**RegEx**

RegEx string me kuch find karne ka advance tarika hai.

RegEx ko ham do forward slashes ke bich me likhte hain.

E.g.: /cat/g

Ending forward slash ke baad ham alag alag flags ka use karsakte hain. Unme se ek flag hai “g” jiske help se ham global search kara sakte hain. Fir ek “i” flag bhi hai jiske help se ham case insensitive search kara sakte hain.

**Examples**

**E.g.: /e+/g**

Match one or more occurrences of e.

**E.g.: /ea?/g**

Match e and match 0 or 1 occurrences of a.

**E.g.: /ea\*/g**

Match e and match 0 or more occurrences of a.

**E.g.: /e./g**

Match e and match any one character after it except for newline character.

**E.g.: /\./g**

Match any period character.

**E.g.: /.\./g**

Match any character before period and period character.

**E.g.: /\w/g**

Match any alphabetic and numeric character.

**E.g.: /\s/g**

Match any whitespace character.

**E.g.: /\S/g**

Match any non-whitespace character.

**E.g.: /\W/g**

Match any non-word character.

**E.g.: /\w{4,5}/g**

Match any word that is 4 to 5 characters long.

**E.g.: /\w{4,}/g**

Match any word that is 4 or more characters long.

**E.g.: /[a-zA-z0-9]at/g**

Match any word that starts with a-z, A-Z or 0-9 and ends with at.

**E.g.: /(t|T)he/g**

() se ham expressions ko group karsakte hai. Match any word that has t or T and ends with he. The pipe symbol is for or.

**E.g.: /(t|e|r){2, 3}\./g**

Match any portion of a word that has t or e or r, and is 2 to 3 characters long and ends with a period.

**E.g.: /^T/gm**

Match the beginning of each like that starts with T. Agar m(multi-line) flag nahi lagayenge to sirf pure text ka starting match hooga.

**E.g.: /\.$/g**

Match the ending part of the string that ends with a period.

**Look Aheads and Look Behinds**

**Positive Look Behind**

**/(?<=[tT]he)./g**

Match any character after the or The.

**Negative Look Behind**

**/(?<![tT]he)./g**

Match any thing except the character after the or The.

**Positive Look Ahead**

**/.(?=at)/g**

Match any character before at.

**Negative Look Ahead**

**/.(?=at)/g**

Match anything that is not a character before at.

**Real World Use Case**

**Input**

1234567890

123-456-7890

123 456 7890

(123) 456 7890

+1 123 456 7890

**E.g.: /(?:(\+91)[ -])?\(?(?<areacode>\d{3})\)?[ -]?(\d{3})[ -]?(\d{4})/gm**

?<name> se ham capturing groups ko naam desakte hain.

? se ham batate hai ke ye optional hai.

\+ karna pada kyu ke + ek special character hai similarly \( aur \) bhi karna pada dono round brackets ko escape karne ke liye.

(\d{3}) yane 3 digits ko ek group me capture karo.

[ -]? yane – aur space optional hai.

?:(\+91) se ham batarahe hai ke +91\_ (\_ is for space) ke group ko nahi capture karna hai.