機能	説明	例

ネームスペース	モデル要素を参照する際は、要素名にプレフィックスを付ける必要があります。使用可能なプレフィックス:
	dp: (データプールアイテム)、**ev:**(イベント)、**v:**(ローカル変数)、**f:** (関数)
		dp:x = 100; // set a datapool item
		fire ev:back(); // fire an event
		f:trace_string("hello world"); // call a function

データプールアイテムへのアクセス	データプールアイテムに値を書き込むには、代入式の左辺に配置します。データプールアイテムから値を読み取るには、式の左辺以外の場所に配置します。リダイレクトリンク(=>)はデータプールアイテム配置の特殊形態です。	dp:x = 5; // writing to x
		dp:x = dp:y + dp:z; // reading y and z
		length dp:aList; // read the length of a list datapool item
		dp:refX => dp:x; // redirect link

イベントの送信	構文:
	fire ev:<identifier>(<parameter-list>);
	
	イベントは、タイムアウト後に送信できます。
	この遅延イベントは、**cancel_fire**式を使って取り消すことができます。
	
	構文:
	fire_delayed <timeout>, **ev:**<identifier>(<parameter-list>);
	cancel_fire ev:<identifier>;
		fire ev:back();
		fire ev:mouseClick(10, 20);
		
		fire_delayed 3000, ev:back(); // send the event "back" in 3 seconds.
		cancel_fire ev:back; // cancel the event

イベントへの応答	イベントに応答するには、**match_event**を使用します。**if-then-else**文の特殊形態です。**If**と**else**分岐は、常に同じタイプでなければなりません。代入式の右辺に使う場合は、**else**分岐が必須です。
	
	構文:
	match_event v:<identifier> = **ev:**<identifier>
	in <sequence>
	else <sequence>
		match_event v:event = ev:back in {
		f:trace_string("back event received");
		}
		
		v:this.x = match_event v:event = ev:back in 10 else 0;

イベントパラメータへのアクセス	**match_event**の**in**式で、イベントパラメータにアクセスできます。イベントパラメータにアクセスするには、ドット記法を使用します	match_event v:event = ev:mouseClick in {
		v:this.x = v:event.x;
		v:this.y = v:event.y;
		}

ウィジェットプロパティへのアクセス	スクリプトがウィジェットの一部(ウィジェットアクション、入力への応答)である場合は、スクリプトからそのウィジェットのプロパティにアクセスできます。現在のウィジェットを参照するために、特殊な変数**v:this**が用意されています。ウィジェットプロパティを参照するには、ドット記法を使用します。	v:this.text = "hello world"; v:this.x = 10;

ウィジェットツリーのナビゲーション	スクリプトがウィジェットの一部である場合は、スクリプトから他のウィジェットのプロパティにアクセスできます。ウィジェットツリーナビゲーション演算子:
	->。親ウィジェットへのアクセスに使用する識別子: **^**。
		v:this->^->caption.text = "Play"; // goto parent, goto caption, property text
		v:this->^.x-= 1; // goto parent, property x

文字列の書式設定	**+**演算子で、文字列を連結できます。他の文字列変換関数については、ドキュメンテーションを参照してください。	v:this.text = "current speed: " + f:int2string(dp:speed) + "km/h";
文字列の比較	大文字と小文字を区別して2つの文字列を比較するには、等価演算子**== or !=**を使用します。 大文字と小文字を区別せずに2つの文字列を比較するには、等価演算子**=Aa=**を使用します。	"name" == "NAME" // false "name" != "NAME" // true "name" =Aa= "NAME" // true
言語の変更	EB GUIDEモデルのすべてのデータプールアイテムの言語を変更するには、**language**を使用します。この動作は非同期で実行されます。 構文: **f:language(l:**<identifier>**)**	f:language(l:Standard) // changes language to the standard language f:language(l:German) // changes language to German

機能	説明	例
スキンの変更	EB GUIDEモデルのすべてのデータプールアイテムのスキンを変更するには、**skin**を使用します。この動作は非同期で実行されます。 構文: **f:skin(s:**<identifier>**)**	f:skin(s:Standard) // changes to the standard skin f:skin(s:"myskin") // changes to a user-defined skin
定数	文字列の定数は、引用符で囲まずに記述できます。 色定数は、RGBA形式で指定します。	"hello world" // string constant Napoleon // string constant 5 // integer constant color:0,235,0,255 // EB green
算術、論理、代入演算子	加算と文字列連結: **+**、減算: **-**、乗算: *****、除算: **/**、剰余: **%**、より大きい: **>**、より小さい: **<**、以上: **>=**、以下: **<=**、等しい: **==**、等しくない: **!=**、論理積: **&&**、論理和: **||**、否定: **!**、代入: **=**、加算して代入: **+=**、減算して代入: **-=**	dp:myString = "Hello" + "World"; dp:count += 1; // increment one

シーケンス	シーケンスは、1つの式または一連の式を波括弧で囲んだものです。シーケンスの最後の式が、シーケンス全体の値になります。	if(dp:something)
		  dp:x = 5; // single expression
		if(dp:other) {
		  dp:x = 5; // sequence enclosed
		  dp:y = 10; // in curly braces
		}

ローカル変数	ローカル変数を指定するには、**let**バインディングを使用します。初期値が設定されてない変数は使用できません。
	letバインディングはネストできます。
	
	構文:
	let v:<identifier> = <expression>**;**
	  v:<identifier2> = <expression>**;**
	  ...
	  in <sequence>
		let v:x = 42;
		  v:text = "hello world";
		in {
		  v:this.x = v:x;
		  v:this.text = v:text;
		}

Whileループ	**while**ループは、条件と本文の2つで構成されます。条件が偽になるまで、本文が繰り返し評価されます。
	
	構文:
	while(<expression> **)** <sequence>
		dp:i = 0;
		while(dp:i <= 10) {
		  dp:sum += i;
		  dp:i += 1;
		}

If-then-else	**If-then-else** は、CやJavaの三項条件演算子と似た動作をします。代入式の右辺に使う場合は、**else**分岐が必須で、両方の分岐は同じ型でなければなりません。
	
	構文:
	if(<expression> **)** <sequence> **else** <sequence>
		if(dp:buttonClicked) {
		  v:this.x = dp:x;
		}
		else {
		  v:this.x = 0;
		}
		
		v:this.x = if(dp:buttonClicked) dp:x else 0;

コメント	C言語式のブロックコメントと C++式の行コメントを使用できます。	/* this is a C style block comment */ // this is a C++ style line comment
戻り値	スクリプトの最後の式が戻り値です。 戻り値の型を強制的に無効にするには、**unit**または次を使用します: **{}**	dp:x + 2; // returns datapool item x plus 2