

# South African phone (+27)

South African phone in +27 format.

```
^(\+27|0)[\s\-]?[0-9]{2}[\s\-]?[0-9]{3}[\s\-]?[0-9]{4}$
```

## Token-by-token breakdown

| Token    | Meaning                                         |
|----------|-------------------------------------------------|
| ^        | start of string (or line in multiline mode)     |
| (        | start of a capturing group                      |
| \+       | literal “+”                                     |
| 27       | literal text “27”                               |
|          | OR — try the alternative                        |
| 0        | literal text “0”                                |
| )        | end of group                                    |
| [\s\-]?  | optional (zero or one): any of: whitespace, “-” |
| [0-9]{2} | exactly 2 times: digit (0–9)                    |
| [\s\-]?  | optional (zero or one): any of: whitespace, “-” |
| [0-9]{3} | exactly 3 times: digit (0–9)                    |
| [\s\-]?  | optional (zero or one): any of: whitespace, “-” |
| [0-9]{4} | exactly 4 times: digit (0–9)                    |
| \$       | end of string (or line in multiline mode)       |

## Quick usage in 7 languages

|                   |                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>JavaScript</b> | <pre>const re = /^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$/;<br/>re.test(value);</pre>                  |
| <b>Python</b>     | <pre>import re<br/>re.match(r"^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$", value)</pre>                  |
| <b>Java</b>       | <pre>Pattern.compile("^(\+27 0)[\s\\-]?[0-9]{2}[\s\\-]?[0-9]{3}[\s\\-]?[0-9]{4}\$").matcher(value).matches();</pre> |
| <b>C# / .NET</b>  | <pre>Regex.IsMatch(value, @"^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$");</pre>                          |
| <b>Go</b>         | <pre>regexp.MustCompile(`^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$`).MatchString(value)</pre>           |
| <b>Ruby</b>       | <pre>/^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$/.match?(value)</pre>                                    |
| <b>PHP</b>        | <pre>preg_match('~^(\+27 0)[\s-]?[0-9]{2}[\s-]?[0-9]{3}[\s-]?[0-9]{4}\$~', \$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.
- **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 south african phone (+27) against a source of truth (database, API, or checksum) where it matters.