



-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____



 Elementor 

condition 



```
$this->add_control(  
    'unique-control-name',  
    [  
        'label' => esc_html__( 'Control Label', 'textdomain' ),  
        'type' => \Elementor\Controls_Manager::TEXT,  
        'condition' => [  
            'dependent-control-name' => 'exact-value',  
        ],  
    ]  
);
```



XXXXXXXXXX

```
$this->add_control(  
    'unique-control-name',  
    [  
        'label' => esc_html__( 'Control Label', 'textdomain' ),  
        'type' => \Elementor\Controls_Manager::TEXT,  
        'condition' => [  
            'dependent-control-name' => [ 'value-1', 'value-2' ],  
        ],  
    ]  
);
```

XXXXXXXXXX

Elementor XXXX

OR (||) XXXX



XXXXXXXXXXXXXXXXXXXX

condition XXXXX

array XXXXX

AND (&&)

XXXXXXXXXX

XX

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

```
$this->add_control(  
    'unique-control-name',
```



```

    []'label' => esc_html__( 'Border Style', 'textdomain' ),
    []'type' => \Elementor\Controls_Manager::SELECT,
    []'options' => [
        []'' => esc_html__( 'None', 'textdomain' ),
        []'solid' => esc_html__( 'Solid', 'textdomain' ),
        []'double' => esc_html__( 'Double', 'textdomain' ),
        []'dotted' => esc_html__( 'Dotted', 'textdomain' ),
        []'dashed' => esc_html__( 'Dashed', 'textdomain' ),
        []'groove' => esc_html__( 'Groove', 'textdomain' ),
    ],
    []'selectors' => [
        [][]'{{WRAPPER}} .inner_class' => 'border-style: {{VALUE}}',
    ],
    []'condition' => [
        [][]'border' => 'yes',
    ],

    []
);

```

```

$this->add_control(
    []'border_color',
    [][
        []'label' => esc_html__( 'Color', 'textdomain' ),
        []'type' => \Elementor\Controls_Manager::COLOR,
        []'default' => '',
        []'selectors' => [
            [][]'{{WRAPPER}} .inner_class' => 'border-color: {{VALUE}}',
        ],
        []'condition' => [
            [][]'border' => 'yes',
            [][]'border_style!' => '',
        ],
    ],
    []
);

```

```

$this->add_responsive_control(
    []'border_width',
    [][
        []'label' => esc_html__( 'Width', 'textdomain' ),
    ],
);

```

```

    'type' => \Elementor\Controls_Manager::DIMENSIONS,
    'selectors' => [
        '{{WRAPPER}} .inner_class' => 'border-width: {{TOP}}{{UNIT}} {{RIGHT}}{{UNIT}}
        {{BOTTOM}}{{UNIT}} {{LEFT}}{{UNIT}};',
    ],
    'condition' => [
        'border' => 'yes',
        'border_style!' => '',
    ],
]
);

```

##

Elementor ***** **s** `conditions` `condition`

###

`conditions`

```

<table>
<thead>
<tr>
<th>/th>
<th>/th>
</tr>
</thead>
<tbody>
<tr>
<th><code>relation</code></th>
<td>(<code>string</code> <code>and</code> / <code>or</code> )<br>
<code>and</code></td>
</tr>
<tr>
<th><code>terms</code></th>
<td>(<code>array</code>
<li><code>name</code></li>
<li><code>value</code></li>
<li><code>operator</code> <code>==</code>, <code>!=</code>, <code>!==</code>,

```

```

<code>in</code>, <code>!in</code>, <code>contains</code>, <code>!contains</code>,
<code><</code>, <code><=</code>, <code>>>/code>, <code>>=</code>, <code>===</code>
<code>===</code>
</li>
</ul>
</td>
</tr>
</tbody>
</table>

```

More Operators

Before implementing the 'conditions' argument, you could only use equality and inequality checks. Now you have many more operators at your disposal: `==`, `!=`, `!==`, `in`, `!in`, `contains`, `!contains`, `<`, `<=`, `>`, `>=`, and `===`.

We can replace the simple `condition` argument:

```

```php
'condition' => [
 ['control-name!' => 'control-value',
],

```

with the advanced `conditions` argument, and include the `operator` rule inside the `term` array:

```

'conditions' => [
 ['terms' => [
 [
 ['name' => 'control-name',
 ['operator' => '!=',
 ['value' => 'control-value',
],
],
],
],

```

This also makes it easier to use equality operators for numeric values (`<`, `<=`, `>`, `>=`, `==`, `===`, `!=` and `!==`):

```

'conditions' => [
 ['terms' => [
 [

```

```

 {
 'name' => 'spacing',
 'operator' => '>=',
 'value' => 0,
 },
],
],
],

```

Below is an example of checking against multiple values:

```
'conditions' => [
 []'terms' => [
 [][
 [][]'name' => 'background_type',
 [][]'operator' => 'in',
 [][]'value' => ['classic', 'gradient', 'video', 'slideshow'],
 []],
 []],
]
```

Below is an example of checking if the term contains a value:

```
'conditions' => [
 []'terms' => [
 [][
 [][]'name' => 'heading',
 [][]'operator' => 'contains',
 [][]'value' => 'elementor',
],
],
],
```



□□□□□□ □ condition □□□□□□□□□□

```
'condition' => [
 []'border' => 'yes',
 []'border_style!' => '',
],
```

conditions

```
'conditions' => [
 'terms' => [
 [
 'name' => 'border',
 'operator' => '===',
 'value' => 'yes',
],
 [
 'name' => 'border_style',
 'operator' => '!=',
 'value' => '',
],
],
],
```

Diagram illustrating a join condition. The condition is represented by a box labeled "condition" with 16 slots. The relation is represented by a box labeled "relation" with 8 slots. The condition box is connected to the relation box by a line.

`conditions`

--	--	--	--

[illegible]

```
'conditions' => [
 []'relation' => 'or',
 []'terms' => [
 [][
 [][]'name' => 'background',
 [][]'operator' => '!=',
 [][]'value' => '',
],
 [][
 [][]'name' => 'border',
 [][]'operator' => '!=',
 [][]'value' => '',
],
],
],
],
```



```
$this->add_control(
 'display_list',
 [
 'label' => esc_html__('Display List', 'textdomain'),
 'type' => \Elementor\Controls_Manager::SWITCHER,
 'label_on' => esc_html__('Yes', 'textdomain'),
 'label_off' => esc_html__('No', 'textdomain'),
 'return_value' => 'yes',
 'default' => 'yes',
]
);

$this->add_control(
 'list',
 [
 'label' => esc_html__('Repeater List', 'textdomain'),
 'type' => \Elementor\Controls_Manager::REPEATER,
 'fields' => [
 [
 'name' => 'list_title',
 'label' => esc_html__('Title', 'textdomain'),
 'type' => \Elementor\Controls_Manager::TEXT,
],
 [
 'name' => 'list_content',
 'label' => esc_html__('Content', 'textdomain'),
 'type' => \Elementor\Controls_Manager::WYSIWYG,
],
],
 'condition' => [
 'display_list' => 'yes',
],
]
);
```



XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
$this->add_control(
 'display_content',
 [
 'label' => esc_html__('Display Content', 'textdomain'),
 'type' => \Elementor\Controls_Manager::SWITCHER,
 'label_on' => esc_html__('Yes', 'textdomain'),
 'label_off' => esc_html__('No', 'textdomain'),
 'return_value' => 'yes',
 'default' => 'yes',
]
);

$this->add_control(
 'list',
 [
 'label' => esc_html__('Repeater List', 'textdomain'),
 'type' => \Elementor\Controls_Manager::REPEATER,
 'fields' => [
 [
 'name' => 'list_title',
 'label' => esc_html__('Title', 'textdomain'),
 'type' => \Elementor\Controls_Manager::TEXT,
],
 [
 'name' => 'list_content',
 'label' => esc_html__('Content', 'textdomain'),
 'type' => \Elementor\Controls_Manager::WYSIWYG,
 'condition' => [
 'display_content' => 'yes',
],
],
],
]
);
```

XXXXXXXXXXXX

## #2

2026-03-26 17:08:44 UTC

2026-03-26 17:08:44 UTC