Deprecated Arguments

- Use Cases
- How To Deprecate Arguments
- Updating Deprecated Arguments

Elementor offers the `Deprecation::deprecated_argument()` method to handle the deprecation process for arguments inside the Elementor code.

Use Cases

As our code improves, some arguments become redundant. The same functionality can be processed by retrieving the data internally, without the need of external arguments.

For example, when registering controls, you no longer need to pass the ID argument as the data comes from the class itself, not from a separate method.

How To Deprecate Arguments

Our class has a `register()` method. This method used to accept an `$id`, but now the ID is part of the class and the `$id` argument will eventually not be supported in the registration process.

For example, take the following code:

```
class Items_Manager {

	private $items;

	function register( $instance, $id = null ): void {
		$this->items[ $id ] = $instance;
	}
```

```
}
```

The original method above required the `$id` argument - the new method should work with or without this argument.

The new code, with the deprecation handler, will look like this:

```
use Elementor\Plugin;

class Items_Manager {

	private $items;

	function register( $instance, $id = null ): void {
		if ( $id ) {
			// TODO: Remove this in the future.
			Plugin::instance()->modules_manager->get_modules( 'dev-tools' )->deprecation->deprecated_argument(
				'$id',
				'3.5.0'
			);
		} else {
			// TODO: Leave this in the future.
			$id = $instance->get_id();
		}

		$this->items[ $id ] = $instance;
	}

}
```

The `register()` method will work with or without the `$id` argument. In addition, the deprecation handler will sending some notifications to developers still using the deprecated argument.

After eight major versions, the code will look like this:

```
class Items_Manager {

	private $items;
```

```
□function register( $instance ): void {
□□$id = $instance->get_id();
□□$this->items[ $id ] = $instance;
□}

}
```

Updating Deprecated Arguments

Addon developers who use the `register()` method with the `$id` argument should simply remove the argument, as follows:

```
function register_new_items( $items_manager ) {

□require_once( __DIR__ . '/items/item-1.php' );
□require_once( __DIR__ . '/items/item-2.php' );

-□$items_manager->register( new \My_Item_1(), 'item-1' );
-□$items_manager->register( new \My_Item_1(), 'item-2' );
+□$items_manager->register( new \My_Item_1() );
+□$items_manager->register( new \My_Item_2() );

}
add_action( 'items/register', 'register_new_items' );
```

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