#### choose (Choose Element)

The choose element wraps if/[else](else.docx) blocks into a choose block.

[Example: Consider the following example of a choose element in a DrawingML diagram:

<choose [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](if.docx)>

 <[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>

In this example, a choose element is used to define two different sets of parameters associated with a snake algorithm depending upon the direction in which the user wants the algorithm to flow (RTL or LTR). [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [else](else.docx) (§); [forEach](forEach.docx) (§); if (§); [layoutNode](layoutNode.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [else](else.docx) (Else) | § |
| [if](if.docx) (If) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [name](name.docx) (Name) | A unique name associated with the choose statement.[Example: Consider the following example of a choose element in a DrawingML diagram:<choose [name](name.docx)="Name1">...</choose>In this example, the choose element is named Name1. [end](end.docx) example]The possible values for this attribute are defined by the XML Schema string datatype. |

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

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

 <sequence>

 <element [name](name.docx)="[if](if.docx)" type="CT\_When" maxOccurs="unbounded"/>

 <element [name](name.docx)="[else](else.docx)" type="CT\_Otherwise" minOccurs="0"/>

 </sequence>

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

</complexType>