Rendering Style

- Styling Elements
- Selector Tokens

A common widget rendering use case is using the control value to style different HTML elements. Usually used by controls from the content tab.

Styling Elements

Let's say you have a widget with two controls in which users can set a title, and control the element "color".

```
<?php
class Elementor_Test_Widget extends \Elementor\Widget_Base {

	protected function register_controls(): void {

		$this->start_controls_section(
			'section_content',
			[
				'label' => esc_html__( 'Content', 'textdomain' ),
				'tab' => \Elementor\Controls_Manager::TAB_CONTENT,
			]
		);

		$this->add_control(
			'title',
			[
				'label' => esc_html__( 'Title', 'textdomain' ),
				'type' => \Elementor\Controls_Manager::TEXT,
```

```
        'placeholder' => esc_html__( 'Enter your title', 'textdomain' ),

    ]
    );

    $this->end_controls_section();

    $this->start_controls_section(
    'section_style',
    [
    'label' => esc_html__( 'Style', 'textdomain' ),
    'tab' => \Elementor\Controls_Manager::TAB_STYLE,
    ]
    );

    $this->add_control(
    'color',
    [
    'label' => esc_html__( 'Color', 'textdomain' ),
    'type' => \Elementor\Controls_Manager::COLOR,
    'default' => '#f00',
    'selectors' => [
    '{{WRAPPER}} h3' => 'color: {{VALUE}}',
    ],
    ]
    );

    $this->end_controls_section();

}

protected function render(): void {
    $settings = $this->get_settings_for_display();

    if ( empty( $settings['title'] ) ) {
        return;
    }
    ?>
    <h3>
        <?php echo $settings['title']; ?>
```

```
□□</h3>
□□<?php
□}

□protected function content_template(): void {
□□?>
□□<#
□□if ( '' === settings.title ) {
□□□return;
□□}
□□#>
□□<h3>
□□□{{{ settings.title }}}
□□</h3>
□□<?php
□}

}
```

Note that each element on the page includes a set of style settings that are printed as CSS rules under a unique element selector. The widget can easily use the returned control value as a design rule.

Our test widget will add custom CSS rules to the page's CSS file. Those rules will only affect this widget as the `{ { WRAPPER } }` prefix means that only this instance of the widget will use this color.

Selector Tokens

When you select which category to assign your widget to, you can select one of the default categories:

Label	Description
{ { WRAPPER } }	Widget wrapper element
{ { VALUE } }	Control value
{ { UNIT } }	Unit control value of selected unit
{ { URL } }	The URL returned by the media control
{ { SELECTOR } }	Group control CSS selector

□□□ #2
□ □□□ □□□ 2026-03-26 17:08:44 UTC

□ □□□ □□□ 2026-03-26 17:08:44 UTC