

Inspection

Inspect blank values without coercion, whitespace normalization or recursive traversal

Anonymous · 26/10/2026

ValueInspector

`ValueInspector::isBlank(mixed $value): bool` returns true for exactly three values: `null`, `''` and `[]`.

```
use Derafu\Value\ValueInspector;

ValueInspector::isBlank(null);    // true
ValueInspector::isBlank('');      // true
ValueInspector::isBlank([]);      // true
ValueInspector::isBlank(false);   // false
ValueInspector::isBlank(0);        // false
ValueInspector::isBlank('0');     // false
ValueInspector::isBlank(' ');     // false
ValueInspector::isBlank([null]);  // false
```

This is an inspection of the supplied value, not PHP's `empty()` behavior. It does not determine whether a key exists, whether a value was supplied by a caller or whether an application requires it.

Whitespace is not trimmed, array elements are not inspected recursively, and objects are neither counted nor converted to text. Even an empty `Countable` object is treated as an object rather than as `[]`.

Inspection and Interpretation

The parser performs conversions; the inspector does not:

```
use Derafu\Value\ValueInspector;
use Derafu\Value\ValueParser;

ValueInspector::isBlank('null'); // false; text exists
ValueParser::parse('null');      // null; a recognized literal
ValueInspector::isBlank(' ');    // false; the string contains a space
ValueParser::parse(' ');         // false; boolean interpretation
```

Choose the operation explicitly. A blank check does not imply a type conversion, and parsing does not

decide application-level requiredness.

Last updated on 26/10/2026