

Pound Sterling amount

```
^-?£?(\\d{1,3}(,\\d{3})+|\\d+)(\\.\\d{2})?$/
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
-?	optional (zero or one): the literal "-"
£?	optional (zero or one): the literal "£"
(	start of a capturing group
\\d{1,3}	between 1 and 3 times: digit (0–9)
(	start of a capturing group
,	literal text ","
\\d{3}	exactly 3 times: digit (0–9)
)+	end of group, one or more
	OR — try the alternative
\\d+	one or more: digit (0–9)
)	end of group
(	start of a capturing group
\\.	literal "."
\\d{2}	exactly 2 times: digit (0–9)
)?	end of group, optional (zero or one)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$", value)</pre>
Java	<pre>Pattern.compile("^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$");</pre>
Go	<pre>regexp.MustCompile(`^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$`).MatchString(value)</pre>
Ruby	<pre>/^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^-?£?(\\d{1,3}(,\\d{3})+ \\d+)(\\.\\d{2})?\$~', \$value);</pre>

Common pitfalls

- **Anchored to the whole string.** This pattern uses ^ and \$, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.

- **Watch for backtracking.** This pattern nests quantifiers; on adversarial input that can cause exponential backtracking (ReDoS). Test with long non-matching strings, or use an RE2-based engine.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (`\\d`); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the pound sterling amount against a source of truth (database, API, or checksum) where it matters.