

# Regular Expression

46

→ What are Regular Expressions?

→ Regular expressions are the patterns that are used to match character combinations in strings. Regular expressions are a powerful way of doing search and replace in strings.

It is a small language which is a part of many programming languages like JavaScript, Python, and Java. ~~But~~ Regular expression allows us to check a string of characters like a

password for patterns, to see if the set password matches with the pattern defined by that regular expression.

### SYNTAX

/ PATTERN / MODIFIERS ;

### EXAMPLE

```
LET STR = /CODE WITH HARRY/i;
```

→ Here /Code with Harry/i is a regular expression. "Code with Harry" is a pattern and "i" is a modifier that modifies the search to be case-insensitive. If we write /CODE WITH HARRY/g, here "g" performs a global match that will find all matches rather than stopping after the first match.

### → Regular Expressions Methods:

→ Regular expressions are used with the RegExp methods like test() and exec() and with the string methods match(), replace(), search(), and split(). These methods are explained in detail below with examples.

### → EXEC():

→ This method will executes a search for

Date: 

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

  
a match in a string. It returns an array of information or null on a mismatch. Ex:

→ LET OBJ = |h|.exec("CODE WITH HARRY");

→ TEST ( ) :

→ This method tests for match in a string. It returns true or false. Ex:

→ LET STR = |CODE|;  
STR.TEST("CODE WITH HARRY!");

→ MATCH ( ) :

→ This method will return an array containing all of the matches, including capturing groups, or null if no match is found.

Ex:

→ LET STR = "JAVASCRIPT TUTORIAL FOR BEGINNER";  
LET RESULT = STR.MATCH(/i|a|/);

→ SEARCH ( ) :

→ This method will test for a match in a string. It returns the index of the match, or -1 if the search fails. Ex:

→ LET STR = 'CODE WITH HARRY';

LET REG = /OD/;

// Search if the pattern is in string variable.

LET RESULT = STR.SEARCH(REG);

```
CONSOLE.LOG (RESULT);
```

→ REPLACE ( ) :

→ This method will execute a search for a match in a string, and replaces the matched substring with a replacement substring. Ex:

```
→ LET STR = "CODE WITH ABC!";  
LET RESULT = STR.REPLACE ("ABC", "HARRY");
```

→ SPLIT ( ) :

→ This method uses a regular expression or a fixed string to break a string into an array of substrings. Ex:

```
→ // splitting strings into array elements  
CONST TEST = /[ \s,]+/;  
LET RES = 'HELLO WORLD!'.SPLIT (RE);  
CONSOLE.LOG (RESULT); // [ "3", "AM", "LEARNING"  
"JAVASCRIPT", "REGEX" ]
```