REGEX CHEAT SHEET

regexguide.com/patterns/timezone-offset.html

# Timezone offset

UTC offset in ±HH:MM or ±HHMM format.

```
^[+-](0\d|1[0-4]):?[0-5]\d$
```

## Token-by-token breakdown

| Token | Meaning |
|---|---|
| `^` | start of string (or line in multiline mode) |
| `[+-]` | any of: “+”, “-” |
| `(` | start of a capturing group |
| `0` | literal text “0” |
| `\d` | any digit (0–9) |
| `\|` | OR — try the alternative |
| `1` | literal text “1” |
| `[0-4]` | any of: 0–4 |
| `)` | end of group |
| `:?` | optional (zero or one): the literal “:” |
| `[0-5]` | any of: 0–5 |
| `\d` | any digit (0–9) |
| `$` | end of string (or line in multiline mode) |

## Quick usage in 7 languages

**JavaScript**

```
const re = /^[+-](0\d|1[0-4]):?[0-5]\d$/;
re.test(value);
```

**Python**

```
import re
re.match(r"^[+-](0\d|1[0-4]):?[0-5]\d$", value)
```

**Java**

```
Pattern.compile("^[+-](0\\d|1[0-4]):?[0-5]\\d$").matcher(value).matches();
```

**C# / .NET**

```
Regex.IsMatch(value, @"^[+-](0\d|1[0-4]):?[0-5]\d$");
```

**Go**

```
regexp.MustCompile(`^[+-](0\d|1[0-4]):?[0-5]\d$`).MatchString(value)
```

**Ruby**

```
/^[+-](0\d|1[0-4]):?[0-5]\d$/.match?(value)
```

**PHP**

```
preg_match('~^[+-](0\d|1[0-4]):?[0-5]\d$~', $value);
```

## 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 timezone offset against a source of truth (database, API, or checksum) where it matters.