**Regular Expression / Pattern Matching –**

**In General Programming, We have to search or match data from given values. Which is beneficial in validate data. Or to check data given in specific format or not.**

**Search and Replace…**

**Here, python provides re module for regular expression.**

**(no need to use pip (internally available))**

**Function is re module=>**

**match ()=>Which checks whether given pattern is from beginning of line.**

**search()=> which is used to search a given pattern and it occurance any where in string.**

**findall()=>Which matches a pattern with whole string. And create a new list of matching patterns(if repeated)**

**compile()=>It is similar to search function.**

**sub()=> It removes a data which not matches with pattern and return a string.(Subtract a pattern)**

\d

Matches any decimal digit; this is equivalent to the class [0-9].

\D

Matches any non-digit character; this is equivalent to the class [^0-9].

\s

Matches any whitespace character; this is equivalent to the class [ \t\n\r\f\v].

\S

Matches any non-whitespace character; this is equivalent to the class [^ \t\n\r\f\v].

\w

Matches any alphanumeric character; this is equivalent to the class [a-zA-Z0-9\_].

\W

Matches any non-alphanumeric character; this is equivalent to the class [^a-zA-Z0-9\_].

|  |  |
| --- | --- |
| **Sr.No.** | **Pattern & Description** |
| 1 | **^**  Matches beginning of line. |
| 2 | **$**  Matches end of line. |
| 3 | **.**  Matches any single character except newline. Using m option allows it to match newline as well. |
| 4 | **[]**  Matches any single character in brackets. |
| 5 | **[^...]**  Matches any single character not in brackets |
| 6 | **re\***  Matches 0 or more occurrences of preceding expression. |
| 7 | **re+**  Matches 1 or more occurrence of preceding expression. |
| 8 | **re?**  Matches 0 or 1 occurrence of preceding expression. |
| 9 | **re{10}**  Matches exactly n number of occurrences of preceding expression. |
| 10 | **re{ n,}**  Matches n or more occurrences of preceding expression. |
| 11 | **re{ n, m}**  Matches at least n and at most m occurrences of preceding expression. |
| 12 | **a| b**  Matches either a or b. |
| 13 | **(re)**  Groups regular expressions and remembers matched text. |
| 14 | **(?imx)**  Temporarily toggles on i, m, or x options within a regular expression. If in parentheses, only that area is affected. |
| 15 | **(?-imx)**  Temporarily toggles off i, m, or x options within a regular expression. If in parentheses, only that area is affected. |
| 16 | **(?: re)**  Groups regular expressions without remembering matched text. |
| 17 | **(?imx: re)**  Temporarily toggles on i, m, or x options within parentheses. |
| 18 | **(?-imx: re)**  Temporarily toggles off i, m, or x options within parentheses. |
| 19 | **(?#...)**  Comment. |
| 20 | **(?= re)**  Specifies position using a pattern. Doesn't have a range. |
| 21 | **(?! re)**  Specifies position using pattern negation. Doesn't have a range. |
| 22 | **(?> re)**  Matches independent pattern without backtracking. |
|  |  |

## Validate a Email-Id

Case sensitive to verify if a email id is a valid or not.

## Check Password Complexity

This regular expression will tests if the input consists of 6 or more letters, digits, underscores, and hyphens.  
The input must contain at least one uppercase letter, one lowercase letter and one digit.

## Match Hexadecimal Color Values

Another interesting tool for web developers! It allows you to match/validate a hexadecimal color value.

## Highlight a Word From a Text

This very useful regular expression will find a specific word in a string and highlight it with . Extremely useful for search results. Remember that it’s case sensitive.

## Validate a Domain Name

Case sensitive regex to verify if a string is a valid domain name. This is very useful when validating [web forms](https://catswhocode.com/bootstrap-forms/).