

# UGC-NET roll number

12-digit roll number for the UGC-NET (lectureship) exam.

```
^\d{12}$
```

## Token-by-token breakdown

| Token               | Meaning                                     |
|---------------------|---------------------------------------------|
| <code>^</code>      | start of string (or line in multiline mode) |
| <code>\d{12}</code> | exactly 12 times: digit (0–9)               |
| <code>\$</code>     | end of string (or line in multiline mode)   |

## Quick usage in 7 languages

|                   |                                                                   |
|-------------------|-------------------------------------------------------------------|
| <b>JavaScript</b> | <pre>const re = /^\d{12}\$/;<br/>re.test(value);</pre>            |
| <b>Python</b>     | <pre>import re<br/>re.match(r"^\d{12}\$", value)</pre>            |
| <b>Java</b>       | <pre>Pattern.compile("^\d{12}\$").matcher(value).matches();</pre> |
| <b>C# / .NET</b>  | <pre>Regex.IsMatch(value, @"^\d{12}\$");</pre>                    |
| <b>Go</b>         | <pre>regexp.MustCompile(`^\d{12}\$`).MatchString(value)</pre>     |
| <b>Ruby</b>       | <pre>/^\d{12}\$/ .match?(value)</pre>                             |
| <b>PHP</b>        | <pre>preg_match('~^\d{12}\$~', \$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 ugc-net roll number against a source of truth (database, API, or checksum) where it matters.