

***NameOfAttribute*(ArchiNode n)** : return the value of an attribute for a given node
ex : Length(current_node)
Returns the Length attribute value of current_node

NodesFilter(Closure c) : returns a list of nodes in the scene given a conditional closure
ex : NodesFilter({ iteration_node_scene -> Scale(iteration_node_scene) == 2})
Returns the list of node in the scene that have a decomposition level equals to 2

NodesList(int scale=-1, int order=-1, String type="") : returns a list of nodes in the scene given 3 parameters :
int scale : the level of decomposition
int order : the order of ramification
String type : the type name
ex : NodesList(2, -1, "A")
Returns the list of node in the scene that have a decomposition level equals to 2 and a type name equals to "A" (ramification order has the default value and then is not used)

Root(ArchiNode n, int scale=1) : returns the node at the given decomposition level in the path from current node to the Scene root node.
ex : Root(current_node, 1)
Returns the node of the individu (decomposition level = 1) linked to the current_node.

Complex(ArchiNode n, Closure c=null) : returns the complex node of the current node which validates the closure condition (if given)
ex : Complex(current_node, {node -> Scale(node) == 2})
Returns the complex of current_node at decomposition level 2

Components (ArchiNode n, Closure c=null): returns the list of component nodes of the current node which validates the closure condition (if given)
ex : Components(current_node, {node -> Scale(node) == 4})
Returns a list of component nodes at decomposition level 4 from the current_node

Successors(ArchiNode n, Closure c=null) : returns the list of successor nodes of the current node which validates the closure condition (if given)
ex : Successors(current_node, {node -> Branches(node) > 0})
Returns a list of successor nodes from the current_node that have at least one branching

Successor(ArchiNode n, boolean inAxis=true) : return the successor of the current node. inAxis parameter must be true to seek the successor whatever the axis decomposition structure. If false, it will seek the successor only in the current decomposition structure (e.g. in the current growth unit).
ex : Successor(current_node)
Returns the successor of the current_node in the current axis

Predecessors(ArchiNode n, Closure c=null) : returns the list of predecessor nodes of the current node which validates the closure condition (if given)

ex : Predecessors(current_node, {node -> Branches(node) > 0})

Returns a list of predecessor nodes from the current_node that have at least one branching

Predecessor(ArchiNode n, boolean inAxis=true) : return the predecessor of the current node.

inAxis parameter must be true to seek the predecessor whatever the axis decomposition structure. If false, it will seek the predecessor only in the current decomposition structure (e.g. in the current growth unit).

ex : Predecessor(current_node)

Returns the predecessor of the current_node in the current axis

Branches (ArchiNode n, Closure c=null): returns the list of branched nodes of the current node which validates the closure condition (if given)

ex : Branches(current_node, {node -> Scale(node) == 2})

Returns a list of branched nodes at decomposition level 2 from the current_node

Ancestors2 (ArchiNode n, Closure c=null): returns the list of nodes from the current node to the Scene root node, which validates the closure condition (if given)

ex : Ancestors(current_node, {node -> Scale(node) == 3})

Returns a list of nodes at decomposition level 3 in the path from the current_node to root scene node

Descendants (ArchiNode n, Closure c=null): returns the list of nodes in the the subtree rooting from the current node, which validates the closure condition (if given)

ex : Descendants(current_node, {node -> Scale(node) == 3})

Returns a list of nodes at decomposition level 3 in the subtree rooting from current_node

Bearer(ArchiNode n, int order=-1) : returns the bearer node at the given ramification order.

ex : Bearer(current_node, 1)

Returns the bearer node in the axis order 1 of the current_node.

Previous (ArchiNode n, Closure c=null): returns the previous node which validates the closure condition (if given) whatever its topological relation

ex : Previous(current_node, {node -> Scale(node) == 3})

Returns the previous node at decomposition level 3 in the path from the current_node to root scene node

Order(ArchiNode n) : Returns the ramification order of a node (the first axis is order 1)

Scale(ArchiNode n) : Returns the decomposition level of a node (Scene decomposition level is 0)

Rank(ArchiNode n, int complexScale = -1, boolean fromBottom = true) : Returns the rank (topological position) of a node in the complex. In the case of multiple decomposition level, the scale of the complex can be given. fromBottom must be set to true to count the rank from the bottom of the structure.

ex : Rank(current_node, 2, false)

Returns the topological position of current_node in the axis complex (scale 2) starting from the top of this axis. (begins from 0)

TypeName(ArchiNode n) : Returns the type of a node

Table(Object[][] data, String [] columnNames) : Display a table with the given data and column names

ex :

*//get the list of nodes for scale = 4, type = "UC" and ramification order = 1
nodes = NodesList(4, 1, "UC")*

*//collect the position in scale 2, the Length attribute and the Diameter attribute
res = nodes.collect({n -> [Position(n,2), Length(n), Diameter(n)]})*

//display the result in a table

Table(res,["Position", "Length", "Diameter"])