

## 12.1.41 Editors - Geometry Nodes Editor - Header - Add Menu - Utilities

### Table of content

Detailed table of content.....	1
Add menu - Utilities.....	3
Index Switch.....	3
Menu Switch.....	4
Random Value.....	5
Repeat Zone.....	6
Switch.....	8

### Detailed table of content

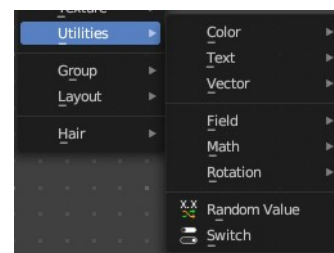
### Detailed table of content

Detailed table of content.....	1
Add menu - Utilities.....	3
Index Switch.....	3
Inputs.....	3
Index.....	3
0 , 1 .....	3
Properties.....	3
Data Type.....	3
Outputs.....	3
Output.....	3
Menu Switch.....	4
Inputs.....	4
Menu.....	4
A , B .....	4
Properties.....	5
Data Type.....	5
Outputs.....	5
Output.....	5
Random Value.....	5
Input.....	5
Min.....	5
Max.....	5
ID.....	5
Seed.....	5
Properties.....	6
Data Type.....	6
Output.....	6
Value.....	6
Repeat Zone.....	6
Repeat Zone Input.....	7
Input.....	8
Iterations.....	8
Geometry.....	8

- Output.....8
  - Geometry.....8
- Repeat Zone Output.....8
  - Input.....8
    - Geometry.....8
  - Output.....8
    - Geometry.....8
- Switch.....8
  - Inputs.....8
    - Switch.....8
      - A.....8
      - B.....8
  - Properties.....9
    - Input Type.....9
  - Outputs.....9
    - Output.....9

## Add menu - Utilities

Utility nodes are mainly for mathematical operations.



## Index Switch

The Index Switch node is meant as a simpler version of the Menu Switch node. It doesn't allow naming items or displaying them in a dropdown, but still allows choosing between an arbitrary number of items, unlike the regular Switch node where you can just switch between two states.

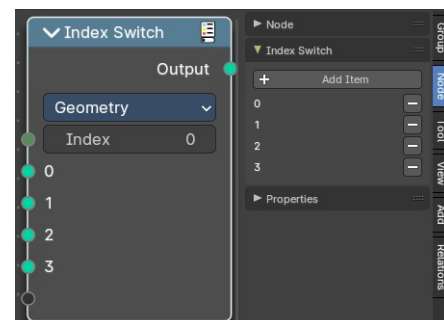
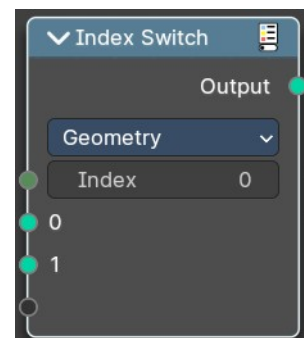
### Inputs

#### *Index*

Wich input to choose.

#### *0, 1 ...*

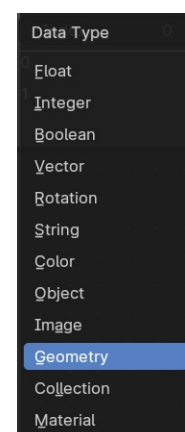
The available input index types that you want to switch between. You can define more input sockets in the sidebar.



### Properties

#### *Data Type*

What input type to evaluate.



### Outputs

#### *Output*

The output index.

## Menu Switch

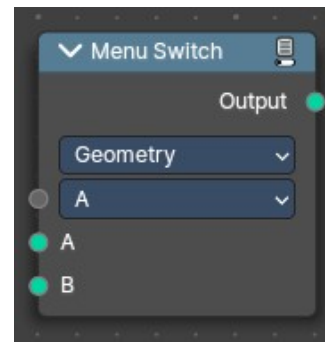
The Menu Switch node selects and computes one input based on a user-defined menu. Only the selected input is computed.

Menu entries can be added, removed, renamed, and reordered. Renaming keeps existing input links.

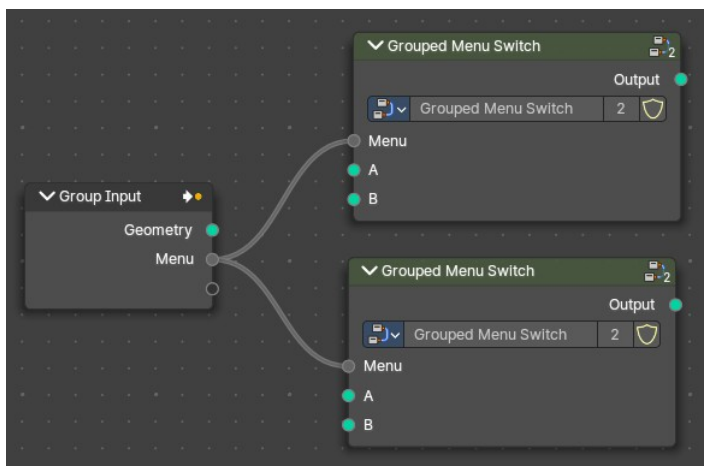
The menu can be used in node groups and the nodes modifier UI. Connecting the menu to a Group Input node exposes it as a group input. A menu needs to be connected to a Menu Switch node to work. An unconnected menu is empty.

Connecting multiple Menu Switch nodes to the same output creates a conflict. To avoid this, a menu switch can be grouped. Multiple groups of the same type can connect to the same menu.

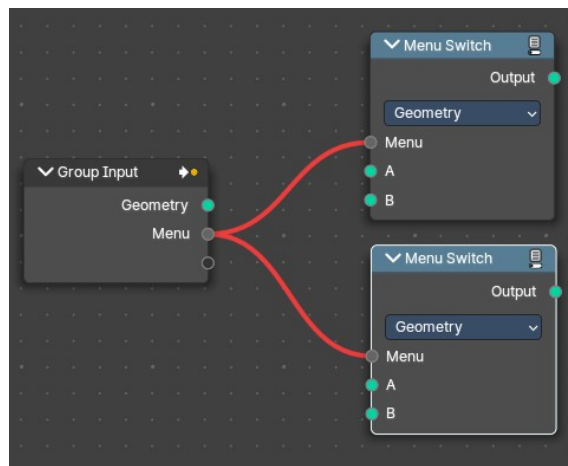
**Note:** You cannot plug in the same value socket into the Menu selector. If you'd like to use the same value for various Menu Switches, group the Menu Switch into a Node Group and use the top-level switch override.



Top Level Grouped Menu Switches with same inputs:



Conflict caused by Same Level menu Switch inputs:



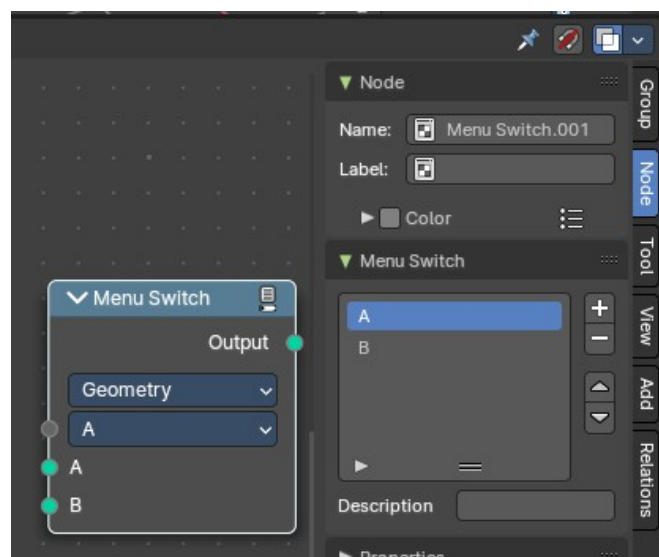
## Inputs

### Menu

Determines which of the input options will be selected and passed through.

### A, B ...

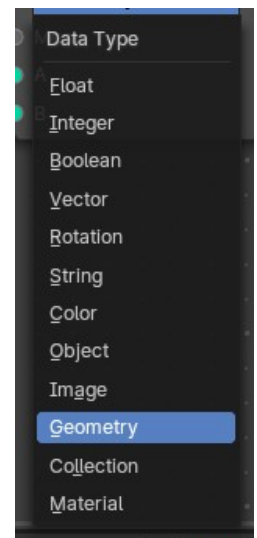
The available input index types that you want to switch between. You can define more input sockets in the sidebar Menu Switch panel.



## Properties

### *Data Type*

Determines the type of the data that is handled by the node.



## Outputs

### *Output*

The output index.

## Random Value

Generates a random value.

### Input

#### Min

The minimum value of the range. This input is only available for Float, Integer, and Vector types.

#### Max

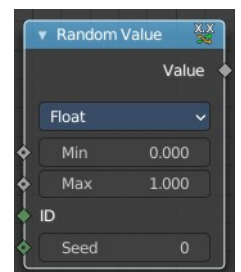
The maximum value of the range. This input is only available for Float, Integer, and Vector types.

#### ID

An ID to drive the random number generator seed. By default, this input uses the same value as if the ID Node, which is the id attribute of the context geometry if it exists, and otherwise the index.

#### Seed

The random seed for the random number generation.



## Properties

### *Data Type*

What kind of random value to create. The items should be self explaining.

<u>F</u> loat	0.000
<u>I</u> nteger	1.000
<u>V</u> ector	
Boolean	0
Data Type	

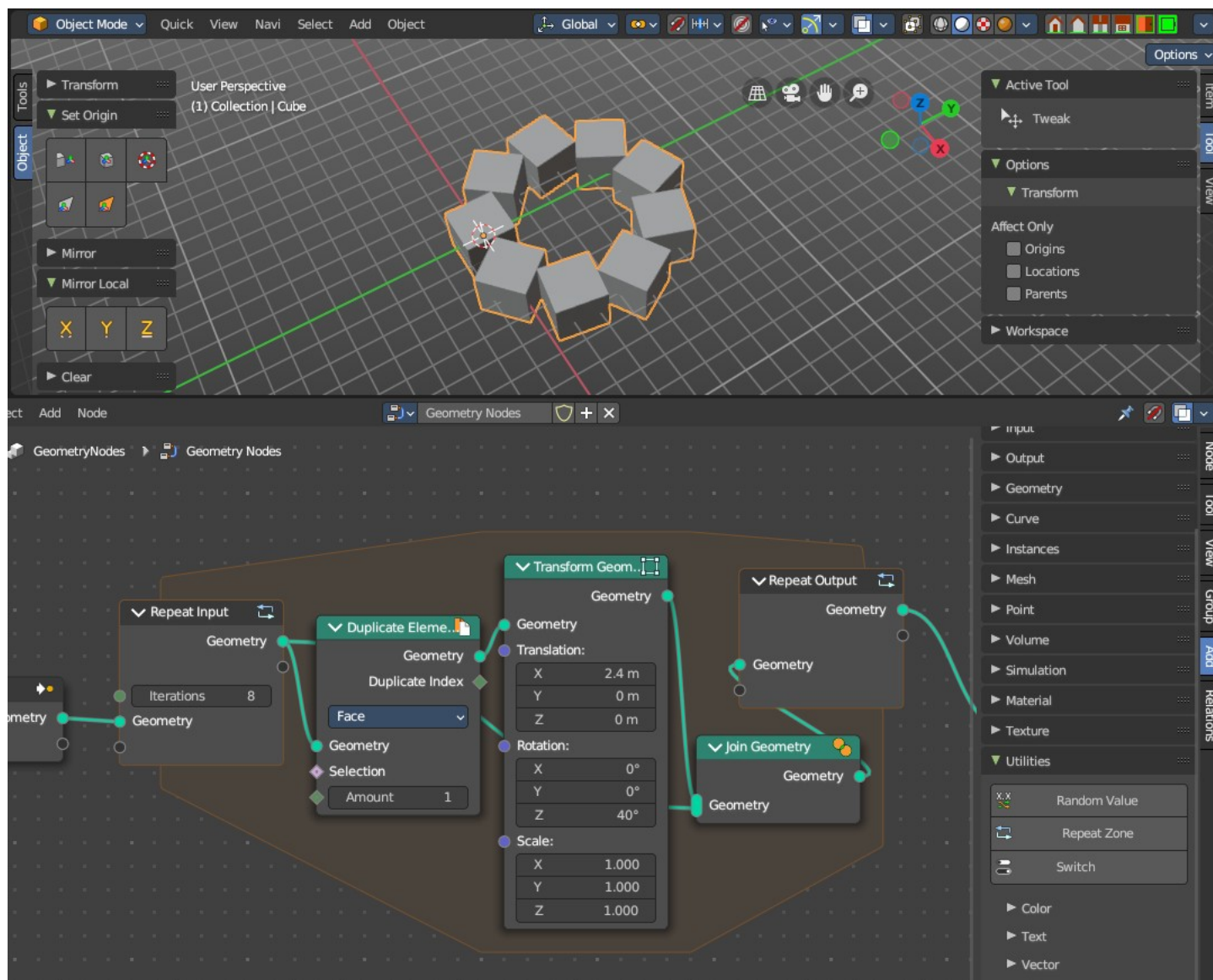
## Output

### *Value*

The output value.

---

## Repeat Zone



When adding a repeat zone, two nodes are added with a “zone” set between them. The inputs connected to the *Repeat Input* node reads and gets data at the beginning of the loop before starting the looping - then the data is processed within the zone, here you can set any changes to the data to then repeat the execution again at the beginning of the chain for the next iteration of the loop. This chain of operations is repeated the specified number of times in the *Repeat Input* node.

In the example in the image above, we duplicate the cube, transform it with a rotation, then join it together again. We do this 8 times in a loop, creating a circular array.

**Note:** It is not possible to set data outside the Repeat Zone, you can only get data from outside the Repeat Zone. Any data connected from the outside of the zone are constant throughout every iteration based on their value at the current frame. The result of the looping can only be accessed via the Repeat Output node.

## Repeat Zone Input

The beginning of the iteration or loop.

## Input

### ***Iterations***

Number of repetitions or loops.

### ***Geometry***

Standard geometry input.

## Output

### ***Geometry***

Standard geometry output.

## ***Repeat Zone Output***

The result and output of the iteration or loop. You can define custom attributes outputs here from the

## Input

### ***Geometry***

Standard geometry output.

## Output

### ***Geometry***

Standard geometry output.

---

## Switch

Switch between two inputs values based on a boolean.

## Inputs

### ***Switch***

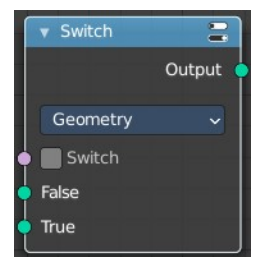
The boolean switch.

### ***A***

The input value A. Used when the switch is off.

### ***B***

The input value B. Used when the switch is on.

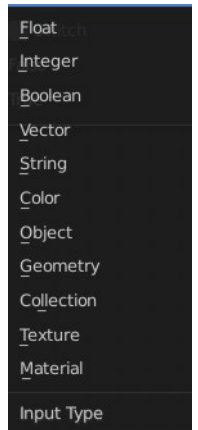




## Properties

### *Input Type*

What input type the values are, which defines what type to output then.



## Outputs

### *Output*

Numerical value output.