#### if (If)

Like an if statement in a programming language, wraps elements which are to be used under the conditions defined by its attributes.

[Example: Consider the following example of an if element in a DrawingML diagram within the context of a [choose](choose.docx) statement:

<[choose](choose.docx) [name](name.docx)="Name1">

 <if name="Name2" func="var" arg="[dir](dir.docx)" op="equ" [val](val.docx)="[norm](norm.docx)">

 <[alg](alg.docx) type="snake">

 <[param](param.docx) type="grDir" [val](val.docx)="tL"/>

 <[param](param.docx) type="flowDir" [val](val.docx)="[row](row.docx)"/>

 <[param](param.docx) type="contDir" [val](val.docx)="sameDir"/>

 <[param](param.docx) type="[off](off.docx)" [val](val.docx)="ctr"/>

 </[alg](alg.docx)>

 </if>

 <[else](else.docx) [name](name.docx)="Name3">

 <[alg](alg.docx) type="snake">

 <[param](param.docx) type="grDir" [val](val.docx)="tR"/>

 <[param](param.docx) type="flowDir" [val](val.docx)="[row](row.docx)"/>

 <[param](param.docx) type="contDir" [val](val.docx)="sameDir"/>

 <[param](param.docx) type="[off](off.docx)" [val](val.docx)="ctr"/>

 </[alg](alg.docx)>

 </[else](else.docx)>

</[choose](choose.docx)>

In this example, a if element is used to define a set of parameters associated with the snake algorithm when the diagram is in the normal direction. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [choose](choose.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [alg](alg.docx) (Algorithm) | § |
| [choose](choose.docx) (Choose Element) | § |
| [constrLst](constrLst.docx) (Constraint List) | § |
| [extLst](extLst.docx) (Extension List) | § |
| [forEach](forEach.docx) (For Each) | § |
| [layoutNode](layoutNode.docx) (Layout Node) | § |
| [presOf](presOf.docx) (Presentation Of) | § |
| [ruleLst](ruleLst.docx) (Rule List) | § |
| [shape](shape.docx) (Shape) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| arg (Argument) | Specifies the variable to use as part of the test in an if element.  Ignored unless the function attribute is set to "var".The possible values for this attribute are defined by the [ST\_FunctionArgument](ST_FunctionArgument.docx) simple type (§). |
| axis (Axis) | Specifies the axis on which to select data from the data model. [Example: For example, axis="ch" will select children of the current point node and axis="des" will select all descendants. [end](end.docx) example]The possible values for this attribute are defined by the [ST\_AxisTypes](ST_AxisTypes.docx) simple type (§). |
| cnt (Count) | Specifies the count of items to use in a data set.[Example: Consider the following example of a [forEach](forEach.docx) in a DrawingML diagram:<[forEach](forEach.docx) name="Name5" ref="" axis="ch" ptType="node" cnt="2">...</[forEach](forEach.docx)>In this example, [up](up.docx) to two children will be obtained through this [forEach](forEach.docx). [end](end.docx) example]The possible values for this attribute are defined by the [ST\_UnsignedInts](ST_UnsignedInts.docx) simple type (§). |
| func (Function) | The function used to evaluate the if condition.[Example: Consider the following example of func being used in DrawingML:<if name="Name2" func="var" arg="[dir](dir.docx)" op="equ" [val](val.docx)="[norm](norm.docx)">...</if>In this example, func is set to var. [end](end.docx) example]The possible values for this attribute are defined by the [ST\_FunctionType](ST_FunctionType.docx) simple type (§). |
| hideLastTrans (Hide Last Transition) | In algorithms that support transitions, this attribute specifies that the last transition will not be rendered.  This allows for diagrams that [start](start.docx) and [end](end.docx) with a node.The possible values for this attribute are defined by the [ST\_Booleans](ST_Booleans.docx) simple type (§). |
| [name](name.docx) (Name) | A unique identifier for the [layout](layout.docx) node.The function used to evaluate the if condition.[Example: Consider the following example of name being used in DrawingML:<if name="Name2" func="var" arg="[dir](dir.docx)" op="equ" [val](val.docx)="[norm](norm.docx)">...</if>In this example, the name attribute is set to Name2. [end](end.docx) example]The possible values for this attribute are defined by the XML Schema string datatype. |
| op (Operator) | The operator used to evaluate the condition.[Example: Consider the following example of op being used in DrawingML:<if name="Name2" func="var" arg="[dir](dir.docx)" op="equ" [val](val.docx)="[norm](norm.docx)">...</if>In this example, op is being used to test the equality of the argument and value. [end](end.docx) example]The possible values for this attribute are defined by the [ST\_FunctionOperator](ST_FunctionOperator.docx) simple type (§). |
| ptType (Data Point Type) | Specifies the type of data point to select.[Example: Consider the following example of a [forEach](forEach.docx) in a DrawingML diagram:<[forEach](forEach.docx) name="Name5" ref="" axis="ch" ptType="node" cnt="2">...</[forEach](forEach.docx)>In this example, the [forEach](forEach.docx) will select all node type points in the set. [end](end.docx) example]The possible values for this attribute are defined by the [ST\_ElementTypes](ST_ElementTypes.docx) simple type (§). |
| [st](st.docx) (Start) | Specifies where to [start](start.docx) in a data set.[Example: Consider the following example of a [forEach](forEach.docx) in a DrawingML diagram:<[presOf](presOf.docx) axis="desOrSelf" ptType="node" [st](st.docx)="2"/>In this example, the second element in the set will be the first point returned. [end](end.docx) example]The possible values for this attribute are defined by the [ST\_Ints](ST_Ints.docx) simple type (§). |
| step (Step) | Specifies the step to use in a data set. A step with a value of 2 will return every other item in the set.The possible values for this attribute are defined by the [ST\_Ints](ST_Ints.docx) simple type (§). |
| [val](val.docx) (Value) | An absolute value.The possible values for this attribute are defined by the [ST\_FunctionValue](ST_FunctionValue.docx) simple type (§). |

The following XML Schema fragment defines the contents of this element:

<complexType [name](name.docx)="CT\_When">

 <choice minOccurs="0" maxOccurs="unbounded">

 <element name="[alg](alg.docx)" type="CT\_Algorithm" minOccurs="0" maxOccurs="1"/>

 <element name="shape" type="CT\_Shape" minOccurs="0" maxOccurs="1"/>

 <element name="[presOf](presOf.docx)" type="CT\_PresentationOf" minOccurs="0" maxOccurs="1"/>

 <element name="[constrLst](constrLst.docx)" type="CT\_Constraints" minOccurs="0" maxOccurs="1"/>

 <element name="[ruleLst](ruleLst.docx)" type="CT\_Rules" minOccurs="0" maxOccurs="1"/>

 <element [name](name.docx)="[forEach](forEach.docx)" type="CT\_ForEach"/>

 <element [name](name.docx)="[layoutNode](layoutNode.docx)" type="CT\_LayoutNode"/>

 <element [name](name.docx)="[choose](choose.docx)" type="CT\_Choose"/>

 <element name="[extLst](extLst.docx)" type="a:CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

 </choice>

 <attribute name="name" type="xsd:string" use="optional" default=""/>

 <attributeGroup ref="AG\_IteratorAttributes"/>

 <attribute [name](name.docx)="func" type="[ST\_FunctionType](ST_FunctionType.docx)" use="required"/>

 <attribute name="arg" type="[ST\_FunctionArgument](ST_FunctionArgument.docx)" use="optional" default="none"/>

 <attribute [name](name.docx)="op" type="[ST\_FunctionOperator](ST_FunctionOperator.docx)" use="required"/>

 <attribute [name](name.docx)="[val](val.docx)" type="[ST\_FunctionValue](ST_FunctionValue.docx)" use="required"/>

</complexType>