modifiers

```
- g(global)
- i(case insensitive)
- m(multiline match)
```

function

```
1.exec() - return the result along with its info if result
2.test() - return true if result is true
3.match() - return array of result or null
4.search() - return index of first match
5.replace() - replace the given word with where regexp match
```

object

```
.input
.index
.source
```

```
    let reg1 = /hi/g;
    let reg2 = /riya/i;

    console.log(reg1.source);
    // show the regular expression literal - hi

    let s = " my name hi is hi Riya good to see ya";
    let a = " riya is good girl riya is best happy girl"
    //console.log(a);
    //console.log(s);

    result1 = reg1.exec(s); // show the string
    //console.log(result1);

    let result2 = reg2.test(s);
    //console.log(result2);

    let result3 = s.match(reg1)
    //console.log(result3); // ['hi','hi']

    let result4 = s.search(reg2);
    //console.log(result4);

    let result5 = s.replace(reg1,"BISHT");
```

```
    console.log(result5);
```

```
let str1 = "Twttttttinkle 23 twinkle . 2 little 23star dog fish dish this is
dish "
//console.log(str1);
let rex1 = /dog|cat|fish/
let result1 = rex1.test(str1)
//console.log(result1);

let rex2 = /twinkle/ig //case insensitive and global
let result2 =  str1.match(rex2)
//console.log(result2);

/* extract match */

// wildcard period
//1. print words having one letter before is  "dis , fis , his"
let rex3 = /.is/g
let result3 = str1.match(rex3)
//console.log(result3);
//2.print words , with one letter after is
let rex4= /is./g
let result4 = str1.match(rex4)
//console.log(result4);

// single character with multiple Possibilities
let rex5 = /w[aeiou]n/g
/*return all 3 letter starting with w and ending with n
,2rd letter any one among a,e,i,o,u  */
let result5 = str1.match(rex5)
//console.log(result5);

// match letters of alphabet
// letter can be anything between [a-z]
let rex6 = /[a-z]/ig
let result6 = str1.match(rex6)
//console.log(result6);

//match number and letter of alphabet
let rex7 = /[a-z0-9]/ig;
let result7 = str1.match(rex7)
////console.log(result7);

//don;t match if no or vowels
let rex8 = /[^0-9aeiou]/g
let result8= str1.match(rex8)
////console.log(result8);

//match character that  occour one or more times [t,ttt,ttttt,ttt]
let rex9 = /t+/gi
```

```
let result9= str1.match(rex9)
//console.log(result9);

//match character that occur zero or more times
// [s-z]t* - any letter between "s-z" before t , t can occur 0 or more
times
let rex10 = /[s-z]t*/gi
let result10= str1.match(rex10)
//console.log(result10)

//find character with lazy matching

let str2 = "<h1>Spring_**is coming at 7:00</h1>"
console.log(str2);
let re1 = /<.*>/
let re2 = /<.*?>/
//console.log(str2.match(re1)); //"<h1>Spring is coming</h1>"
//console.log(str2.match(re2));//"<h1>" - lazy match

// beginning of the string "^"
let re3 = /^tw/i
//console.log(str1.match(re3));

//match ending string patterns "$"
let re4 = /dish $/
console.log(str1.match(re4));

// match all letters and numbers
/*
\w = [a-zA-Z0-9_]
\W = every thing which is not letter, digit, underscore
\d = match digit
\D = match all non number
\s = whitespace character
*/
let re5 = /\w/g
console.log(str2.match(re5));

let re6 = /\W/g
console.log(str2.match(re6));

let re7 = /\d/g
console.log(str2.match(re7));

let re8 = /\D/g
console.log(str2.match(re8));
```

```
/*
\w = [a-zA-Z0-9_]
\W = every thing which is not letter, digit, underscore
\d = match digit
```

```
\D = match all non number
\s = whitespace character
\S = non whitesapce
*/



r1= /\w+@gmail.com/
str = "riya152@gmail.com"
console.log(str.match(r1));

//{min match , max match}- min 2 match  and after , max match
/*
1) If there are number, they must be at the end.
2) Letters can be lowercase and uppercase
3) At least two characters long. Two-letter names can't have numbers
 */
let r2 = /^[a-zA-Z]{2,}\d*$/i
let username = "a1523"
console.log(username.match(r2));

let str2 = "ohhhhh no"
let r3 = /oh{2,} no/
console.log(str2.match(r3));

// specify only the lower number of matches
str3 = "hxxxxzzzz"
let r4 = /x{2,}/
//['xxxx']

// match string "favorite" and "favourite"
str4 = "favorite"
let r5= /favou?rite/ // may or may not have a "u"
console.log(str4.match(r5));


// look ahead
/*
1- positive look ahead (?=<expression>)
2- negative look ahead (?!<expression>)
 */
let quit = "qu quick qwak"
re6 = /q(?=u)/g
re7 = /q(?!u)/g
console.log(quit.match(re6)); // return q not u check at - "qu , quick"
console.log(quit.match(re7)); // return q not u  "qwak"

/*
passwords greater than 5 character long
have 2 consecutive digit

*/
re8 = /(?=\w{5})(?=\D*\d{2})/
```

```
/*
Reuse pattern using capture group
1 - any string of letter with space after string
\1 -
*/
s1 = "riya riya"
r6 = /(\w+)\s\1/
console.log(s1.match(r6));    // ["riya riya " , "riya"]
/*
-full expression
- what ever in the capture group
- "42 42 42 42 78732 83883"
- only 3 similar number separated by space
 */

s2 = "42 42 42"
console.log(s2);
r7 = /^(\d+)\s\1\s\1$/g
console.log(s2.match(r7));


/*
- replace code comp with camp code
 */
console.log("code camp");
"code camp".replace(/(\w+)\s(\w+)/,'$1 $2')

/*
    remove whitespace at the beginning and end
    " Hello, World!  ";
*/
let hello = "  Hello, world! "
console.log(hello);
let regx = /^\s+|\s+$/g
console.log(hello.replace(regx,""));
console.log(hello);
```