

JavaScript

Regular Expression

What Is a Regular Expression?

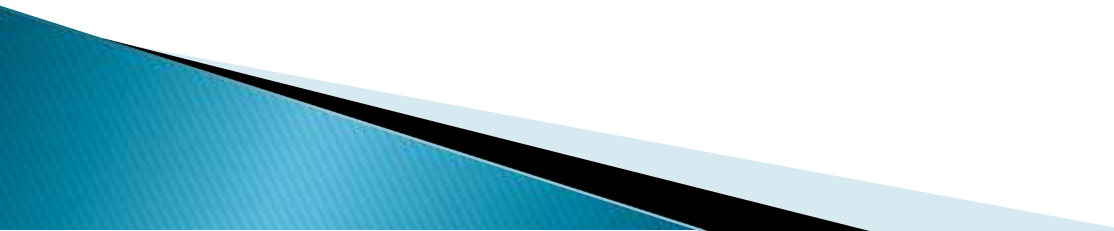
- ▶ A regular expression is a sequence of characters that forms a **search pattern**.
- ▶ A regular expression can be a single character, or a more complicated pattern.
- ▶ Regular expressions can be used to perform all types of **text search** and **text replace** operations.
- ▶ **Syntax:**

/pattern/modifiers;



Example

`/w3schools/i;`

- ▶ `/w3schools/i` is a regular expression.
 - ▶ `w3schools` is a pattern (to be used in a search).
 - ▶ `i` is a modifier (modifies the search to be case-insensitive).
- 

Regular Expression Modifiers

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

Regular Expression Modifiers

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Regular Expressions</h2>
```

```
<p>Do a case-insensitive search for "w3schools" in a string:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let text = "Visit W3Schools";
```

```
let pattern = /w3schools/i;
```

```
let result = text.match(pattern);
```

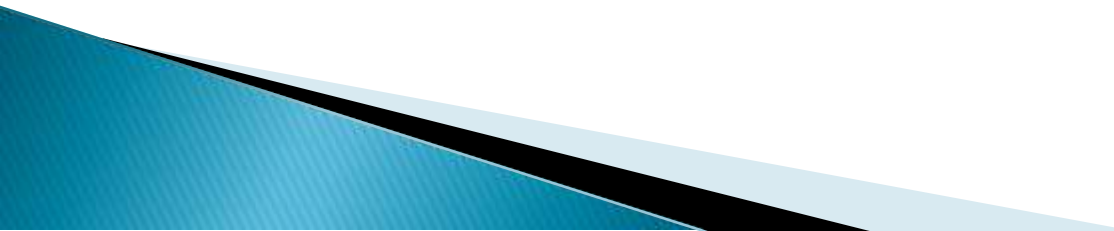
```
document.getElementById("demo").innerHTML = result;
```

```
</script>
```

```
</body>
```

```
</html>
```

Regular Expression Patterns

- ▶ **Brackets** are used to find a range of characters
 - ▶ **Metacharacters** are characters with a special meaning
 - ▶ **Quantifiers** define quantities
- 

Regular Expression Patterns

Expression	Description
[abc]	Find any of the characters between the brackets
[0-9]	Find any of the digits between the brackets
(x y)	Find any of the alternatives separated with

Regular Expression Patterns

```
const text = "apple orange banana grape";  
const pattern = /[aeiou]/g; // Match any vowel  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'a', 'e', 'o', 'a', 'a', 'e' ]
```

```
const text = "A B c d E F";  
const pattern = /[a-z]/g; // Match any lowercase letter  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'c', 'd' ]
```

```
const text = "apple123 orange456 banana789";  
const pattern = /[a-z0-9]/g; // Match any lowercase letter or digit  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'a', 'p', 'p', 'l', 'e', '1', '2', '3', 'o', 'r', 'a',  
'n', 'g', 'e', '4', '5', '6', 'b', 'a', 'n', 'a', '7', '8', '9' ]
```


Regular Expression Patterns

Metacharacter	Description
<code>\d</code>	Find a digit
<code>\s</code>	Find a whitespace character
<code>\b</code>	Find a match at the beginning of a word like this: <code>\bWORD</code> , or at the end of a word like this: <code>WORD\b</code>
<code>\uxxxx</code>	Find the Unicode character specified by the hexadecimal number <code>xxxx</code>

Regular Expression Patterns

Quantifier	Description
n^+	Matches any string that contains at least one n
n^*	Matches any string that contains zero or more occurrences of n
$n?$	Matches any string that contains zero or one occurrences of n

Regular Expression Patterns

```
const text = "cat bat mat";  
const pattern = /b.t/g; // Match "bat" and "bet"  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'bat' ]
```

```
const text = "aaaab aab abb";  
const pattern = /a*b/g; // Match "aaaab", "aab", and "abb"  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'aaaab', 'aab', 'abb' ]
```

```
const text = "abc abbc aabbc";  
const pattern = /ab+c/g; // Match "abc", "abbc", and "aabbc"  
const matches = text.match(pattern);  
console.log(matches); // Output: [ 'abc', 'abbc', 'abbc' ]
```

Regular Expression Patterns

```
const text = "123 1234 12345";  
const pattern = /\d{4}/g; // Match four-digit numbers  
const matches = text.match(pattern);  
console.log(matches); // Output: [ '1234', '1234']
```

Regular Expression Patterns

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Regular Expressions</h2>
```

```
<p>Do a case-insensitive search for "w3schools" in a string:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let text = "Visit W3Schools";
```

```
let pattern = /w3schools/i;
```

```
let result = text.match(pattern);
```

```
document.getElementById("demo").innerHTML = result;
```

```
</script>
```

```
</body>
```

```
</html>
```

RegExp Object

- ▶ In JavaScript, the RegExp object is a regular expression object with predefined properties and methods.
- ▶ Using test()
 - The test() method is a RegExp expression method.
 - It searches a string for a pattern, and returns true or false, depending on the result.

```
const pattern = /e/;  
pattern.test("Welcome!");  
Output :true
```

```
/e/.test("Welcome!");
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Regular Expressions</h2>
```

```
<p>The exec() method tests for a match in a string:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let text = "Hello world!";
```

```
// look for "Hello"
```

```
let result1 = /Hello/.exec(text);
```

```
// look for "W3Schools"
```

```
let result2 = /world/.exec(text);
```

```
document.getElementById("demo").innerHTML =
```

```
result1 + "<br>" + result2;
```

```
</script>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript RegExp</h2>
```

```
<p>The exec() method tests for a match in a string:</p>
```

```
<p>Search a string for the character "e":</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let text = "The best things in life are free";
```

```
let result = /e/.exec(text);
```

```
document.getElementById("demo").innerHTML = result;
```

```
</script>
```

```
</body>
```

```
</html>
```