

# Package ‘pylintR’

May 9, 2026

**Type** Package

**Title** Lint 'Python' Files with a R Command or a 'RStudio' Addin

**Version** 0.1.0

**Author** Stéphane Laurent

**Maintainer** Stéphane Laurent <laurent\_step@outlook.fr>

**Description** Allow to run 'pylint' on Python files with a R command or a 'RStudio' addin. The report appears in the RStudio viewer pane as a formatted HTML file.

**License** GPL-3

**Encoding** UTF-8

**SystemRequirements** pylint

**URL** <https://github.com/stla/pylintR>

**BugReports** <https://github.com/stla/pylintR/issues>

**Imports** fansi, utils, htmlwidgets, rstudioapi

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-10-12 09:00:01 UTC

## Contents

pylint . . . . .	2
pylintR-shiny . . . . .	2
<b>Index</b>	<b>4</b>

---

pylint

*Run 'pylint' on a file or a folder*

---

### Description

Run 'pylint' on a file or a folder.

### Usage

```
pylint(modules)
```

### Arguments

modules            one or more Python files, or a folder containing Python files

### Value

A htmlwidget object.

### Examples

```
## Not run:
sample_code_file <- system.file("sample_code.py", package = "pylintR")
code_lines <- readLines(sample_code_file)
nlines <- length(code_lines)
# Here is the code:
cat(paste0(format(seq_len(nlines), width = 2), ") ", code_lines), sep = "\n")
# let's copy this Python file in a temporary folder
file_copy <- tempfile(fileext = ".py")
file.copy(sample_code_file, file_copy)
wd <- setwd(tempdir())
# let's lint it with pylint:
pylint(basename(file_copy))
# restore current directory
setwd(wd)

## End(Not run)
```

---

pylintR-shiny

*Shiny bindings for pylintR*

---

### Description

Output and render functions for using 'pylintR' within Shiny applications and interactive Rmd documents.

**Usage**

```
pylintROutput(outputId, width = "100%", height = "400px")
```

```
renderPylintR(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

<code>outputId</code>	output variable to read from
<code>width, height</code>	a valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended
<code>expr</code>	an expression that generates an output of <code>pylint</code>
<code>env</code>	the environment in which to evaluate <code>expr</code>
<code>quoted</code>	whether <code>expr</code> is a quoted expression (with <code>quote()</code> ); this is useful if you want to save an expression in a variable

**Value**

`pylintROutput` returns an output element that can be included in a Shiny UI definition, and `renderPylintR` returns a `shiny.render.function` object that can be included in a Shiny server definition.

# Index

`pylint`, [2](#), [3](#)  
`pylintR-shiny`, [2](#)  
`pylintROutput` (`pylintR-shiny`), [2](#)  
`renderPylintR` (`pylintR-shiny`), [2](#)