.

Returns < bool > Quando `true` indica que a coleção manterá o estado de todos os seus objetos não permitindo que eles sejam alterados, no entanto os valores uma vez definidos PODEM ser excluídos.

public *AeonDigital\Collection\BasicCollection::isAppendOnly()*

Indica se a coleção implementa a interface *iAppendOnlyCollection*.

Returns < bool > Quando `true` indica que a coleção pode ser apenas incrementada mas jamais modificada nem reduzida.

public *AeonDigital\Collection\BasicCollection::isReadOnly()*

Indica se a coleção implementa a interface *iReadOnlyCollection*.

Returns < bool > Quando `true` indica que a coleção não pode ser alterada após ser definida durante a construção da instância.

public *AeonDigital\Collection\BasicCollection::isCaseInsensitive()*

Indica se a coleção implementa a interface *iCaseInsensitiveCollection*.

Returns < bool > Quando `true` indica que os nomes das chaves de cada entrada de dados será tratado de forma `case insensitive`, ou seja, `KeyName = keyname = KEYNAME`.

public *AeonDigital\Collection\BasicCollection::isAutoIncrement()*

Uma instância com a característica `autoincrement` deve permitir que seja omitido o nome das chaves no método `set` pois este deve ser controlado internamente como se fosse um

array iniciado em zero.

Ainda assim o tratamento das chaves sempre se dará como se fossem `strings` e não numerais inteiros como ocorre em um array comum.

As implementações desta característica devem também controlar os índices quando estes são removidos. A regra geral é que TODOS os itens existentes mantenham como chave o índice correspondente a sua real posição.

```

““ php
    // Neste caso uma coleção com 10 itens que execute 5 vezes a instrução:
    $collection->remove("0");
    // Ficaré, ao final com 5 itens cada qual ocupando uma posição entre 0 e 4.
““

```

Returns < bool > Retorna `true` quando a coleção é do tipo `autoincrement`.

```
public AeonDigital\Collection\BasicCollection::has($key)
```

Indica se a chave de nome indicado existe entre os itens da coleção.

Parameters

- < string > **\$key** Nome da chave que será identificada.

Returns < bool > Retorna `true` caso a chave indicada existir entre os itens da coleção ou `false` se não existir.

```
public AeonDigital\Collection\BasicCollection::set($key, $value)
```

Insere um novo item chave/valor para a coleção de dados.

Se já existe um valor com chave de mesmo nome então, o valor antigo será substituído.

Parameters

- < string > **\$key** Nome da chave. Pode ser usado `'` caso a instância seja do tipo `autoincrement`.
- < mixed > **\$value** Valor que será associado a esta chave.

Returns < bool > Retorna `true` quando a inserção/atualização do item foi bem sucedido.

```
public AeonDigital\Collection\BasicCollection::get($key)
```

Resgata um valor da coleção a partir do nome da chave indicada.

Parameters

- < string > **\$key** Nome da chave cujo valor deve ser retornado.

Returns < ?mixed > Valor armazenado na `collection` que correspondente a chave passada. DEVE retornar `null` quando a chave de nome indicado não existir.

```
public AeonDigital\Collection\BasicCollection::remove($key)
```

Remove da coleção o item com a chave indicada.

Parameters

- < string > **\$key** Nome da chave do valor que será excluído.

Returns < bool > Retornará `true` se a chave foi removida, ou, se, ela não existia dentro da coleção atual e `false` caso por algum motivo não seja possível executar este método.

```
public AeonDigital\Collection\BasicCollection::__construct($initialValues=[],
                                                             $autoincrement=false)
```

Inicia uma nova coleção de dados.

Parameters

- < ?array > **\$initialValues** Valores com os quais a instância deve iniciar.
- < bool > **\$autoincrement** Quando `true` permite que seja omitido o nome da chave dos valores pois eles serão definidos internamente conforme fosse um `array` começando em zero.

Throws < InvalidArgumentException > Caso algum dos valores iniciais a serem definidos não seja aceito.

```
public AeonDigital\Collection\BasicCollection::offsetExists($key)
```

Método que permite a verificação da existência de um valor usando a notação de `array associativo` em conjunto com as funções `isset()` e `empty()` do PHP:

```
““ php
    $oCollect = new iBasicCollection();
    ...
    if (isset($oCollect["keyName"])) { ... }
““
```

Parameters

- < string > **\$key** Chave que será verificada.

Returns < bool >

```
public AeonDigital\Collection\BasicCollection::offsetGet($key)
```

Método que permite resgatar o valor de um item da coleção da instância usando a notação de `array associativo`.

```

““ php
    $oCollect = new iBasicCollection();
    if ($oCollect["keyName"] == $value) { ... }
““

```

Parameters

- < string > **\$key** Nome da chave cujo valor deve ser retornado.

Returns < mixed | null >

public AeonDigital\Collection\BasicCollection::offsetSet (\$key, \$value)

Método que permite inserir um novo valor para a coleção de dados da instância usando a notação de um array associativo.

```

““ php
    $oCollect = new iBasicCollection();
    $oCollect["keyName"] = $value;
““

```

Parameters

- < string > **\$key** Nome da chave.
- < mixed > **\$value** Valor que será associado.

Returns < void >

public AeonDigital\Collection\BasicCollection::offsetUnset (\$key)

Método que permite remover o valor de um item da coleção da instância usando a notação de array associativo em conjunto com a função unset () do PHP:

```

““ php
    $oCollect = new iBasicCollection();
    unset($oCollect["keyName"]);
““

```

Parameters

- < string > **\$key** Nome da chave cujo valor deve ser retornado.

Returns < mixed | null >

public AeonDigital\Collection\BasicCollection::count ()

Método que permite a verificação da quantidade de itens na coleção atual usando a função count () do PHP.

```
““ php
    $oCollect = new iBasicCollection();
    ...
    if (count($oCollect) > 1) { ... }
““
```

Returns < int >

public AeonDigital\Collection\BasicCollection::getIterator()

Método que permite a iteração sobre os valores armazenados na coleção de dados da instância usando `foreach()` do PHP.

```
““ php
    $oCollect = new iBasicCollection();
    ...
    foreach($oCollect as $key => $value) { ... }
““
```

Returns < \Traversable >

public AeonDigital\Collection\BasicCollection::__set(\$name, \$value)

Desabilita a função mágica `__set` para assegurar a que apenas alterações dentro das regras definidas para a classe sejam possíveis.

2.1.1.1.4 Collection

class AeonDigital\Collection\Collection

Classe plena para lidar com coleções de dados.

Implementa `iCollection` que expande as funcionalidades de coleções do tipo `iBasicCollection` permitindo a manipulação dos valores da coleção em processos de lote.

Parent *AeonDigital\Collection\BasicCollection*

Implements *AeonDigital\Interfaces\Collection\iCollection*

2.1.1.1.5 Methods

public AeonDigital\Collection\Collection::toArray(\$originalKeys=false)

Retorna toda a coleção atualmente armazenada em um array associativo

```
[ string => mixed ].
```

Em caso de uma coleção vazia será retornado [].

Prioriza o retorno das chaves conforme usadas internamente pois considera que se há uma alteração nelas deve-se a alguma importância relacionado a seu formato de uso.

Parameters

- < ?bool > **\$originalKeys** Quando `true` irá usar as chaves conforme foram definidas na função `set`. Se no armazenamento interno elas sofrerem qualquer alteração e for definido `false` então elas retornarão seu formato alterado.

Returns < array >

```
public AeonDigital\Collection\Collection::insert($newValues)
```

Permite inserir multiplos dados de uma única vez na coleção.

Parameters

- < array > **\$newValues** array associativo contendo os novos valores a serem definidos para a coleção.

Returns < bool > Retornará `true` caso TODOS os novos valores sejam adicionados e `false` caso 1 deles falhe.

Throws < InvalidArgumentException > DEVE ser lançado caso algum dos valores passados seja undefined.

```
public AeonDigital\Collection\Collection::clean()
```

Limpa totalmente a coleção de dados eliminando toda informação armazenada no momento.

Returns < bool > Retornará `true` caso a exclusão dos dados tenha sido executada com sucesso e `false` caso ocorra algum erro em algum dos itens. Neste caso, a coleção ficará pela metade.

```
public AeonDigital\Collection\Collection::__construct($initialValues=[], $autoincrement=false)
```

Inicia nova coleção de dados.

Parameters

- < ?array > **\$initialValues** Valores com os quais a instância deve iniciar.
- < bool > **\$autoincrement** Quando `true` permite que seja omitido o nome da chave dos valores pois eles serão definidos internamente conforme fosse um array começando em zero.

Throws < InvalidArgumentException > Caso algum dos valores iniciais a serem definidos não seja aceito.

2.1.1.1.1.6 TypeList

class `AeonDigital\Collection\TypeList`

Permite a criação de uma collection especializada em um tipo de dados definido ao instanciar a classe.

Parent `AeonDigital\Collection\Collection`

Implements `AeonDigital\Interfaces\Collection\iTypeList`

2.1.1.1.1.7 Properties

2.1.1.1.1.8 Methods

public `AeonDigital\Collection\TypeList::isNullable()`

Indica se a coleção aceita valores `null` para seus pares de chave/valor.

Returns `< bool >`

public `AeonDigital\Collection\TypeList::getType()`

Retorna o tipo de dado que é aceito para o valor dos itens da coleção.

Se nenhum tipo for definido, o valor padrão é `mixed`.

Um sinal de interrogação `?` no início do nome do tipo indica que além de objetos daquele próprio tipo, é aceito também `null` como um valor válido de ser armazenado na coleção.

Returns `< string >`

public `AeonDigital\Collection\TypeList::__construct($type="", $initialValues=[], $autoincrement=false)`

Inicia nova lista de dados.

Parameters

- `< string >` **\$type** Tipo de dados que devem ser aceitos para cada item da lista. Os tipos de classes e interfaces, quando usados devem vir com seus nomes completos, ou seja `namespace + classname`.
- `< ?array >` **\$initialValues** Valores com os quais a instância deve iniciar.
- `< bool >` **\$autoincrement** Quando `true` permite que seja omitido o nome da chave dos valores pois eles serão definidos internamente conforme fosse um array começando em zero.

Throws `< InvalidArgumentException >` Caso algum dos valores iniciais a serem definidos não seja aceito.

2.1.1.2 SimpleType

AeonDigitalSimpleType

2.1.1.2.1 Namespaces

2.1.1.2.1.1 Abstracts

AeonDigitalSimpleTypeAbstracts

2.1.1.2.1.2 Classes

2.1.1.2.1.3 aBasic

```
class AeonDigital\SimpleType\Abstracts\aBasic
```

Classe abstrata que fundamenta a construção de tipos simples.

Implements AeonDigital\Interfaces\SimpleType\iBasic

2.1.1.2.1.4 Methods

```
public abstract static validate( $v)
```

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

```
public static AeonDigital\SimpleType\Abstracts\aBasic::toString($v)
```

Tenta efetuar a conversão do valor indicado para o tipo `string`. Caso não seja possível converter o valor, retorna `null`.

Parameters

- < mixed > **\$v** Valor que será convertido.

Returns < ?string >

```
public abstract static parseIfValidate( $v, &$err=null)
```

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.1.5 aFloat

class `AeonDigital\SimpleType\Abstracts\AFloat`

Estende a classe `aBasic` dando funções de controle para lidar com números inteiros.

Parent `AeonDigital\SimpleType\Abstracts\ABasic`

Implements `AeonDigital\Interfaces\SimpleType\IFloat`

2.1.1.2.1.6 Methods

public abstract static min()

Retorna o menor valor possível para o tipo definido.

Returns < float >

public abstract static max()

Retorna o maior valor possível para o tipo definido.

Returns < float >

public static `AeonDigital\SimpleType\Abstracts\AFloat::validate($v)`

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

public static `AeonDigital\SimpleType\Abstracts\AFloat::parseIfValidate($v, &$err=null)`

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.1.7 aInt

class `AeonDigital\SimpleType\Abstracts\AInt`

Estende a classe `aBasic` dando funções de controle para lidar com números inteiros.

Parent `AeonDigital\SimpleType\Abstracts\ABasic`

Implements `AeonDigital\Interfaces\SimpleType\iInt`

2.1.1.2.1.8 Methods

public abstract static min()

Retorna o menor valor possível para o tipo definido.

Returns < int >

public abstract static max()

Retorna o maior valor possível para o tipo definido.

Returns < int >

public static `AeonDigital\SimpleType\Abstracts\AInt::validate($v)`

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

public static `AeonDigital\SimpleType\Abstracts\AInt::parseIfValidate($v, &$err=null)`

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.2 Classes**2.1.1.2.2.1 stBool**

```
class AeonDigital\SimpleType\stBool
```

Definições para o tipo bool.

Parent *AeonDigital\SimpleType\Abstracts\abasic*

Implements *AeonDigital\Interfaces\SimpleType\iBool*

2.1.1.2.2.2 Methods

```
public static AeonDigital\SimpleType\stBool::validate($v)
```

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

```
public static AeonDigital\SimpleType\stBool::toString($v)
```

Tenta efetuar a conversão do valor indicado para o tipo `string`. Caso não seja possível converter o valor, retorna `null`.

Parameters

- < mixed > **\$v** Valor que será convertido.

Returns < ?string >

```
public static AeonDigital\SimpleType\stBool::parseIfValidate($v, &$err=null)
```

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.2.3 stByte

class AeonDigital\SimpleType\stByte

Definições para o tipo byte (integer 8 bits).

Parent *AeonDigital\SimpleType\Abstracts\Int*

2.1.1.2.2.4 Methods

public static AeonDigital\SimpleType\stByte::min()

Retorna o menor valor possível para o tipo definido.

Returns < int >

public static AeonDigital\SimpleType\stByte::max()

Retorna o maior valor possível para o tipo definido.

Returns < int >

2.1.1.2.2.5 stDateTime

class AeonDigital\SimpleType\stDateTime

Definições para o tipo DateTime.

Os valores mínimos e máximos são definidos para estarem dentro de um intervalo que abrange a imensa maior parte das aplicações comerciais.

Parent *AeonDigital\SimpleType\Abstracts\Basic*

Implements AeonDigital\Interfaces\SimpleType\iDateTime

2.1.1.2.2.6 Methods

```
public static AeonDigital\SimpleType\stDateTime::min()
```

Retorna o menor valor possível para este tipo de data. Se for definido `null`, o limite será dado pelo próprio sistema.

Returns < ?DateTime >

```
public static AeonDigital\SimpleType\stDateTime::max()
```

Retorna o maior valor possível para este tipo de data. Se for definido `null`, o limite será dado pelo próprio sistema.

Returns < ?DateTime >

```
public static AeonDigital\SimpleType\stDateTime::validate($v)
```

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

É esperado uma string usando o formato **Y-m-d H:i:s**, um inteiro representando um **timestamp** ou um objeto `DateTime` dentro dos limites especificados.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

```
public static AeonDigital\SimpleType\stDateTime::parseIfValidate($v,  
                                                                    &$err=null)
```

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.2.7 stDouble

class AeonDigital\SimpleType\stDouble

Definições para o tipo double (float 64 bits).

Parent *AeonDigital\SimpleType\Abstracts\Float*

2.1.1.2.2.8 Methods

public static AeonDigital\SimpleType\stDouble::**min**()

Retorna o menor valor possível para o tipo definido.

Returns < float >

public static AeonDigital\SimpleType\stDouble::**max**()

Retorna o maior valor possível para o tipo definido.

Returns < float >

2.1.1.2.2.9 stFloat

class AeonDigital\SimpleType\stFloat

Definições para o tipo float (float 32 bits).

Parent *AeonDigital\SimpleType\Abstracts\Float*

2.1.1.2.2.10 Methods

public static AeonDigital\SimpleType\stFloat::**min**()

Retorna o menor valor possível para o tipo definido.

Returns < float >

public static AeonDigital\SimpleType\stFloat::**max**()

Retorna o maior valor possível para o tipo definido.

Returns < float >

2.1.1.2.2.11 stInt

class AeonDigital\SimpleType\stInt

Definições para o tipo int (integer 32 bits).

Parent *AeonDigital\SimpleType\Abstracts\Int*

2.1.1.2.2.12 Methods

public static AeonDigital\SimpleType\stInt::min()

Retorna o menor valor possível para o tipo definido.

Returns < int >

public static AeonDigital\SimpleType\stInt::max()

Retorna o maior valor possível para o tipo definido.

Returns < int >

2.1.1.2.2.13 stLong

class AeonDigital\SimpleType\stLong

Definições para o tipo long (integer 64 bits).

Importante

Em sistemas de 64 bits o limiar mínimo para valores inteiros é de **-9223372036854775808** e o máximo é de **9223372036854775807**. No entanto, a partir destes próprios números o PHP passa a tratá-los como sendo valores de ponto flutuante e sua representação passa a ser feita usando notação científica o que impede comparações com precisão.

Para evitar tal comportamento e manter a precisão no uso deste tipo de valor inteiro, optou-se por reduzir em 1 cada um dos limites. Com isso, dentro da coleção de possíveis valores, toda comparação realizada será precisa.

Uma demonstração de possíveis problemas que este comportamento pode causar estão documentados nos testes desta mesma classe.

Parent *AeonDigital\SimpleType\Abstracts\Int*

2.1.1.2.2.14 Methods

```
public static AeonDigital\SimpleType\stLong::min()
```

Retorna o menor valor possível para o tipo definido.

Returns < int >

```
public static AeonDigital\SimpleType\stLong::max()
```

Retorna o maior valor possível para o tipo definido.

Returns < int >

2.1.1.2.2.15 stReal

```
class AeonDigital\SimpleType\stReal
```

Definições para o tipo real.

Parent *AeonDigital\SimpleType\Abstracts\abasic*

Implements AeonDigital\Interfaces\SimpleType\iReal

2.1.1.2.2.16 Methods

```
public static AeonDigital\SimpleType\stReal::min()
```

Retorna o menor valor possível para o tipo definido. Se for definido `null`, não haverá limites para a representação numérica a ser utilizada.

Returns < ?string >

```
public static AeonDigital\SimpleType\stReal::max()
```

Retorna o maior valor possível para o tipo definido. Se for definido `null`, não haverá limites para a representação numérica a ser utilizada.

Returns < ?string >

```
public static AeonDigital\SimpleType\stReal::validate($v)
```

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- < mixed > **\$v** Valor que será verificado.

Returns < bool >

```
public static AeonDigital\SimpleType\stReal::parseIfValidate($v, &$err=null)
```

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.2.2.17 stShort

```
class AeonDigital\SimpleType\stShort
```

Definições para o tipo `short` (integer 16 bits).

Parent *AeonDigital\SimpleType\Abstracts\aint*

2.1.1.2.2.18 Methods

```
public static AeonDigital\SimpleType\stShort::min()
```

Retorna o menor valor possível para o tipo definido.

Returns < int >

```
public static AeonDigital\SimpleType\stShort::max()
```

Retorna o maior valor possível para o tipo definido.

Returns < int >

2.1.1.2.2.19 stString

```
class AeonDigital\SimpleType\stString
```

Definições para o tipo `string`.

Parent *AeonDigital\SimpleType\Abstracts\abasic*

Implements *AeonDigital\Interfaces\SimpleType\istring*

2.1.1.2.2.20 Methods

```
public static AeonDigital\SimpleType\stString::validate($v)
```

Verifica se o valor indicado pode ser convertido e usado como um valor válido dentro das definições deste tipo.

Parameters

- `< mixed > $v` Valor que será verificado.

Returns < bool >[illegible]

Efetuará a conversão do valor indicado para o tipo que esta classe representa apenas se passar na validação.

Caso não passe retornará um código que identifica o erro ocorrido na variável `$err`.

Parameters

- < mixed > **\$v** Valor que será convertido.
- < ?string > **\$err** Código do erro da validação.

Returns < mixed >

2.1.1.3 Tools

AeonDigitalTools

2.1.1.3.1 Classes

2.1.1.3.1.1 Image

```
class AeonDigital\Tools\Image
```

Coleção de métodos estáticos para a manipulação de imagens.

2.1.1.3.1.2 Methods

```
public static AeonDigital\Tools\Image::transform($absoluteSystemPathToOriginalImage,  
                                                $absoluteSystemPathTo-  
                                                NewImage=null,           $resi-  
                                                zeType=&#34;auto&#34;;,  
                                                $imgMaxWidth=null,       $img-  
                                                MaxHeight=null,    $imgCrop=null,  
                                                $imgCropWidth=null,    $img-  
                                                CropHeight=null, $imgCropX=null,  
                                                $imgCropY=null)
```

Efetua a transformação de uma imagem conforme os parâmetros de configuração.

Parameters

- < string > **\$absoluteSystemPathToOriginalImage** Caminho completo até a imagem.
- < string > **\$absoluteSystemPathToNewImage** Caminho completo até o local onde a nova imagem será armazenada. Se não for definido a imagem original será alterada.
- < string > **\$resizeType** Tipo de ajuste que será feito.

Os seguintes valores são aceitos:

exact : redimensiona a imagem exatamente na medida definida em `$imgMaxWidth` e `$imgMaxHeight`. Se um destes valores não for definido, manterá o valor inicial da imagem.

portrait : redimensiona (verticalmente) a imagem parando quando encontra chegar na altura máxima definida em `$imgMaxHeight`.

landscape : redimensiona (horizontalmente) a imagem parando quando encontra chegar na largura máxima definida em `$imgMaxWidth`.

auto : redimensiona a imagem até que uma das dimensões encontre um dos valores máximos definidos por `$imgMaxWidth` e `$imgMaxHeight`.

- < ?int > **\$imgMaxWidth** Largura máxima que a imagem deverá ter. Se não for definido, este valor será calculado conforme o tipo `$resizeType`.
- < ?int > **\$imgMaxHeight** Altura máxima que a imagem deverá ter. Se não for definido, este valor será calculado conforme o tipo `$resizeType`.
- < ?bool > **\$imgCrop** Quando `true`, irá, após redimensionar a imagem, efetuar um crop(corte) na imagem resultante e salvará este corte ao invés da imagem como um todo.

Para evitar que um crop exceda os limites de uma imagem que será redimensionada por um método dinâmico (`portrait` | `landscape` | `auto`) é recomendável, mas não obrigatório, que esta opção seja usada em conjunto com o método `exact`.

- < ?int > **\$imgCropWidth** Largura do crop que será feito. Apenas surte efeito se `$resizeType` for definido como `crop`.
- < ?int > **\$imgCropHeight** Altura do crop que será feito. Apenas surte efeito se `$resizeType` for definido como `crop`.
- < ?int > **\$imgCropX** Posição no eixo X a partir de onde o corte da imagem deve ocorrer. Apenas surte efeito se `$resizeType` for definido como `crop`.
- < ?int > **\$imgCropY** Posição no eixo Y a partir de onde o corte da imagem deve ocorrer. Apenas surte efeito se `$resizeType` for definido como `crop`.

Returns < bool >

```
public static AeonDigital\Tools\Image::resize($absoluteSystemPathToOriginalImage,
                                             $absoluteSystemPathToNewImage=null,
                                             $resizeType=&#34;auto&#34;, $img-
                                             MaxWidth=null, $imgMaxHeight=null)
```

Efetua o redimensionamento de uma imagem conforme os parametros de configuração.

Parameters

- < string > **\$absoluteSystemPathToOriginalImage** Caminho completo até a imagem.
- < string > **\$absoluteSystemPathToNewImage** Caminho completo até o local onde a nova imagem será armazenada. Se não for definido a imagem original será alterada.
- < string > **\$resizeType** Tipo de ajuste que será feito.

Os seguintes valores são aceitos:

exact : redimensiona a imagem exatamente na medida definida em `$imgMaxWidth` e `$imgMaxHeight`. Se um destes valores não for definido, manterá o valor inicial da imagem.

portrait : redimensiona (verticalmente) a imagem parando quando encontra chegar na altura máxima definida em `$imgMaxHeight`.

landscape : redimensiona (horizontalmente) a imagem parando quando encontra chegar na largura máxima definida em `$imgMaxWidth`.

auto : redimensiona a imagem até que uma das dimensões encontre um dos valores máximos definidos por `$imgMaxWidth` e `$imgMaxHeight`.

- < ?int > **\$imgMaxWidth** Largura máxima que a imagem deverá ter. Se não for definido, este valor será calculado conforme o tipo `$resizeType`.
- < ?int > **\$imgMaxHeight** Altura máxima que a imagem deverá ter. Se não for definido, este valor será calculado conforme o tipo `$resizeType`.

Returns < bool >

```
public static AeonDigital\Tools\Image::crop($absoluteSystemPathToOriginalImage, $ab-
                                             soluteSystemPathToNewImage=null, $img-
                                             CropWidth=null, $imgCropHeight=null,
                                             $imgCropX=null, $imgCropY=null)
```

Efetua o crop de uma imagem conforme os parametros de configuração.

Parameters

- < string > **\$absoluteSystemPathToOriginalImage** Caminho completo até a imagem.
- < string > **\$absoluteSystemPathToNewImage** Caminho completo até o local onde a nova imagem será armazenada. Se não for definido a imagem original será alterada.
- < ?int > **\$imgCropWidth** Largura do crop que será feito. Apenas surte efeito se `$resizeType` for definido como crop.
- < ?int > **\$imgCropHeight** Altura do crop que será feito. Apenas surte efeito se `$resizeType` for definido como crop.

- < ?int > **\$imgCropX** Posição no eixo X a partir de onde o corte da imagem deve ocorrer. Apenas surte efeito se `$resizeType` for definido como `crop`
- < ?int > **\$imgCropY** Posição no eixo Y a partir de onde o corte da imagem deve ocorrer. Apenas surte efeito se `$resizeType` for definido como `crop`

Returns < bool >

2.1.1.3.1.3 JSON

class AeonDigital\Tools\JSON

Coleção de métodos estáticos para tratamento de arquivos JSON.

2.1.1.3.1.4 Methods

public static AeonDigital\Tools\JSON::**retrieve** (*\$absoluteSystemPathToFile*, *\$assoc=true*)

Carrega o conteúdo de um arquivo JSON na memória e retorna um Array Associativo ou um objeto do tipo `stdClass`. Caso o arquivo alvo não exista será retornado `null`.

Apesar do padrão JSON não assumir a possibilidade de haver comentários este método irá remover os mesmos se existirem e carregará o conteúdo normalmente.

Parameters

- < string > **\$absoluteSystemPathToFile** Caminho completo até o arquivo que será carregado.

Returns < ?array | \stdClass >

public static AeonDigital\Tools\JSON::**indent** (*\$strJSON*)

Identifica adequadamente uma string representante de um objeto JSON.

Parameters

- < string > **\$strJSON** String que será indentada.

Returns < string >

public static AeonDigital\Tools\JSON::**save** (*\$absoluteSystemPathToFile*, *\$JSON*, *\$options=0*)

Salva o um objeto JSON (representado por uma String, Array Associativo ou objeto `stdClass` no caminho informado).

Parameters

- < string > **\$absoluteSystemPathToFile** Caminho completo até o arquivo que será salvo.
- < string | array | \stdClass > **\$JSON** Objeto que será salvo como um arquivo JSON.

- < int > **\$options** [Flags](http://php.net/manual/pt_BR/json.constants.php) para salvar o documento JSON.

Returns < bool >

2.1.1.3.1.5 MinifyCSS

class AeonDigital\Tools\MinifyCSS

Coleção de métodos estáticos para minificar arquivos CSS.

2.1.1.3.1.6 Methods

public static AeonDigital\Tools\MinifyCSS::**minifyCode** (*\$cssCode*)

Minifica o código CSS informado.

Parameters

- < string > **\$cssCode** Código CSS que será minificado.

Returns < string >

public static AeonDigital\Tools\MinifyCSS::**minifyFile** (*\$absoluteSystemPathToFile*)

Minifica o conteúdo de um arquivo CSS.

Parameters

- < string > **\$absoluteSystemPathToFile** Endereço local do arquivo que será minificado.

Returns < string >

public static AeonDigital\Tools\MinifyCSS::**minifyFiles** (*\$absoluteSystemPathToFiles*)

Minifica uma coleção de arquivos CSS.

Parameters

- < string[] > **\$absoluteSystemPathToFiles** Endereço local dos arquivos que serão minificados.

Returns < string >

public static AeonDigital\Tools\MinifyCSS::**createMinifyFile** (*\$absoluteSystemPathToFiles*,
\$absoluteSystemPathToMinifiedFile)

Minifica uma coleção de arquivos CSS em um único arquivo.

Parameters

- < string[] > **\$absoluteSystemPathToFiles** Endereço local dos arquivos que serão minificados.

- < string > **\$absoluteSystemPathToMinifiedFile** Endereço completo onde o arquivo minificado será armazenado.

Returns < bool >

2.1.1.3.1.7 MinifyJS

class AeonDigital\Tools\MinifyJS

Coleção de métodos estáticos para minificar arquivos JS.

2.1.1.3.1.8 Methods

public static AeonDigital\Tools\MinifyJS::**minifyCode** (*\$jsCode*)

Minifica o código JS informado.

Parameters

- < string > **\$jsCode** Código JS que será minificado.

Returns < string >

public static AeonDigital\Tools\MinifyJS::**minifyFile** (*\$absoluteSystemPathToFile*)

Minifica o conteúdo de um arquivo JS.

Parameters

- < string > **\$absoluteSystemPathToFile** Endereço local do arquivo que será minificado.

Returns < string >

public static AeonDigital\Tools\MinifyJS::**minifyFiles** (*\$absoluteSystemPathToFiles*)

Minifica uma coleção de arquivos JS.

Parameters

- < string[] > **\$absoluteSystemPathToFiles** Endereço local dos arquivos que serão minificados.

Returns < string >

public static AeonDigital\Tools\MinifyJS::**createMinifyFile** (*\$absoluteSystemPathToFiles*,
\$absoluteSystemPathToMinifiedFile)

Minifica uma coleção de arquivos JS em um único arquivo.

Parameters

- < string[] > **\$absoluteSystemPathToFiles** Endereço local dos arquivos que serão minificados.

- < string > **\$absoluteSystemPathToMinifiedFile** Endereço completo onde o arquivo minificado será armazenado.

Returns < bool >

2.1.1.3.1.9 Zip

class AeonDigital\Tools\Zip

Coleção de métodos estáticos para uso de arquivos Zip.

2.1.1.3.1.10 Methods

public static AeonDigital\Tools\Zip::**packTargets** (*\$absoluteSystemPaths*, *\$absoluteSystemPathToFile*)

Zipa um conjunto de arquivos e diretórios e gera um pacote com os dados no local indicado.

Parameters

- < string[] > **\$absoluteSystemPaths** Coleção de endereços dos arquivos e diretórios que serão zipados.
- < string > **\$absoluteSystemPathToFile** Endereço completo onde o novo arquivo zip será gerado.

Returns < bool >

public static AeonDigital\Tools\Zip::**pack** (*\$absoluteSystemPaths*, *\$absoluteSystemPathToFile=null*)

Zipa um arquivo ou diretório (e todo seu conteúdo) gerando um pacote com os dados encontrados no mesmo local onde estão os dados apontados.

Parameters

- < string > **\$absoluteSystemPaths** Endereço completo do arquivo ou diretório que será zipado.
- < ?string > **\$absoluteSystemPathToFile** Quando definido, deve indicar o local de destino do pacote gerado e seu respectivo nome.

Returns < bool >

public static AeonDigital\Tools\Zip::**unpack** (*\$absoluteSystemPathToFile*, *\$absoluteSystemPathToDir=null*)

Deszipa um pacote e adiciona seu conteúdo no local indicado.

SE o local não existir, cria-o.

Parameters

- < string > **\$absoluteSystemPathToFile** Caminho completo até o arquivo zipado.

- < ?string > **\$absoluteSystemPathToDir** Caminho completo até o diretório onde o pacote será descompactado. Caso não seja definido, criará um diretório no mesmo local onde o arquivo **.zip** se encontra. O novo diretório terá como nome: **dirname_unpacked** e, caso já exista, será adicionado um index.

Returns < bool >

```
public static AeonDigital\Tools\Zip::unpackTargets ($absoluteSystemPathToFile, $targets)
```

Extraí um ou mais arquivos ou diretórios de dentro de um arquivo zipado e aloca-os em seus respectivos destinos.

Parameters

- < string > **\$absoluteSystemPathToFile** Caminho completo até o arquivo zipado.
- < array > **\$targets** Array de arrays associativos onde:
 - key** : Deve ser o caminho para o arquivo ou diretório dentro do zip (a partir da raiz do zip).
 - value** : Deve ser o caminho completo do diretório onde o arquivo ou diretório será extraído.

Returns < bool >

2.1.1.4 Traits

AeonDigitalTraits

2.1.1.4.1 Traits

2.1.1.4.1.1 MainCheckArgumentException

```
trait AeonDigital\Traits\MainCheckArgumentException
```

Coleção de métodos que tem como função prover às classes implementadoras um formato padronizado para verificação de argumentos e mensagens de erro para quando os mesmos forem disparados.

2.1.1.4.1.2 Properties

2.1.1.4.1.3 Methods

2.1.2 Classes

2.1.2.1 BObject

```
class AeonDigital\BObject
```

Classe básica.

2.1.2.1.1 Methods

public AeonDigital\BObject::__set(\$name, \$value)

Desabilita a função mágica __set para assegurar a que apenas alterações dentro das regras definidas para a classe sejam possíveis.

public AeonDigital\BObject::__get(\$name)

Desabilita a função mágica __sget para assegurar a que apenas alterações dentro das regras definidas para a classe sejam possíveis.

public AeonDigital\BObject::__unset(\$name)

Desabilita a função mágica __unset.

public AeonDigital\BObject::__toString()

Desabilita a função mágica __toString.

public AeonDigital\BObject::__invoke(\$x)

Desabilita a função mágica __invoke.

public static AeonDigital\BObject::__set_state(\$assoc_array)

Desabilita a função mágica __set_state.

2.1.2.2 Realtype

class AeonDigital\Realtype

Permite operações matemáticas com números reais de qualquer precisão decimal.

Utiliza a extensão matemática **BC Math**.

Parent *AeonDigital\BObject*

Used traits AeonDigital\TraitsMainCheckArgumentException

2.1.2.2.1 Properties

2.1.2.2.2 Methods

public AeonDigital\Realtype::value()

Retorna o valor que esta instância está representando.

Returns < string >

```
public AeonDigital\Realtye::getIntegerPart()
```

Retorna apenas a parte inteira do numeral representado por esta instância.

Returns < string >

```
public AeonDigital\Realtye::getDecimalPart()
```

Retorna apenas a parte decimal do numeral representado por esta instância.

Returns < string >

```
public AeonDigital\Realtye::precision()
```

Retorna o total de dígitos que compõe este numeral somando o total de casas antes e após o separador decimal.

Returns < int >

```
public AeonDigital\Realtye::integerPlaces()
```

Retorna o total de dígitos que são usados para representar a parte inteira do numeral atual.

Returns < int >

```
public AeonDigital\Realtye::decimalPlaces()
```

Retorna o total de dígitos que são usados para representar a parte decimal do numeral atual.

Returns < int >

```
public static AeonDigital\Realtye::defineGlobalDecimalPlaces($v)
```

Permite definir um valor padrão para o argumento \$dPlaces usado em vários métodos desta classe.

Quando algum método que usa o argumento \$dPlaces for igual a null, o valor aqui definido é que será usado.

Parameters

- < int > \$v Valor padrão a ser usado.

Returns < void >

```
public static AeonDigital\Realtye::getGlobalDecimalPlaces()
```

Retorna o número de casas decimais sendo usadas no momento para fins de cálculos com esta classe.

Returns < int >

```
public static AeonDigital\Realtye::defineGlobalRoundType($roundType, $sensibility)
```

Define a forma padrão pela qual os valores, quando calculados, serão arredondados.

Parameters

- < string > **\$roundType** Indica o tipo de arredondamento que será feito. Valores inválidos não incorrerão em erros e nem em nenhuma conversão.

Os valores aceitos são: **floor** : Arredondará para baixo qualquer valor a partir do **digito sensível**.

ceil [Arredondará para cima todo valor diferente de zero a partir] do **digito sensível**.

floor-n [Arredondará para baixo todo **digito sensível** que seja] igual ou menor que **n** e para cima todo **digito sensível** maior que **n**.

ceil-n [Arredondará para cima todo **digito sensível** que seja igual] ou maior que **n** e para baixo todo **digito sensível** menor que **n**.

- < AeonDigital\Realtye > **\$sensibility** A sensibilidade é sempre um valor que indica qual será exatamente o digito que será sensível ao arredondamento.

Por exemplo: 0.001 fará o arredondamento do número a partir do 3º digito após o ponto decimal enquanto 10 fará o arredondamento das casas das dezenas.

Returns < void >

```
public static AeonDigital\Realtye::getRoundType()
```

Retorna o tipo de arredondamento definido para os cálculos realizados com esta classe.

Returns < ?string >

```
public static AeonDigital\Realtye::getRoundSensibility()
```

Retorna o nível de sensibilidade usada para os arredondamentos.

Returns < ?string >

```
public AeonDigital\Realtye::__construct($v=0)
```

Inicia um novo objeto Realtye com o valor indicado.

Parameters

- < mixed > **\$v** É esperado valores Realtye, int, float ou uma string numérica.

Throws < InvalidArgumentException > Lançado se o valor inicial indicado não for aceitável para iniciar o objeto.

```
public static AeonDigital\Realtye::isValidRealtye($v)
```

Identifica se o valor passado é um Realtye válido.

Parameters

- < mixed > **\$v** É esperado valores Realtye, int, float ou uma string numérica.

Returns < bool > Retorna true se o valor passado for válido.

```
public AeonDigital\Realtye::isEqualAs($v)
```

Verifica se o valor atual desta instância é igual ao valor passado para comparação.

Parameters

- < mixed > **\$v** Valor usado para comparação. É esperado valores Realtye, int, float ou uma string numérica.

Returns < bool > Retorna true se o valor atual desta instância e o valor passado em \$v forem **IDÊNTICOS**.

```
public AeonDigital\Realtye::isGreaterThan($v)
```

Verifica se o valor atual desta instância é maior que o valor passado para comparação.

Parameters

- < mixed > **\$v** Valor usado para comparação. É esperado valores Realtye, int, float ou uma string numérica.

Returns < bool > Retornará true se o valor atual desta instância é **MAIOR** que o valor indicado em \$v.

```
public AeonDigital\Realtye::isGreaterOrEqualAs($v)
```

Verifica se o valor atual desta instância é maior ou igual ao valor passado para comparação.

Parameters

- < mixed > **\$v** Valor usado para comparação. É esperado valores Realtye, int, float ou uma string numérica.

Returns < bool > Retornará true se o valor atual desta instância é **MAIOR** ou **IGUAL** ao o valor indicado em \$v.

```
public AeonDigital\Realtye::isLessThan($v)
```

Verifica se o valor atual desta instância é menor que o valor passado para comparação.

Parameters

- < mixed > **\$v** Valor usado para comparação. É esperado valores Realtye, int, float ou uma string numérica.

Returns < bool > Retornará `true` se o valor atual desta instância é **MENOR** que o valor indicado em `$v`.

```
public AeonDigital\Realtye::isLessOrEqualAs ($v)
```

Verifica se o valor atual desta instância é menor ou igual ao valor passado para comparação.

Parameters

- < mixed > `$v` Valor usado para comparação. É esperado valores `Realtye`, `int`, `float` ou uma `string` numérica.

Returns < bool > Retornará `true` se o valor atual desta instância é **MENOR** ou **IGUAL** ao o valor indicado em `$v`.

```
public AeonDigital\Realtye::isZero ()
```

Verifica se o valor atual desta instância é zero.

Returns < bool > Retornará `true` se o valor atual desta instância for zero.

```
public AeonDigital\Realtye::isPositive ()
```

Verifica se o valor atual desta instância é um número positivo.

Returns < bool > Retornará `true` se o valor atual desta instância for um número positivo.

```
public AeonDigital\Realtye::isNegative ()
```

Verifica se o valor atual desta instância é um número negativo.

Returns < bool > Retornará `true` se o valor atual desta instância for um número negativo.

```
public AeonDigital\Realtye::toPositive ()
```

Retorna uma nova instância `Realtye` com o mesmo valor atual desta instância mas com o sinal positivo.

Returns < \AeonDigital\Realtye >

```
public AeonDigital\Realtye::toNegative ()
```

Retorna uma nova instância `Realtye` com o mesmo valor atual desta instância mas com o sinal negativo.

Returns < \AeonDigital\Realtye >

```
public AeonDigital\Realtye::invertSignal ()
```

Retorna uma nova instância `Realtye` com o mesmo valor atual desta instância mas com o sinal invertido.

Returns < \AeonDigital\Realtyype >

```
public static AeonDigital\Realtyype::roundTo ($v, $roundType, $sensitivity)
```

Efetua o arredondamento de valores conforme as regras indicadas.

Parameters

- < AeonDigital\Realtyype > **\$v** Valor que será arredondado.
- < string > **\$roundType** Indica o tipo de arredondamento que será feito. Valores inválidos não incorrerão em erros e nem em nenhuma conversão.

Os valores aceitos são: **floor** : Arredondará para baixo qualquer valor a partir do **digito sensível**.

ceil [Arredondará para cima todo valor diferente de zero a partir] do **digito sensível**.

floor-n [Arredondará para baixo todo **digito sensível** que seja] igual ou menor que **n** e para cima todo **digito sensível** maior que **n**.

ceil-n [Arredondará para cima todo **digito sensível** que seja igual] ou maior que **n** e para baixo todo **digito sensível** menor que **n**.

- < AeonDigital\Realtyype > **\$sensitivity** A sensibilidade é sempre um valor que indica qual será exatamente o digito que será sensível ao arredondamento.

Por exemplo: 0.001 fará o arredondamento do número a partir do 3º digito após o ponto decimal enquanto 10 fará o arredondamento das casas das dezenas.

Returns < \AeonDigital\Realtyype > Nova instância Realtyype com o resultado do arredondamento indicado.

```
public AeonDigital\Realtyype::sum ($v, $dPlaces=null)
```

Efetua uma adição do valor atual desta instância com o valor indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realtyype, int, float ou uma *string* numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se *null* for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns < \AeonDigital\Realtyype > Nova instância com o resultado desta operação.

```
public AeonDigital\Realtyype::sub ($v, $dPlaces=null)
```

Efetua uma subtração do valor atual desta instância com o valor indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realtyype, int, float ou uma *string* numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se *null* for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns <\AeonDigital\Realty> Nova instância com o resultado desta operação.

```
public AeonDigital\Realty::mul($v, $dPlaces=null)
```

Efetua uma multiplicação do valor atual desta instância com o valor indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realty, int, float ou uma string numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se null for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns <\AeonDigital\Realty> Nova instância com o resultado desta operação.

```
public AeonDigital\Realty::div($v, $dPlaces=null)
```

Efetua uma divisão do valor atual desta instância com o valor indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realty, int, float ou uma string numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se null for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns <\AeonDigital\Realty> Nova instância com o resultado desta operação.

```
public AeonDigital\Realty::mod($v, $dPlaces=null)
```

Calcula o módulo da divisão do valor atual desta instância pelo valor indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realty, int, float ou uma string numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se null for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns <\AeonDigital\Realty> Nova instância com o resultado desta operação.

```
public AeonDigital\Realty::pow($v, $dPlaces=null)
```

Eleva o valor atual desta instância pelo expoente indicado.

Parameters

- < mixed > **\$v** Valor usado para o cálculo. É esperado valores Realty, int, float ou uma string numérica.
- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se null for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns <\AeonDigital\Realty> Nova instância com o resultado desta operação.

```
public AeonDigital\Realtye::sqrt ($dPlaces=null)
```

Calcula a raiz quadrada do valor atual desta instância.

Parameters

- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se `null` for passado, usará o padrão definido em `static::$globalDecimalPlaces`.

Returns < AeonDigital\Realtye > Raiz quadrada do valor atual desta instância.

```
public AeonDigital\Realtye::__toString()
```

Configura a forma como uma instância deve se comportar quando forçada a ser convertida para uma `string`.

Returns < string >

```
public static AeonDigital\Realtye::__set_state ($state)
```

Permite definir um novo objeto baseado no estado completo passado pelo parametro `$state`.

Parameters

- < array > **\$state** Dados que serão adicionados ao novo objeto.

Returns < AeonDigital\Realtye > Nova instância preenchida com os valores do estado indicado em `$state`.

```
public AeonDigital\Realtye::format ($dPlaces=null, $dec, $tho)
```

Formata o valor atual desta instância usando o pontuador decimal e de milhar indicados.

Parameters

- < ?int > **\$dPlaces** Total de casas decimais a serem levadas em conta. Se `null` for passado, usará o padrão definido em `static::$globalDecimalPlaces`.
- < string > **\$dec** Pontuador decimal a ser usado.
- < string > **\$tho** Pontuador de milhar a ser usado.

Returns < string > Valor atual desta instância formatado conforme definido.

2.1.2.3 Tools

```
class AeonDigital\Tools
```

Coleção de métodos estáticos para diversos fins.

2.1.2.3.1 Methods

```
public static AeonDigital\Tools::getScalarType($o)
```

Retorna o tipo `scalar` do objeto passado.

Se não for um objeto do tipo `scalar` retornará `null`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < ?string >

```
public static AeonDigital\Tools::isScalarType($o, $type)
```

Verifica se o objeto passado corresponde ao tipo esperado.

Parameters

- < mixed > **\$o** Objeto que será verificado.
- < string > **\$type** Nome do tipo a ser testado. Esperado um dos seguintes: `null` | `bool` | `int` | `float` | `string` | `array`

Returns < bool >

```
public static AeonDigital\Tools::isScalar($o)
```

Identifica se o objeto passado é um tipo `scalar`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isNull($o)
```

Verifica se o objeto passado é do tipo `null`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isBool($o)
```

Verifica se o objeto passado é do tipo `bool`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isNumeric($o)
```

Verifica se o objeto passado é do tipo `int` ou `float` ou ainda se trata-se de uma `string` numérica.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isInt($o)
```

Verifica se o objeto passado é do tipo `int`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isFloat($o)
```

Verifica se o objeto passado é do tipo `float`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isString($o)
```

Verifica se o objeto passado é do tipo `string`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isArray($o)
```

Verifica se o objeto passado é do tipo `array`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isArrayAssoc($o)
```

Verifica se o objeto passado um `array` associativo.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isDateTime($o)
```

Verifica se o objeto passado é do tipo `DateTime`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::isRealtype($o)
```

Verifica se o objeto passado é do tipo `Realtype`.

Parameters

- < mixed > **\$o** Objeto que será verificado.

Returns < bool >

```
public static AeonDigital\Tools::toBool($o)
```

Tenta converter o tipo do valor passado para `bool`.

Apenas valores realmente compatíveis serão convertidos.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?bool > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toNumeric($o)
```

Tenta converter o tipo do valor passado para `number` (seja `int`, `float` ou `double`).

Apenas valores realmente compatíveis serão convertidos.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?int | ?float > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toInt($o)
```

Tenta converter o tipo do valor passado para `int`.

Apenas valores realmente compatíveis serão convertidos.

Números com ponto flutuante serão arredondados pela função `intval`.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?int > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toFloat($o)
```

Tenta converter o tipo do valor passado para `float`.

Apenas valores realmente compatíveis serão convertidos.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?float > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toString($o)
```

Tenta converter o tipo do valor passado para `string`.

Apenas valores realmente compatíveis serão convertidos.

Números de ponto flutuante serão convertidos e mantidos com no máximo 15 dígitos ao todo (parte inteira + parte decimal).

A parte decimal ficará com : (15 - (número de dígitos da parte inteira)) casas.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?string > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toArray($o)
```

Tenta converter o tipo do valor passado para `array`.

Apenas valores realmente compatíveis serão convertidos.

Returns < ?float > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toArrayStr($o, $force=false)
```

Converte todos os valores do array passado para `string` e retorna um novo array

contendo todos os valores convertidos. A conversão ocorre apenas entre valores escalares.

Se ao menos 1 dos valores originais não for passível de ser convertido, então o processamento falhará e `null` será retornado.

Parameters

- < ?array > **\$o** Coleção de valores originais.
- < bool > **\$force** Indica se deve forçar a conversão de tipos não escalares. Neste caso será verificado se o objeto possui implementado o método mágico `__toString` e, caso positivo, irá utilizá-lo, senão, irá retornar o nome completo da classe a qual este objeto pertence.

Returns < ?array >

```
public static AeonDigital\Tools::toDateTime($o, $format=&#34;Y-m-d H:i:s&#34;)
```

Tenta converter o tipo do valor passado para `DateTime`.

Apenas valores realmente compatíveis serão convertidos.

O formato padrão é o **Y-m-d H:i:s**.

Se no objeto original não houver o valor `time` e em `$format` também não existir definições para horários (H | i | s) esta conversão assumirá o valor zero para cada uma destas posições.

Parameters

- < string | int > **\$o** Objeto que será convertido. Para ser efetivamente processado, é esperado uma `string` ou um `int`, correspondente a um **timestamp**.
- < string > **\$format** Se o valor original for uma `string`, este parametro deve indicar qual formato que a data se encontra. Padrão é **Y-m-d H:i:s**

Returns < ?DateTime > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toDateTimeString($o,          $inputFormat=&#34;Y-
                                                    m-d          H:i:s&#34;,
                                                    $outputFormat=&#34;Y-m-d
                                                    H:i:s&#34;)
```

Tenta converter o tipo do valor passado para uma `DateTime` `string` compatível com o formato de saída escolhido.

Apenas valores realmente compatíveis serão convertidos.

Parameters

- < string | int | \DateTime > **\$o** Objeto que será convertido. Para ser efetivamente processado, é esperado um `DateTime`, uma `string` ou um `int`, correspondente a um **timestamp**.

- < string > **\$inputFormat** Se o valor original for uma `string`, este parametro deve indicar qual formato que a data se encontra. Padrão é **Y-m-d H:i:s**.
- < string > **\$outputFormat** Formato `DateTime` `string` em que o valor deve ser retornado. Padrão é **Y-m-d H:i:s**.

Returns < ?string > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toRealtyype($o)
```

Tenta converter o tipo do valor passado para `Realtyype`.

Apenas valores realmente compatíveis serão convertidos.

Parameters

- < mixed > **\$o** Objeto que será convertido.

Returns < ?AeonDigital\Realtyype > Retornará `null` caso não seja possível efetuar a conversão.

```
public static AeonDigital\Tools::toJSON($o)
```

Tenta converter o tipo do valor passado para uma `string json`.

Apenas valores realmente compatíveis serão convertidos.

Returns < ?string > Retornará `null` caso não seja possível efetuar a conversão.

a

AeonDigital, [52](#)
AeonDigital\Collection, [26](#)
AeonDigital\SimpleType, [36](#)
AeonDigital\SimpleType\Abstracts, [29](#)
AeonDigital\Tools, [43](#)
AeonDigital\Traits, [44](#)

Símbolos

__construct() (AeonDigital\Collection\BasicCollection method), [22](#)

__construct() (AeonDigital\Collection\Collection method), [25](#)

__construct() (AeonDigital\Collection\TypeList method), [26](#)

__construct() (AeonDigital\Realtytype method), [47](#)

__get() (AeonDigital\BObject method), [45](#)

__invoke() (AeonDigital\BObject method), [45](#)

__set() (AeonDigital\BObject method), [45](#)

__set() (AeonDigital\Collection\BasicCollection method), [24](#)

__set_state() (AeonDigital\BObject method), [45](#)

__set_state() (AeonDigital\Realtytype method), [52](#)

__toString() (AeonDigital\BObject method), [45](#)

__toString() (AeonDigital\Realtytype method), [52](#)

__unset() (AeonDigital\BObject method), [45](#)

A

aBasic (classe em AeonDigital\SimpleType\Abstracts), [27](#)

AeonDigital (namespace), [44](#), [45](#), [52](#)

AeonDigital\Collection (namespace), [19](#), [24](#), [26](#)

AeonDigital\SimpleType (namespace), [30](#), [31](#), [33–36](#)

AeonDigital\SimpleType\Abstracts (namespace), [27–29](#)

AeonDigital\Tools (namespace), [37](#), [40–43](#)

AeonDigital\Traits (namespace), [44](#)

aFloat (classe em AeonDigital\SimpleType\Abstracts), [28](#)

aInt (classe em AeonDigital\SimpleType\Abstracts), [29](#)

array_check_required_keys() (global function), [4](#)

array_in_ci() (global function), [4](#)

array_is_assoc() (global function), [4](#)

B

BasicCollection (classe em AeonDigital\Collection), [19](#)

BObject (classe em AeonDigital), [44](#)

C

clean() (AeonDigital\Collection\Collection method), [25](#)

Collection (classe em AeonDigital\Collection), [24](#)

count() (AeonDigital\Collection\BasicCollection method), [23](#)

createMinifyFile() (AeonDigital\Tools\MinifyCSS method), [41](#)

createMinifyFile() (AeonDigital\Tools\MinifyJS method), [42](#)

crop() (AeonDigital\Tools\Image method), [39](#)

D

date_to_first_month_day() (global function), [5](#)

date_to_last_month_day() (global function), [5](#)

decimalPlaces() (AeonDigital\Realtytype method), [46](#)

defineGlobalDecimalPlaces() (AeonDigital\Realtytype method), [46](#)

defineGlobalRoundType() (AeonDigital\Realtytype method), [47](#)

dir_chmod_r() (global function), [5](#)

dir_convert_to_utf8() (global function), [6](#)

dir_copy() (global function), [6](#)

dir_deltree() (global function), [6](#)

dir_scan_w() (global function), [7](#)

div() (AeonDigital\Realtytype method), [51](#)

DS (global constant), [4](#)

F

file_convert_to_utf8() (global function), [7](#)

format() (AeonDigital\Realtytype method), [52](#)

G

`get()` (*AeonDigital\Collection\BasicCollection method*), **21**
`getDecimalPart()` (*AeonDigital\Realty method*), **46**
`getGlobalDecimalPlaces()` (*AeonDigital\Realty method*), **46**
`getIntegerPart()` (*AeonDigital\Realty method*), **46**
`getIterator()` (*AeonDigital\Collection\BasicCollection method*), **24**
`getRoundSensibility()` (*AeonDigital\Realty method*), **47**
`getRoundType()` (*AeonDigital\Realty method*), **47**
`getScalarType()` (*AeonDigital\Tools method*), **53**
`getType()` (*AeonDigital\Collection\TypeList method*), **26**

H

`has()` (*AeonDigital\Collection\BasicCollection method*), **21**
`html_print()` (*global function*), **15**

I

`Image` (*classe em AeonDigital\Tools*), **37**
`indent()` (*AeonDigital\Tools\JSON method*), **40**
`insert()` (*AeonDigital\Collection\Collection method*), **25**
`integerPlaces()` (*AeonDigital\Realty method*), **46**
`invertSignal()` (*AeonDigital\Realty method*), **49**
`isAppendOnly()` (*AeonDigital\Collection\BasicCollection method*), **20**
`isArray()` (*AeonDigital\Tools method*), **54**
`isArrayAssoc()` (*AeonDigital\Tools method*), **54**
`isAutoIncrement()` (*AeonDigital\Collection\BasicCollection method*), **20**
`isBool()` (*AeonDigital\Tools method*), **53**
`isCaseInsensitive()` (*AeonDigital\Collection\BasicCollection method*), **20**
`isDateTime()` (*AeonDigital\Tools method*), **55**
`isEqualAs()` (*AeonDigital\Realty method*), **48**
`isFloat()` (*AeonDigital\Tools method*), **54**
`isGreaterOrEqualAs()` (*AeonDigital\Realty method*), **48**
`isGreaterThan()` (*AeonDigital\Realty method*), **48**
`isInt()` (*AeonDigital\Tools method*), **54**
`isLessOrEqualAs()` (*AeonDigital\Realty method*), **49**

`isLessThan()` (*AeonDigital\Realty method*), **48**
`isNegative()` (*AeonDigital\Realty method*), **49**
`isNull()` (*AeonDigital\Tools method*), **53**
`isNullable()` (*AeonDigital\Collection\TypeList method*), **26**
`isNumeric()` (*AeonDigital\Tools method*), **54**
`isPositive()` (*AeonDigital\Realty method*), **49**
`isProtected()` (*AeonDigital\Collection\BasicCollection method*), **20**
`isReadOnly()` (*AeonDigital\Collection\BasicCollection method*), **20**
`isRealty()` (*AeonDigital\Tools method*), **55**
`isScalar()` (*AeonDigital\Tools method*), **53**
`isScalarType()` (*AeonDigital\Tools method*), **53**
`isString()` (*AeonDigital\Tools method*), **54**
`isValidRealty()` (*AeonDigital\Realty method*), **48**
`isZero()` (*AeonDigital\Realty method*), **49**

J

`JSON` (*classe em AeonDigital\Tools*), **40**

M

`MainCheckArgumentException` (*trait in AeonDigital\Traits*), **44**
`max()` (*AeonDigital\SimpleType\stByte method*), **31**
`max()` (*AeonDigital\SimpleType\stDateTime method*), **32**
`max()` (*AeonDigital\SimpleType\stDouble method*), **33**
`max()` (*AeonDigital\SimpleType\stFloat method*), **33**
`max()` (*AeonDigital\SimpleType\stInt method*), **34**
`max()` (*AeonDigital\SimpleType\stLong method*), **35**
`max()` (*AeonDigital\SimpleType\stReal method*), **35**
`max()` (*AeonDigital\SimpleType\stShort method*), **36**
`mb_str_contains()` (*global function*), **7**
`mb_str_ends_with()` (*global function*), **7**
`mb_str_insert()` (*global function*), **8**
`mb_str_last_pos()` (*global function*), **8**
`mb_str_limit_chars()` (*global function*), **8**
`mb_str_limit_words()` (*global function*), **9**
`mb_str_pattern_match()` (*global function*), **9**
`mb_str_pos_all()` (*global function*), **9**
`mb_str_preserve_chars()` (*global function*), **9**
`mb_str_remove()` (*global function*), **10**
`mb_str_remove_chars()` (*global function*), **10**
`mb_str_remove_glyphs()` (*global function*), **10**
`mb_str_remove_invisible_chars()` (*global function*), **10**
`mb_str_replace_once()` (*global function*), **11**
`mb_str_starts_with()` (*global function*), **11**
`mb_str_to_valid_filename()` (*global function*), **11**

mb_str_uc_first() (global function), **12**
 mb_str_uc_names() (global function), **12**
 mb_str_uc_names_ptbr() (global function), **12**
 mb_str_uc_words() (global function), **13**
 mb_substr_replace() (global function), **13**
 min() (AeonDigital\SimpleType\stByte method), **31**
 min() (AeonDigital\SimpleType\stDateTime method), **32**
 min() (AeonDigital\SimpleType\stDouble method), **33**
 min() (AeonDigital\SimpleType\stFloat method), **33**
 min() (AeonDigital\SimpleType\stInt method), **34**
 min() (AeonDigital\SimpleType\stLong method), **35**
 min() (AeonDigital\SimpleType\stReal method), **35**
 min() (AeonDigital\SimpleType\stShort method), **36**
 minifyCode() (AeonDigital\Tools\MinifyCSS method), **41**
 minifyCode() (AeonDigital\Tools\MinifyJS method), **42**
 MinifyCSS (classe em AeonDigital\Tools), **41**
 minifyFile() (AeonDigital\Tools\MinifyCSS method), **41**
 minifyFile() (AeonDigital\Tools\MinifyJS method), **42**
 minifyFiles() (AeonDigital\Tools\MinifyCSS method), **41**
 minifyFiles() (AeonDigital\Tools\MinifyJS method), **42**
 MinifyJS (classe em AeonDigital\Tools), **42**
 mod() (AeonDigital\Realty method), **51**
 mul() (AeonDigital\Realty method), **51**

N

numeric_decimal_part() (global function), **13**
 numeric_integer_part() (global function), **14**
 numeric_is_even() (global function), **14**
 numeric_is_odd() (global function), **14**
 numeric_mod() (global function), **14**

O

object_clone() (global function), **15**
 object_convert_values_to_html_entities() (global function), **15**
 offsetExists() (AeonDigital\Collection\BasicCollection method), **22**
 offsetGet() (AeonDigital\Collection\BasicCollection method), **22**
 offsetSet() (AeonDigital\Collection\BasicCollection method), **23**
 offsetUnset() (AeonDigital\Collection\BasicCollection method), **23**

P

pack() (AeonDigital\Tools\Zip method), **43**
 packTargets() (AeonDigital\Tools\Zip method), **43**
 parseIfValidate() (AeonDigital\SimpleType\Abstracts\Float method), **28**
 parseIfValidate() (AeonDigital\SimpleType\Abstracts\Int method), **29**
 parseIfValidate() (AeonDigital\SimpleType\stBool method), **30**
 parseIfValidate() (AeonDigital\SimpleType\stDateTime method), **32**
 parseIfValidate() (AeonDigital\SimpleType\stReal method), **36**
 parseIfValidate() (AeonDigital\SimpleType\stString method), **37**
 pow() (AeonDigital\Realty method), **51**
 precision() (AeonDigital\Realty method), **46**

R

Realty (classe em AeonDigital), **45**
 redirect() (global function), **16**
 remove() (AeonDigital\Collection\BasicCollection method), **22**
 resize() (AeonDigital\Tools\Image method), **39**
 retrieve() (AeonDigital\Tools\JSON method), **40**
 roundTo() (AeonDigital\Realty method), **50**

S

save() (AeonDigital\Tools\JSON method), **40**
 set() (AeonDigital\Collection\BasicCollection method), **21**
 sqrt() (AeonDigital\Realty method), **51**
 stBool (classe em AeonDigital\SimpleType), **30**
 stByte (classe em AeonDigital\SimpleType), **31**
 stDateTime (classe em AeonDigital\SimpleType), **31**
 stDouble (classe em AeonDigital\SimpleType), **33**
 stFloat (classe em AeonDigital\SimpleType), **33**
 stInt (classe em AeonDigital\SimpleType), **34**
 stLong (classe em AeonDigital\SimpleType), **34**
 stReal (classe em AeonDigital\SimpleType), **35**
 stShort (classe em AeonDigital\SimpleType), **36**
 stString (classe em AeonDigital\SimpleType), **36**
 sub() (AeonDigital\Realty method), **50**
 sum() (AeonDigital\Realty method), **50**

T

to_system_path() (global function), **16**
 toArray() (AeonDigital\Collection\Collection method), **24**
 toArray() (AeonDigital\Tools method), **56**
 toArrayStr() (AeonDigital\Tools method), **56**
 toBool() (AeonDigital\Tools method), **55**

toDateTime() (*AeonDigitalTools* method), [57](#)
 toDateTimeString() (*AeonDigitalTools* method),
 [57](#)
 toFloat() (*AeonDigitalTools* method), [56](#)
 toInt() (*AeonDigitalTools* method), [55](#)
 toJSON() (*AeonDigitalTools* method), [58](#)
 toNegative() (*AeonDigitalRealty* method), [49](#)
 toNumeric() (*AeonDigitalTools* method), [55](#)
 Tools (*classe em AeonDigital*), [52](#)
 toPositive() (*AeonDigitalRealty* method), [49](#)
 toRealty() (*AeonDigitalTools* method), [58](#)
 toString() (*AeonDigitalSimpleTypeAbstractsBasic* method),
 [27](#)
 toString() (*AeonDigitalSimpleTypeStBool* method),
 [30](#)
 toString() (*AeonDigitalTools* method), [56](#)
 transform() (*AeonDigitalToolsImage* method), [38](#)
 TypeList (*classe em AeonDigitalCollection*), [26](#)

U

undefined (*global constant*), [3](#)
 unpack() (*AeonDigitalToolsZip* method), [43](#)
 unpackTargets() (*AeonDigitalToolsZip* method),
 [44](#)

V

validate() (*AeonDigitalSimpleTypeAbstractsFloat* method),
 [28](#)
 validate() (*AeonDigitalSimpleTypeAbstractsInt* method), [29](#)
 validate() (*AeonDigitalSimpleTypeStBool* method),
 [30](#)
 validate() (*AeonDigitalSimpleTypeStDateTime* method), [32](#)
 validate() (*AeonDigitalSimpleTypeStReal* method),
 [35](#)
 validate() (*AeonDigitalSimpleTypeStString* method), [37](#)
 value() (*AeonDigitalRealty* method), [45](#)
 var_initi_set() (*global function*), [16](#)
 var_is_defined() (*global function*), [17](#)

W

weekdate_count_weeks() (*global function*), [17](#)
 weekdate_get_first_week() (*global function*),
 [17](#)
 weekdate_get_last_week() (*global function*), [18](#)
 weekdate_to_array() (*global function*), [18](#)
 weekdate_to_datetime() (*global function*), [18](#)

Z

Zip (*classe em AeonDigitalTools*), [43](#)