

# Regular Expression in Hindi - Python in Hindi

---

 [hindilearn.in/tut/python/regular-expression-in-hindi](https://hindilearn.in/tut/python/regular-expression-in-hindi)

## Python - Regular Expression

---

Python में regular expression ये string पर operation करता है |

String पर match या किसी string को find/search करने के लिए regular expression काफी मददगार साबित होता है |

Python में अगर regular expression का इस्तेमाल करना हो तो 're' इस module का इस्तेमाल करना पड़ता है |

किसी भी regular expression के हिस्से का इस्तेमाल करना हो तो program पर 're' इस module को import करना पड़ता है |

जैसे कि.

```
import re
```

## Raw String Notation for Regular Expression Patterns

---

Python में जब regular expression लिखना होता है तब Normal string के बजाय raw string का इस्तेमाल किया जाता है |

Raw String की शुरुआत 'r' इस prefix से की जाती है | अगर string पर 'r'(raw string) prefix दिया जाता है तो backslashes(\\) को किसी भी प्रकार से handle नहीं किया जा सकता है |

## Difference Between Normal String and Raw String

---

### Normal String

Normal String में '\\n'(escape sequence) ये एक ही character होता है |

### Raw String

Raw String में '\\' और 'n' ये दोनों अलग-अलग characters होते हैं |

### In Regular Expression

Regular Expression में raw string का इस्तेमाल pattern के रूप में काफी बार किया जाता है |

```
"\d+\w" #Normal String
```

```
r"\d+\w" #Raw String
```

## Matching a String

---

String से match करने के लिए re module के 'match()' function का इस्तेमाल किया जाता है |

## Syntax for match() Regular Expression Function

---

```
re.match(pattern, raw/string, flags)
```

### Parameters

**pattern** : यहाँ पर pattern दिया जाता है |

**raw/string** : यहाँ पर pattern से match करने के लिए string या raw string दी जाती है |

**flags** : Optional. यहाँ पर एक या एक से ज्यादा flags दिए जाते हैं | Bitwise OR(|) से एक से ज्यादा flags का इस्तेमाल किया जाता है |

## Example for match() Function in Python

---

Example में regular expression के match() function का इस्तेमाल किया गया है |

**pattern** : r"(\w+)\ (\d+)\ (\w+)"

ये raw string दी गयी है | पहले group में characters(\w+) को match किया गया है बाद में एक space(\ ) दिया गया है उसके बाद digits(\d+) को match किया गया है उसके बाद फिर एक space(\ ) और आखिर में फिर characters को match किया गया है |

**str** : Hello 123 Hello

इस normal string पर operation किया गया है |

Source Code :

```
import re

str = "Hello 123 Hello"
a = re.match(r"(\w+)\ (\d+)\ (\w+)", str)

print("group() :",a.group()) #return entire matches ((\w+)\ (\d+)\ (\w+))
print("group(0) :",a.group(0)) #return entire matches ((\w+)\ (\d+)\ (\w+))
print("group(1) :",a.group(1)) #return first subgroup match (\w+)
print("group(2) :",a.group(2)) #return second subgroup match (\d+)
print("group(3) :",a.group(3)) #return third subgroup match (\w+)
```

Output :

```
group() : Hello 123 Hello
group(0) : Hello 123 Hello
group(1) : Hello
group(2) : 123
group(3) : Hello
```

Python में Regular Expression में String पर operation करने के लिए कुछ हिस्से बनाये गए हैं ।

1. Character Classes
2. Quantifiers
3. Metacharacter
4. Modifiers/Flags
5. Regular Expression Functions
6. Regular Expression Object Methods
7. Regular Expression Match Object Methods

## 1. Character Classes for Regular Expression in Python

| Expressions | Description   |
|-------------|---|
| [.]         | कोई भी character या digits ढूँढने के लिए दिया जाता है ।               |
| [^.]        | कोई भी character या digits दिया जाता है , उसे ढूँढा नहीं जाता ।       |
| [0-9]       | 0 से 9 तक के digits को ढूँढा जाता है ।                                |
| [^0-9]      | 0 से 9 तक के digits को ढूँढा नहीं जाता है ।                           |
| [A-Z]       | Uppercase A से लेकर uppercase Z तक characters को ढूँढा जाता है ।      |
| [^A-Z]      | Uppercase A से लेकर uppercase Z तक characters को ढूँढा नहीं जाता है । |
| [a-z]       | lowercase a से लेकर lowercase z तक characters को ढूँढा जाता है ।      |
| [^a-z]      | lowercase a से लेकर lowercase z तक characters को ढूँढा नहीं जाता है । |
| [m]ail      | यहाँ पर mail और fail को string पर ढूँढा जाता है ।                     |
| Grac[ey]    | यहाँ पर Grace और Gracy को string पर ढूँढा जाता है ।                   |

## 2. Quantifiers for Regular Expression in Python

| Expressions | Description   |
|-------------|---|
| n+          | String में कम से कम एक या उससे ज्यादा 'n' occurrences को match किया जाता है । |

|                |  |
|----------------|--|
| <u>n*</u>      | String में zero या उससे ज्यादा 'n' occurrences से match किया जाता है   |
| <u>n?</u>      | String में zero या एक occurrence 'n' से match किया जाता है   |
| <u>n{X}</u>    | String में से n को sequences of X number तक match किया जाता है   |
| <u>n{X, Y}</u> | String में से n को sequences of X number से Y number तक match किया जाता है   |
| <u>n{X,}</u>   | String में से n को sequences of कम से कम X number तक match किया जाता है  |
| <u>n\$</u>     | String में से n को end पर match किया जाता है   अगर multiline(re.M) mode होता है तो हर newline के end पर match किया जाता है     |
| <u>^n</u>      | String में से n को start पर match किया जाता है   अगर multiline(re.M) mode होता है तो हर newline के start पर match किया जाता है |
| <u>p q</u>     | String में से p या q को match किया जाता है   |
| <u>python+</u> | String में से python या python में 'n' एक या एक से ज्यादा match किया जाता है   |
| <u>python*</u> | String में से python या python में 'n' शून्य या उससे ज्यादा match किया जाता है   |
| <u>python?</u> | String में से python या python में 'n' शून्य या एक match किया जाता है  |

### 3. Metacharacters for Regular Expression in Python

| Expressions   | Description  |
|---------------|--|
| <u>dot(.)</u> | single dot से single character ढूँढा जाता है                       |
| <u>b</u>      | String में से word के शुरुआत या end के matches ढूँढे जाते हैं      |
| <u>B</u>      | String में से word के शुरुआत या end के matches ढूँढे नहीं जाते हैं |
| <u>d</u>      | String में से 0 से 9 numbers तक ढूँढा जाता है                      |
| <u>D</u>      | String में से non-numbers character ढूँढा जाता है                  |
| <u>f</u>      | String में से form-feed character को ढूँढा जाता है                 |
| <u>n</u>      | String में से newline character को ढूँढा जाता है                   |
| <u>r</u>      | String में से carriage return character को ढूँढा जाता है           |
| <u>s</u>      | String में से whitespace character को ढूँढा जाता है                |
| <u>S</u>      | String में से non-whitespace character को ढूँढा जाता है            |
| <u>t</u>      | String में से tab character को ढूँढा जाता है                       |
| <u>v</u>      | String में से vertical tab character को ढूँढा जाता है              |

|          |   |
|----------|---|
| <u>w</u> | String में से word character को ढूँढा जाता है     |
| <u>W</u> | String में से non-word character को ढूँढा जाता है |
| <u>0</u> | String में से null character को ढूँढा जाता है     |

## 4. Modifiers/Flags for Regular Expression in Python

Function या Method पर एक से ज्यादा Modifiers/Flags इस्तेमाल करने हो तो बीच में OR(|) Operator इस्तेमाल किया जाता है |

| Expressions                  | Description   |
|------------------------------|---|
| <u>re.S or re.DOTALL</u>     | '.' character हर character को newline के साथ match किया जाता है   |
| <u>re.I or re.IGNORECASE</u> | Regular Expression को case-insensitive किया जाता है               |
| <u>re.L or re.LOCALE</u>     | \w, \W, \b, \B ये current locale का पालन करता है                  |
| <u>re.M or re.MULTILINE</u>  | String की हर line की beginning(^) और end(\$) match किया जाता है   |
| <u>re.U or re.UNICODE</u>    | \w, \W, \b, \B, \d, \D ये unicode character rules का पालन करता है |
| <u>re.X or re.VERBOSE</u>    | pattern पर # के साथ comment को add करने की अनुमति देता है         |

## 5. Regular Expression Functions

| Functions         | Description  |
|-------------------|--|
| <u>compile()</u>  | regular expression object पर regular expression pattern को compile किया जाता है  |
| <u>findall()</u>  | सभी non-overlapping matches list में return करता है  |
| <u>finditer()</u> | सभी non-overlapping matches; callable object में return करता है  |
| <u>match()</u>    | pattern के हिसाब से string के beginning पर ही match करके match object return करता है   |
| <u>search()</u>   | pattern के हिसाब से string के पहले occurrence को match करके match object return करता है  |
| <u>split()</u>    | pattern द्वारा string को split करके अलग-अलग substrings को list में return किया जाता है   |
| <u>sub()</u>      | pattern के हिसाब से string के substring(s) को replace करके नया string return करता है   |
| <u>subn()</u>     | pattern के हिसाब से string के substring(s) को replace करके replaced string और replaced हुए pattern occurrence(s) को tuple में return करता है |

## 6. Regular Expression Object Methods

Object Method के लिए compiled pattern का इस्तेमाल किया जाता है इसके लिए pattern को compile() function में लिखा जाता है |

| Object Methods    | Description  |
|-------------------|--|
| <u>findall()</u>  | दिए हुए pattern के हिसाब से और starting से ending position तक match करके सभी non-overlapping matches को list में return करता है              |
| <u>finditer()</u> | दिए हुए pattern के हिसाब से और starting से ending position तक match करके सभी non-overlapping matches को callable object में return करता है   |
| <u>match()</u>    | दिए हुए pattern के हिसाब से और starting से ending position तक के beginning के match को match object में return करता है                       |
| <u>search()</u>   | दिए हुए pattern के हिसाब से और starting से ending position तक के पहले occurrence को match करके match object में return करता है               |
| <u>split()</u>    | दिए हुए pattern के हिसाब से string को split करके list में return करता है   |
| <u>sub()</u>      | pattern के हिसाब से string के substring(s) को replace करके नया string return करता है   |
| <u>subn()</u>     | pattern के हिसाब से string के substring(s) को replace करके replaced string और replaced हुए pattern occurrence(s) को tuple में return करता है |

## 7. Regular Expression Match Object Methods

|                    |   |
|--------------------|---|
|                    |   |
| <u>group()</u>     | match के एक या एक से ज्यादा subgroups को return करता है                       |
| <u>groups()</u>    | match के एक या एक से ज्यादा subgroups को tuple में return करता है             |
| <u>groupdict()</u> | key और उसके matches के साथ dictionary को return करता है                       |
| <u>start()</u>     | दिए हुए group String पर जहा से match होता है वहा से उसकी index return करता है |
| <u>end()</u>       | दिए हुए group String पर जहा तक match होता है वहा से उसकी index return करता है |