

@return	空の値		
	明示記述用エイリアス。		
@params			
@see	<input type="checkbox"/> 再帰		
@formula	""		

@return	レイアウトTO名		
	省記述用エイリアス。 table occurrence		
@params			
@see	<input type="checkbox"/> 再帰		
@formula	Get (LayoutTableName)		

@return	スクリプト引数		
	省記述用エイリアス。 argument		
@params			
@see	<input type="checkbox"/> 再帰		
@formula	Get (ScriptParameter)		

@return	最終エラー番号		
	省記述用エイリアス。		
@params			
@see	<input type="checkbox"/> 再帰		
@formula	Get (LastError)		

@return	フィールド参照を渡してフィールド名を返す		
	省記述用エイリアス。		
@params	_R	Reference	field reference
@see	<input type="checkbox"/> 再帰		
@formula	GetFieldName（_R）		

@return	URI引数のクエリー文字列を与え、キー名に対応する値を返す		
	webのGETメソッドに使うURL引数と同じ書式で<キー=値&キー=値&...>を与え、<キー>を指定することで<値>を返す。		
@params	_Q	Text	query string
	_K	Text	key name
@see	<input type="checkbox"/> 再帰		
@formula	If (PatternCount ("&" & _Q ; "&" & _K & "=") =0 ; "" ;		
	Let ([_L = Substitute (_Q ; ["¶" ; "¥n"] ; ["&" ; "¶"]) ; _P = Position ("¶" & _L ; "¶" & _K & "=" ; 1 ; 1) ; _Pnext = Position ("¶" & _L & "¶" ; "¶" ; _p +1 ; 1) ; _Row = Middle (_L ; _P ; _Pnext - _P -1) ; _V = Right (_Row ; Length (_Row) - Length (_K) -1)] ; Substitute (_V ; "¥n" ; "¶")))		

@return	スクリプト引数内の「行番号 or キー名」を指定することで値を返す		
	下記用法が非常に多いのでラッパー関数にした。 ・ GetValue() でスクリプト引数の行番号を指定する ・ fnQuery () にスクリプト引数を当ててキー名を指定する		
@params	_K	Text	key name in the script parameter
@see	<input type="checkbox"/> 再帰		
	fnQuery (_Q ; _K)		
@formula	If (_K = GetAsNumber (_K) ; GetValue (Get (ScriptParameter) ; _K) ; fnQuery (Get (ScriptParameter) ; _K))		

@return	該当レコード件数		
	省記述用エイリアス。 count of records in found set.		
@params			
@see	<input type="checkbox"/> 再帰		
@formula	Get (FoundCount)		

@return	制御文字のみURLエンコード		
	UriDecode () とセットで使う。 ・ GetAsUrlEncoded () には対応する逆関数がないため復号化できない。 ・ 制御文字のみエンコード。		
@params	_V	Text	character string
@see	<input type="checkbox"/> 再帰		
@formula	Substitute (_V ; [Char(10) ; "%0D"] ; [Char(13) ; "%0D"] ; ["%" ; "%25"] ; ["¶" ; "%0D"] ; [" " ; "%20"] ; ["!" ; "%21"] ; ["¥" ; "%22"] ; ["#" ; "%23"] ; ["\$" ; "%24"] ; ["&" ; "%26"] ; ["" ; "%27"] ; ["(" ; "%28"] ; [")" ; "%29"] ; ["*" ; "%2A"] ; ["+" ; "%2B"] ; ["," ; "%2C"] ; ["-" ; "%2D"] ; ["." ; "%2E"] ; ["/" ; "%2F"] ; [":" ; "%3A"] ; [";" ; "%3B"] ; ["<" ; "%3C"] ; ["=" ; "%3D"] ; [">" ; "%3E"] ; ["?" ; "%3F"] ; ["@" ; "%40"] ; ["[" ; "%5B"] ; ["¥¥" ; "%5C"] ; ["]" ; "%5D"] ; ["^" ; "%5E"] ; ["`" ; "%60"] ; ["{" ; "%7B"] ; [" " ; "%7C"] ; ["}" ; "%7D"] ; //["~" ; "%7E"])		

@return	制御文字のみURLデコード		
	UriEncode () とセットで使う。 ・ GetAsUrlEncoded () には対応する逆関数がないため復号化できない。 ・ 制御文字のみエンコード。		
@params	_V	Text	character string
@see	<input type="checkbox"/> 再帰		
@formula	Substitute (_V ; ["%25" ; "%"] ; ["%0D" ; "␣"] ; ["%20" ; " "] ; ["%21" ; "!"] ; ["%22" ; "¥"] ; ["%23" ; "#"] ; ["%24" ; "\$"] ; ["%26" ; "&"] ; ["%27" ; "'"] ; ["%28" ; "("] ; ["%29" ; ")"] ; ["%2A" ; "*"] ; ["%2B" ; "+"] ; ["%2C" ; ","] ; ["%2D" ; "-"] ; ["%2E" ; "."] ; ["%2F" ; "/"] ; ["%3A" ; ":"] ; ["%3B" ; ";"] ; ["%3C" ; "<"] ; ["%3D" ; "="] ; ["%3E" ; ">"] ; ["%3F" ; "?"] ; ["%40" ; "@"] ; ["%5B" ; "["] ; ["%5C" ; "¥¥"] ; ["%5D" ; "]"] ; ["%5E" ; "^"] ; ["%60" ; "`"] ; ["%7B" ; "{"] ; ["%7C" ; " "] ; ["%7D" ; "}"] ; //["%7E" ; "~"])		

@return	レコード番号で指定したレコード1行のデータ(改行リストで指定したフィールドのみ)を JSON (連想配列)で返す。		
	数字フィールドに値が無い場合は「""」ではなく 0 を返す仕様にした。 数字フィールドでは「""」と 0 をあえて区別しない。		
@params	_Keys	Text	フィールド名の改行リスト
	_I	Number	レコード番号
@see	<input checked="" type="checkbox"/> 再帰		
	fnSubstring (_V ; _Pst ; _Pov)		
@formula	If (Left (_Keys ; 1) <> " " ; "{") &		
	Let ([_C = ValueCount (_Keys) ; _O = Get (LayoutTableName) & "::" ; _K0 = Trim (GetValue (_Keys ; 1)) ; _K = If (fnSubstring (_K0 ; 1 ; Length (_O) + 1) = _O ; fnSubstring (_K0 ; Length (_O) + 1 ; "") ; _K0) ; _R = GetField (_K) ; _B = (MiddleWords (FieldType (Get (FileName) ; _K) ; 2 ; 1) = "Number") ; _V = GetNthRecord (_R ; _I)] ; Quote (_K) & ":" & If (_B ; Case(Left(_V +0; 1) = "." ; "0" & _V +0 ; Left(_V +0; 2) = "-." ; "-0" & Abs(_V) ; _V +0) ; Quote (GetAsURLEncoded (_V))) & Case (_C =1 ; "}" ; _C >1 ; ", " & jsonNthHash (" " & RightValues (_Keys ; _C -1) ; _I)))		

@return	レコード番号で範囲指定した該当レコードのデータ(改行リストで指定するフィールドのみ)を JSON で返す。		
@params	_Keys	Reference	value list of field names
	_Start	Number	first location
	_End	Number	last location
@see	<input checked="" type="checkbox"/> 再帰		
	jsonNthHash (_Keys ; _I)		
@formula	<pre>If (Left (_Keys ; 1) <> " " ; "[␣]") & Let ([_End = If (Get (FoundCount) < _End ; Get (FoundCount) ; _End) ; _Keys = Trim (_Keys)] ; If (_Start > _End ; "]" ; jsonNthHash (_Keys ; _Start) & Case (_Start = _End ; "␣]" ; _Start < _End ; ",␣" & jsonRowsHash (" " & _Keys ; _Start+1 ; _End))))</pre>		

@return	レコード1行のデータ(レコード番号で指定したフィールドのみ)を JSON (配列)で返す。		
	数字フィールドに値が無い場合は「""」ではなく 0 を返す仕様にした。 数字フィールドでは「""」と 0 をあえて区別しない。		
@params	_Keys	Text	フィールド名の改行リスト
	_I	Number	Nth
@see	<input checked="" type="checkbox"/> 再帰		
	fnSubstring (_V ; _Pst ; _Pov)		
@formula	If (Left (_Keys ; 1) <> " " ; "[") &		
	Let ([_C = ValueCount (_Keys) ; _O = Get (LayoutTableName) & "::" ; _K0 = Trim (GetValue (_Keys ; 1)) ; _K = If (fnSubstring (_K0 ; 1 ; Length (_O) + 1) = _O ; fnSubstring (_K0 ; Length (_O) + 1 ; "") ; _K0) ; _R = GetField (_K) ; _B = (MiddleWords (FieldType (Get (FileName) ; _K) ; 2 ; 1) = "Number") ; _V = GetNthRecord (_R ; _I)] ; If (_B ; Case(Left(_V +0; 1) = "." ; "0" & _V +0 ; Left(_V +0; 2) = "-." ; "-0" & Abs(_V) ; _V +0) ; Quote (GetAsURLEncoded (_V))) & Case (_C =1 ; "]" ; _C >1 ; ", " & jsonNthArray (" " & RightValues (_Keys ; _C -1) ; _I)))		

@return	レコード番号で範囲指定した該当レコードのデータ(改行リストで指定するフィールドのみ)を JSON で返す。		
@params	_Keys	Reference	value list of field names
	_Start	Number	first location
	_End	Number	last location
@see	<input checked="" type="checkbox"/> 再帰		
	jsonNthArray (_Keys ; _I)		
@formula	<pre>If (Left (_Keys ; 1) <> " " ; "[␣]") & Let ([_End = If (Get (FoundCount) < _End ; Get (FoundCount) ; _End) ; _Keys = Trim (_Keys)] ; If (_Start > _End ; "]" ; jsonNthArray (_Keys ; _Start) & Case (_Start = _End ; "␣]" ; _Start < _End ; ",␣" & jsonRowsArray (" " & _Keys ; _Start+1 ; _End))))</pre>		

@return	レコード番号で指定したレコード1行のデータ(改行リストで指定したフィールドのみ)を JSON (連想配列)で返す。		
	数字フィールドに値が無い場合は「""」ではなく 0 を返す仕様にした。 数字フィールドでは「""」と 0 をあえて区別しない。 jsonNthHash との違いは GetAsURLEncoded () の代わりに UrlEncode () を用いて、制御文字以外はエンコードしないことのみ。		
@params	_Keys	Text	フィールド名の改行リスト
	_I	Number	レコード番号
@see	<input checked="" type="checkbox"/> 再帰		
	fnSubstring (_V ; _Pst ; _Pov)		
@formula	If (Left (_Keys ; 1) <> " " ; "{") &		
	Let ([_C = ValueCount (_Keys) ; _O = Get (LayoutTableName) & "::-" ; _K0 = Trim (GetValue (_Keys ; 1)) ; _K = If (fnSubstring (_K0 ; 1 ; Length (_O) + 1) = _O ; fnSubstring (_K0 ; Length (_O) + 1 ; "") ; _K0) ; _R = GetField (_K) ; _B = (MiddleWords (FieldType (Get (FileName) ; _K) ; 2 ; 1) = "Number") ; _V = GetNthRecord (_R ; _I)] ; Quote (_K) & "::-" & If (_B ; Case(Left(_V +0; 1) = "." ; "0" & _V +0 ; Left(_V +0; 2) = "-." ; "-0" & Abs(_V) ; _V +0) ; Quote (UrlEncode (_V))) & Case (_C =1 ; "}" ; _C >1 ; ", " & jsonNthHashLax (" " & RightValues (_Keys ; _C -1) ; _I)))		

@return	レコード1行のデータ(レコード番号で指定したフィールドのみ)を JSON (配列)で返す。		
	数字フィールドに値が無い場合は「""」ではなく 0 を返す仕様にした。 数字フィールドでは「""」と 0 をあえて区別しない。 jsonNthArray との違いは GetAsURLEncoded () の代わりに UrlEncode () を用いて、制御文字以外はエンコードしないことのみ。		
@params	_Keys	Text	フィールド名の改行リスト
	_I	Number	Nth
@see	<input checked="" type="checkbox"/> 再帰		
	fnSubstring (_V ; _Pst ; _Pov)		
@formula	If (Left (_Keys ; 1) <> " " ; "[") & Let ([_C = ValueCount (_Keys) ; _O = Get (LayoutTableName) & "::-" ; _K0 = Trim (GetValue (_Keys ; 1)) ; _K = If (fnSubstring (_K0 ; 1 ; Length (_O) +1) = _O ; fnSubstring (_K0 ; Length (_O) +1 ; "") ; _K0) ; _R = GetField (_K) ; _B = (MiddleWords (FieldType (Get (FileName) ; _K) ; 2 ; 1) = "Number") ; _V = GetNthRecord (_R ; _I)] ; If (_B ; Case(Left(_V +0; 1) = "." ; "0" & _V +0 ; Left(_V +0; 2) = "-." ; "-0" & Abs(_V) ; _V +0) ; Quote (UrlEncode (_V))) & Case (_C =1 ; "]" ; _C >1 ; " , " & jsonNthArrayLax (" " & RightValues (_Keys ; _C -1) ; _I)))		

@return	レコード番号で範囲指定した該当レコードのデータ(改行リストで指定するフィールドのみ)を JSON で返す。		
	jsonRowsArray との違いは GetAsURLEncoded () の代わりに UrlEncode () を用いて、制御文字以外はエンコードしないことのみ。		
@params	_Keys	Reference	value list of field names
	_Start	Number	first location
	_End	Number	last location
@see	<input checked="" type="checkbox"/> 再帰		
	jsonNthArrayLax (_Keys ; _I)		
@formula	<pre>If (Left (_Keys ; 1) <> " " ; "[␣]") & Let ([_End = If (Get (FoundCount) < _End ; Get (FoundCount) ; _End) ; _Keys = Trim (_Keys)] ; If (_Start > _End ; "]" ; jsonNthArrayLax (_Keys ; _Start) & Case (_Start = _End ; "␣]" ; _Start < _End ; ",␣" & jsonRowsArrayLax (" " & _Keys ; _Start+1 ; _End))))</pre>		

@return	レコード番号で範囲指定した該当レコードのデータ(改行リストで指定するフィールドのみ)を JSON で返す。		
	jsonRowsHash との違いは GetAsURLEncoded () の代わりに UriEncode () を用いて、制御文字以外はエンコードしないことのみ。		
@params	_Keys	Reference	value list of field names
	_Start	Number	first location
	_End	Number	last location
@see	<div><input checked="" type="checkbox"/> 再帰</div>		
	jsonNthHashLax (_Keys ; _I)		
@formula	<div>If (Left (_Keys ; 1) <> " " ; "[␣]") &</div> <div>Let (</div> <div>[</div> <div>_End = If (Get (FoundCount) < _End ; Get (FoundCount) ; _End) ;</div> <div>_Keys = Trim (_Keys)</div> <div>];</div> <div>If (</div> <div>_Start > _End ; "]" ;</div> <div>jsonNthHashLax (_Keys ; _Start) &</div> <div>Case (</div> <div>_Start = _End ;</div> <div>"␣]" ;</div> <div>_Start < _End ;</div> <div>" ,␣" & jsonRowsHashLax (" " & _Keys ; _Start+1 ; _End)</div> <div>)</div> <div>)</div> <div>)</div>		