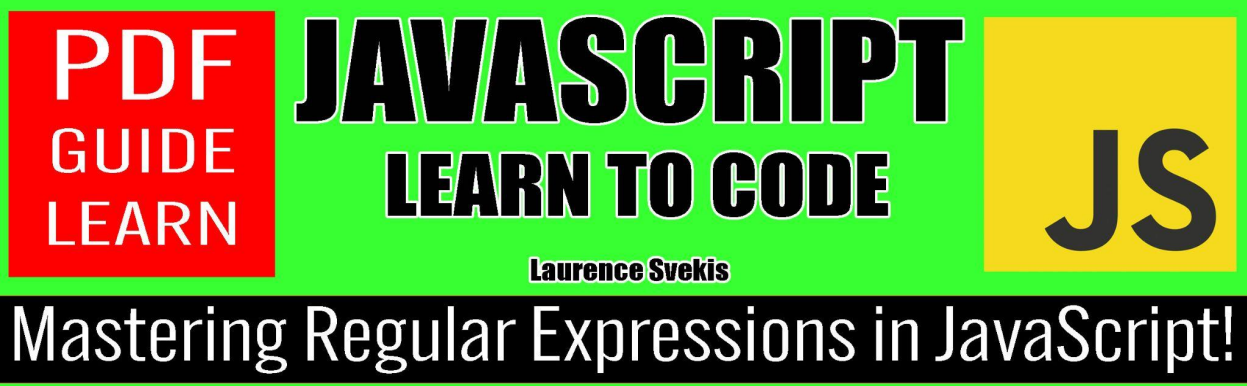


Mastering Regular Expressions in JavaScript!



Whether you're a beginner or a seasoned developer, understanding Regular Expressions (Regex) is a crucial skill in JavaScript. Let's test our knowledge with a quick quiz on Regex concepts. No code included, just a chance to refresh your understanding!

- Basics of Regular Expressions:** 6
 - Creating a Regular Expression: 6
 - Metacharacters: 6
- Regular Expression Methods:** 7
 - test(): 7
 - exec(): 7
- Character Classes:** 7
 - [] - Character Set: 7
 - [^] - Negated Character Set: 8
- Quantifiers:** 8
 - {n}: 8
 - {n,}: 8
 - {n,m}: 8

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Examples:	9
Matching Email Addresses:	9
Extracting Dates from Text:	9
Replacing Text:	9
Advanced Regular Expression Features:	10
Grouping ():	10
Capturing Groups ():	10
Non-capturing Groups (?):	10
Assertions:	10
Positive Lookahead (?=):	11
Negative Lookahead (?!):	11
Quantifiers and Lazy Matching:	11
Validating URLs:	11
Extracting HTML Tags:	11
Validating Passwords:	12
Parsing CSV:	12
Replacing Words with Callback:	12
Coding Exercise Regex	13
Exercise 1: Validate Email Address	13
Exercise 2: Extract Numbers from String	14
Exercise 3: Replace HTML Tags	15
Exercise 4: Validate Password Strength	16
Exercise 5: Extract Domain from URL	17
Exercise 6: Validate Date Format	18
Exercise 7: Extract Hashtags from Text	19
Exercise 8: Validate Credit Card Number	20
Exercise 9: Replace URLs with Links	22
Exercise 10: Validate Phone Number	23
Regex Quiz Questions	24
Question: What is a regular expression in JavaScript?	29

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

Question: Which object in JavaScript is used for working with regular expressions?	29
Question: What does the test() method of a RegExp object do?	29
Question: Which character is used as the delimiter for a regular expression literal?	30
Question: What does the exec() method of a RegExp object return?	30
Question: Which quantifier matches zero or more occurrences of the preceding element?	30
Question: What is the purpose of the ^ anchor in a regular expression?	31
Question: Which character class matches any digit character?	31
Question: In the regular expression /ab+c/, what does the + symbol represent?	31
Question: What is the purpose of the non-capturing group (?:) in a regular expression?	32
Question: What does the ? quantifier indicate in a regular expression?	32
Question: Which assertion checks if a certain element is followed by another element without including the latter in the match?	32
Question: What is the purpose of the g flag in a regular expression?	33
Question: Which character is used for matching any character except a newline?	33
Question: In the regular expression /[aeiou]/, what does the character class [aeiou] represent?	33
Question: Which quantifier matches exactly three occurrences of the preceding element?	34
Question: What does the replace() method do when used with a regular expression?	34
Question: Which method is used to find all occurrences of a pattern in a string, including capturing groups?	34
Question: In the regular expression /^[\w.-]+@[a-z]+\.[a-z]+\$/i, what does the i flag represent?	35
Question: What is the purpose of the \b anchor in a regular expression?	35
Question: Which method returns an array of all occurrences of a pattern in a	

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

string without capturing groups?	35
Question: In the regular expression <code>/[^\0-9]/</code> , what does the character class <code>[^\0-9]</code> represent?	36
Question: Which quantifier matches between two and four occurrences of the preceding element?	36
Question: What does the <code>\d</code> character class represent in a regular expression?	36
Question: In the regular expression <code>/^\d{3}-\d{2}-\d{4}\$/</code> , what does the pattern represent?	37
Question: What does the <code>matchAll()</code> method return?	37
Question: What does the <code>\S</code> character class represent in a regular expression?	37
Question: In the regular expression <code>/(\w+)\s(\w+)/</code> , what do the capturing groups <code>(\w+)</code> represent?	38
Question: Which character is used to escape a metacharacter in a regular expression?	38
Question: What is the purpose of the <code>search()</code> method in JavaScript when used with a regular expression?	38
Question: Which method splits a string into an array of substrings based on a regular expression?	39
Question: What is the purpose of the <code>\w</code> character class in a regular expression?	39
Question: In the regular expression <code>/(\d{2})\V(\d{2})\V(\d{4})/</code> , what do the capturing groups <code>(\d{2})</code> , <code>(\d{2})</code> , and <code>(\d{4})</code> represent?	39
Question: What is the purpose of the <code>\W</code> character class in a regular expression?	40
Question: In the regular expression <code>/(\b\w+\b)\s\1/</code> , what does <code>\1</code> represent?	40
Question: What does the <code>\b</code> anchor do in a regular expression?	40
Question: Which quantifier matches one or more occurrences of the preceding element?	41
Question: In the regular expression <code>/^[A-Z][a-z]*\$/</code> , what does the pattern represent?	41

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

Question: What is the purpose of the <code>\n</code> escape sequence in a regular expression?	41
Question: In the regular expression <code>/(\d+)\s(?:years? yrs?)</code> , what does <code>(?:years? yrs?)</code> represent?	42
Question: What is the purpose of the <code>\s</code> character class in a regular expression?	42
Question: In the regular expression <code>/(\d{3})-(\d{2})-(\d{4})</code> , what does the capturing group <code>(\d{3})</code> represent?	42
Question: Which method is used to test if a string contains a pattern in a regular expression?	43
Question: What does the <code>\S+</code> pattern represent in a regular expression?	43
Question: In the regular expression <code>/^[a-zA-Z]\w*\$</code> , what does the pattern represent?	43
Question: What is the purpose of the <code>\</code> character in a regular expression?	44
Question: In the regular expression <code>/[^aeiou]</code> , what does the character class <code>[^aeiou]</code> represent?	44
Question: What does the <code>flags</code> property of a <code>RegExp</code> object return?	44
Question: What is the purpose of the <code>\$</code> anchor in a regular expression?	45
Question: In the regular expression <code>/([a-z]+\s\d+)</code> , what do the capturing groups <code>([a-z]+)</code> and <code>(\d+)</code> represent?	45
Quiz Answers	45

Regular expressions, often referred to as `regex` or `regexp`, are powerful patterns used for matching character combinations in strings. In JavaScript, regular expressions are represented by the `RegExp` object. They provide a concise and flexible way to search, match, and manipulate text.

Basics of Regular Expressions:

Creating a Regular Expression:

Regular expressions can be created using a literal notation or the `RegExp` constructor.

```
// Literal notation
```

```
let regexLiteral = /pattern/;
```

```
// Using RegExp constructor
```

```
let regexConstructor = new RegExp("pattern");
```

Metacharacters:

Regular expressions consist of ordinary characters (like letters and numbers) and metacharacters that have special meaning.

Some common metacharacters include `.` (any character), `^` (start of a line), `$` (end of a line), `*` (zero or more occurrences), `+` (one or more occurrences), and `?` (zero or one occurrence).

```
let pattern = /ab*c/;
```

Regular Expression Methods:

test():

The test() method checks if a pattern exists in a string and returns true or false.

```
let regex = /world/;  
let result = regex.test("Hello, world!"); // true
```

exec():

The exec() method searches for a match in a string. It returns an array containing the matched text and additional information or null if no match is found.

```
let regex = /world/;  
let result = regex.exec("Hello, world!"); // ["world", index: 7,  
input: "Hello, world!", groups: undefined]
```

Character Classes:

[] - Character Set:

A character set allows you to match any one of a set of characters.

```
let regex = /[aeiou]/;
```

[^] - Negated Character Set:

Match any character that is not in the specified set.

```
let regex = /^[^0-9]/;
```

Quantifiers:

{n}:

Matches exactly n occurrences of the preceding character or group.

```
let regex = /a{3}/;
```

{n,}:

Matches n or more occurrences of the preceding character or group.

```
let regex = /\d{2,}/; // Match two or more digits
```

{n,m}:

Matches between n and m occurrences of the preceding character or group.

```
let regex = /\w{2,4}/; // Match between 2 and 4 word characters
```


Examples:

Matching Email Addresses:

```
let emailRegex =  
/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;  
let isValidEmail = emailRegex.test("user@example.com");
```

Extracting Dates from Text:

```
let dateRegex = /\b(\d{1,2})\b(\d{1,2})\b(\d{4})\b/;  
let dateString = "Today's date is 12/25/2023.";   
let dateMatch = dateString.match(dateRegex);
```

Replacing Text:

```
let text = "Hello, my name is John.";   
let nameRegex = /John/;   
let updatedText = text.replace(nameRegex, "Doe");
```

Regular expressions can be intricate, and mastering them takes practice. Use online tools to test and visualize your patterns, and gradually explore more advanced features to handle diverse scenarios.

Advanced Regular Expression Features:

Grouping ():

Parentheses are used for grouping parts of a pattern together.

```
let regex = /(ab)+/;
```

Capturing Groups ():

Capturing groups allow you to extract matched parts of a pattern.

```
let dateRegex = /(\d{1,2})\V(\d{1,2})\V(\d{4})/;
```

```
let dateString = "Today's date is 12/25/2023.";
```

```
let dateMatch = dateString.match(dateRegex);
```

```
// dateMatch: ["12/25/2023", "12", "25", "2023"]
```

Non-capturing Groups (?:):

Non-capturing groups group expressions without capturing the matched result.

```
let regex = /(?:ab)+/;
```

Assertions:

Assertions are conditions that must be true for a match to occur.

Positive Lookahead (?=):

```
let regex = /Java(?=Script)/;
```

Negative Lookahead (?!):

```
let regex = /Java(?!Script)/;
```

Quantifiers and Lazy Matching:

Quantifiers (*, +, ?, {n}, {n,}, {n,m}) can be modified with ? to perform lazy (non-greedy) matching.

```
let greedyRegex = /".+"/;
```

```
let lazyRegex = /".+?"/;
```

Examples:

Validating URLs:

Check if a string is a valid URL.

```
let urlRegex = /^(https?|ftp):\/\/[^\s/$.?#].[^\s]*$/;
```

```
let isValidURL = urlRegex.test("https://www.example.com");
```

Extracting HTML Tags:

Extract all HTML tags from a string.

```
let htmlRegex = /<[^>]*>/g;
```

```
let htmlString = '<div>Hello</div><p>World</p>';
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
let tags = htmlString.match(htmlRegex);
```

Validating Passwords:

Ensure a password meets specific criteria.

```
let passwordRegex =  
/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d@$!%*?&]{8,}$/;  
let isValidPassword = passwordRegex.test("Passw0rd");
```

Parsing CSV:

Extract values from a comma-separated values string.

```
let csvString = "John,Doe,30\nAlice,Smith,25";  
let csvRegex = /([\^,\n]+)(,|$)/g;  
let csvMatches = [...csvString.matchAll(csvRegex)];  
// csvMatches: [["John", ",",], ["Doe", ",",], ["30", "\n"], ["Alice",  
",",], ["Smith", ",",], ["25", ""]]
```

Replacing Words with Callback:

Use a callback function with `replace` to modify matched substrings.

```
let text = "Hello world, this is awesome."  
let regex = /\b(\w+)\b/g;
```

```
let updatedText = text.replace(regex, (match, word) =>
word.toUpperCase());
// updatedText: "HELLO WORLD, THIS IS AWESOME."
```

Understanding regular expressions is a powerful skill that can greatly enhance your ability to manipulate and validate strings in JavaScript. Practice with different patterns and scenarios to solidify your understanding.

Coding Exercise Regex

Exercise 1: Validate Email Address

Description: Create a function that validates whether a given string is a valid email address.

Steps:

1. Define a regular expression for validating email addresses.
2. Write a function that uses the regular expression to check if the input is a valid email address.

Code Example:

```
function isValidEmail(email) {
// Regular expression for email validation
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
return emailRegex.test(email);
}
```

```
// Test the function
console.log(isValidEmail("user@example.com")); // true
console.log(isValidEmail("invalid-email")); // false
```

Solution:

```
function isValidEmail(email) {
  let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  return emailRegex.test(email);
}
```

Exercise 2: Extract Numbers from String

Description: Write a function that extracts all the numbers from a given string.

Steps:

1. Define a regular expression to match numbers.
2. Write a function that uses match to extract all numbers from the input string.

Code Example:

```
function extractNumbers(inputString) {
  // Regular expression to match numbers
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
let numberRegex = /\d+/g;
return inputString.match(numberRegex);
}
```

// Test the function

```
console.log(extractNumbers("The price is $20 and the quantity is 5.")); // ['20', '5']
```

Solution:

```
function extractNumbers(inputString) {
  let numberRegex = /\d+/g;
  return inputString.match(numberRegex);
}
```

Exercise 3: Replace HTML Tags

Description: Create a function that replaces all HTML tags in a given string with an empty string.

Steps:

1. Define a regular expression to match HTML tags.
2. Write a function that uses replace to remove all HTML tags.

Code Example:

```
function removeHtmlTags(inputString) {
  // Regular expression to match HTML tags
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
let htmlTagRegex = /<[^>]*>/g;
return inputString.replace(htmlTagRegex, "");
}
```

// Test the function

```
console.log(removeHtmlTags("<p>Hello</p><div>World</div>"));
// 'HelloWorld'
```

Solution:

```
function removeHtmlTags(inputString) {
  let htmlTagRegex = /<[^>]*>/g;
  return inputString.replace(htmlTagRegex, "");
}
```

Exercise 4: Validate Password Strength

Description: Write a function that checks whether a given password meets certain criteria.

Steps:

1. Define a regular expression to enforce password criteria.
2. Write a function that uses the regular expression to validate the password.

Code Example:

```
function isValidPassword(password) {
  // Regular expression for password validation
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>


```
let passwordRegex =
/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d@$!%*?&]{8,}$/;
return passwordRegex.test(password);
}
```

// Test the function

```
console.log(isValidPassword("Passw0rd")); // true
console.log(isValidPassword("weakpass")); // false
```

Solution:

```
function isValidPassword(password) {
  let passwordRegex =
/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d@$!%*?&]{8,}$/;
  return passwordRegex.test(password);
}
```

Exercise 5: Extract Domain from URL

Description: Write a function that extracts the domain (excluding protocol and path) from a given URL.

Steps:

1. Define a regular expression to match the domain part of a URL.
2. Write a function that uses match to extract the domain.

Code Example:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
function extractDomain(url) {
  // Regular expression to match domain part of a URL
  let domainRegex =
/^((?:https?:\V)?(?:[^\@\V\n]+@)?(?:www\.)?([^\:\V\n]+))/;
  let match = url.match(domainRegex);
  return match ? match[1] : null;
}

// Test the function
console.log(extractDomain("https://www.example.com/path/to/page")); // 'example.com'
```

Solution:

```
function extractDomain(url) {
  let domainRegex =
/^((?:https?:\V)?(?:[^\@\V\n]+@)?(?:www\.)?([^\:\V\n]+))/;
  let match = url.match(domainRegex);
  return match ? match[1] : null;
}
```

Exercise 6: Validate Date Format

Description: Create a function that validates whether a given string represents a date in the format "MM/DD/YYYY".

Steps:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
 Courses <https://basescripts.com/>

1. Define a regular expression for validating the date format.
2. Write a function that uses the regular expression to check if the input is a valid date.

Code Example:

```
function isValidDateFormat(dateString) {  
  // Regular expression for MM/DD/YYYY format  
  let dateFormatRegex =  
  /^(0[1-9]|1[0-2])\V(0[1-9]|1[12][0-9]|3[01])\V\d{4}$/;  
  return dateFormatRegex.test(dateString);  
}
```

```
// Test the function  
console.log(isValidDateFormat("12/25/2023")); // true  
console.log(isValidDateFormat("31/12/2023")); // false
```

Solution:

```
function isValidDateFormat(dateString) {  
  let dateFormatRegex =  
  /^(0[1-9]|1[0-2])\V(0[1-9]|1[12][0-9]|3[01])\V\d{4}$/;  
  return dateFormatRegex.test(dateString);  
}
```

Exercise 7: Extract Hashtags from Text

Description: Write a function that extracts all hashtags from a given text.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Steps:

1. Define a regular expression to match hashtags.
2. Write a function that uses match to extract all hashtags from the input text.

Code Example:

```
function extractHashtags(text) {  
  // Regular expression to match hashtags  
  let hashtagRegex = /\B#\w+/g;  
  return text.match(hashtagRegex);  
}  
  
// Test the function  
console.log(extractHashtags("Check out this #awesome post!  
#coding #javascript")); // ['#awesome', '#coding', '#javascript']
```

Solution:

```
function extractHashtags(text) {  
  let hashtagRegex = /\B#\w+/g;  
  return text.match(hashtagRegex);  
}
```

Exercise 8: Validate Credit Card Number

Description: Create a function that validates whether a given string represents a valid credit card number.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Steps:

1. Define a regular expression for validating credit card numbers.
2. Write a function that uses the regular expression to check if the input is a valid credit card number.

Code Example:

```
function isValidCreditCardNumber(cardNumber) {  
  // Regular expression for credit card validation (simple example)  
  let creditCardRegex = /^\\d{4}-\\d{4}-\\d{4}-\\d{4}$/;  
  return creditCardRegex.test(cardNumber);  
}
```

```
// Test the function  
console.log(isValidCreditCardNumber("1234-5678-9012-3456"));  
// true  
console.log(isValidCreditCardNumber("invalid-card-number")); //  
false
```

Solution:

```
function isValidCreditCardNumber(cardNumber) {  
  let creditCardRegex = /^\\d{4}-\\d{4}-\\d{4}-\\d{4}$/;  
  return creditCardRegex.test(cardNumber);  
}
```

Exercise 9: Replace URLs with Links

Description: Write a function that replaces all URLs in a given text with clickable links.

Steps:

1. Define a regular expression to match URLs.
2. Write a function that uses `replace` to replace URLs with HTML links.

Code Example:

```
function replaceUrlsWithLinks(text) {  
  // Regular expression to match URLs  
  let urlRegex = /https?:\V\S+/g;  
  return text.replace(urlRegex, (url) => `target="_blank">${url}</a>`);  
}
```

```
// Test the function  
console.log(replaceUrlsWithLinks("Check out  
https://www.example.com and https://github.com!"));  
// 'Check out <a href="https://www.example.com"  
target="_blank">https://www.example.com</a> and <a  
href="https://github.com"  
target="_blank">https://github.com</a>!'
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Solution:

```
function replaceUrlsWithLinks(text) {
  let urlRegex = /https?:\V\S+/g;
  return text.replace(urlRegex, (url) => `<a href="${url}"
target="_blank">${url}</a>`);
}
```

Exercise 10: Validate Phone Number

Description: Create a function that validates whether a given string represents a valid phone number.

Steps:

1. Define a regular expression for validating phone numbers.
2. Write a function that uses the regular expression to check if the input is a valid phone number.

Code Example:

```
function isValidPhoneNumber(phoneNumber) {
  // Regular expression for phone number validation (simple
example)
  let phoneRegex = /^\\d{3}-\\d{3}-\\d{4}$/;
  return phoneRegex.test(phoneNumber);
}
```

// Test the function

```
console.log(isValidPhoneNumber("123-456-7890")); // true
console.log(isValidPhoneNumber("invalid-phone-number")); //
false
```

Solution:

```
function isValidPhoneNumber(phoneNumber) {
  let phoneRegex = /^\\d{3}-\\d{3}-\\d{4}$/;
  return phoneRegex.test(phoneNumber);
}
```

These exercises cover a range of scenarios where regular expressions can be applied. Feel free to customize them or create additional exercises to further enhance your understanding of regular expressions in JavaScript.

Regex Quiz Questions

Quiz Questions:

1. Question: What is a regular expression in JavaScript?
2. Question: Which object in JavaScript is used for working with regular expressions?
3. Question: What does the test() method of a RegExp object do?
4. Question: Which character is used as the delimiter for a regular expression literal?

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

5. Question: What does the `exec()` method of a `RegExp` object return?
6. Question: Which quantifier matches zero or more occurrences of the preceding element?
7. Question: What is the purpose of the `^` anchor in a regular expression?
8. Question: Which character class matches any digit character?
9. Question: In the regular expression `/ab+c/`, what does the `+` symbol represent?
10. Question: What is the purpose of the non-capturing group `(?:)` in a regular expression?
11. Question: What does the `?` quantifier indicate in a regular expression?
12. Question: Which assertion checks if a certain element is followed by another element without including the latter in the match?
13. Question: What is the purpose of the `g` flag in a regular expression?
14. Question: Which character is used for matching any character except a newline?
15. Question: In the regular expression `/[aeiou]/`, what does the character class `[aeiou]` represent?

16. Question: Which quantifier matches exactly three occurrences of the preceding element?
17. Question: What does the `replace()` method do when used with a regular expression?
18. Question: Which method is used to find all occurrences of a pattern in a string, including capturing groups?
19. Question: In the regular expression `/^[\\w.-]+@[a-z]+\\. [a-z]+$/i`, what does the `i` flag represent?
20. Question: What is the purpose of the `\\b` anchor in a regular expression?
21. Question: Which method returns an array of all occurrences of a pattern in a string without capturing groups?
22. Question: In the regular expression `/[^0-9]/`, what does the character class `[^0-9]` represent?
23. Question: Which quantifier matches between two and four occurrences of the preceding element?
24. Question: What does the `\\d` character class represent in a regular expression?
25. Question: In the regular expression `/^\\d{3}-\\d{2}-\\d{4}$/`, what does the pattern represent?
26. Question: What does the `matchAll()` method return?

27. Question: What does the `\S` character class represent in a regular expression?
28. Question: In the regular expression `/(\w+)\s(\w+)/`, what do the capturing groups `(\w+)` represent?
29. Question: Which character is used to escape a metacharacter in a regular expression?
30. Question: What is the purpose of the `search()` method in JavaScript when used with a regular expression?
31. Question: Which method splits a string into an array of substrings based on a regular expression?
32. Question: What is the purpose of the `\w` character class in a regular expression?
33. Question: In the regular expression `/(\d{2})\V(\d{2})\V(\d{4})/`, what do the capturing groups `(\d{2})`, `(\d{2})`, and `(\d{4})` represent?
34. Question: What is the purpose of the `\W` character class in a regular expression?
35. Question: In the regular expression `/(\b\w+\b)\s\1/`, what does `\1` represent?
36. Question: What does the `\b` anchor do in a regular expression?
37. Question: Which quantifier matches one or more occurrences of the preceding element?

38. Question: In the regular expression `/^[A-Z][a-z]*$/`, what does the pattern represent?
39. Question: What is the purpose of the `\n` escape sequence in a regular expression?
40. Question: In the regular expression `/(\d+)\s(?:years?|yrs?)`, what does `(?:years?|yrs?)` represent?
41. Question: What is the purpose of the `\s` character class in a regular expression?
42. Question: In the regular expression `/(\d{3})-(\d{2})-(\d{4})`, what does the capturing group `(\d{3})` represent?
43. Question: Which method is used to test if a string contains a pattern in a regular expression?
44. Question: What does the `\S+` pattern represent in a regular expression?
45. Question: In the regular expression `/^[a-zA-Z]\w*$/`, what does the pattern represent?
46. Question: What is the purpose of the `\` character in a regular expression?
47. Question: In the regular expression `/[^\aeiou]/`, what does the character class `[^\aeiou]` represent?
48. Question: What does the `flags` property of a `RegExp` object return?

49. Question: What is the purpose of the \$ anchor in a regular expression?
50. Question: In the regular expression `/([a-z]+\s\d+)/`, what do the capturing groups `([a-z]+)` and `(\d+)` represent?

Question: What is a regular expression in JavaScript?

- A) A string
- B) A data type
- C) A pattern describing a certain amount of text

Question: Which object in JavaScript is used for working with regular expressions?

- A) `RegexObject`
- B) `String`
- C) `RegExp`

Question: What does the `test()` method of a `RegExp` object do?

- A) Returns a new regular expression
- B) Tests a string for a match against a regular expression
- C) Replaces text in a string with a specified value

Question: Which character is used as the delimiter for a regular expression literal?

- A) /
- B) |
- C) #

Question: What does the exec() method of a RegExp object return?

- A) An array containing matched text
- B) The matched text
- C) null if no match is found

Question: Which quantifier matches zero or more occurrences of the preceding element?

- A) +
- B) *
- C) ?

Question: What is the purpose of the ^ anchor in a regular expression?

- A) Matches the end of a string
- B) Matches the start of a string
- C) Matches any character

Question: Which character class matches any digit character?

- A) \d
- B) \w
- C) \s

Question: In the regular expression /ab+c/, what does the + symbol represent?

- A) Matches the character "a"
- B) Matches the character "b"
- C) Matches one or more occurrences of the character "b"

Question: What is the purpose of the non-capturing group (?:) in a regular expression?

- A) Captures the matched text
- B) Creates a capturing group
- C) Groups expressions without capturing the matched result

Question: What does the ? quantifier indicate in a regular expression?

- A) Matches zero or one occurrence of the preceding element
- B) Matches one or more occurrences of the preceding element
- C) Matches any character

Question: Which assertion checks if a certain element is followed by another element without including the latter in the match?

- A) Positive Lookahead (?=)
- B) Negative Lookahead (?!)
- C) Positive Lookbehind (?<=)

Question: What is the purpose of the g flag in a regular expression?

- A) Global match (find all matches)
- B) Case-insensitive match
- C) Multiline match

Question: Which character is used for matching any character except a newline?

- A) .
- B) \n
- C) *

Question: In the regular expression `/[aeiou]/`, what does the character class `[aeiou]` represent?

- A) Matches any vowel
- B) Matches any consonant
- C) Matches any digit

Question: Which quantifier matches exactly three occurrences of the preceding element?

- A) {3}
- B) {3,}
- C) {0,3}

Question: What does the replace() method do when used with a regular expression?

- A) Removes text from a string
- B) Replaces text in a string with a specified value
- C) Returns a new string

Question: Which method is used to find all occurrences of a pattern in a string, including capturing groups?

- A) exec()
- B) test()
- C) matchAll()

Question: In the regular expression

`/^[\\w.-]+@[a-z]+\\. [a-z]+$/i`, what does the `i` flag represent?

- A) Case-sensitive match
- B) Global match
- C) Case-insensitive match

Question: What is the purpose of the `\\b` anchor in a regular expression?

- A) Matches a word boundary
- B) Matches the end of a string
- C) Matches the start of a string

Question: Which method returns an array of all occurrences of a pattern in a string without capturing groups?

- A) `exec()`
- B) `match()`
- C) `search()`

Question: In the regular expression `/[^\0-9]/`, what does the character class `[^\0-9]` represent?

- A) Matches any digit
- B) Matches any non-digit
- C) Matches any word character

Question: Which quantifier matches between two and four occurrences of the preceding element?

- A) `{2,4}`
- B) `{2}`
- C) `{4,}`

Question: What does the `\d` character class represent in a regular expression?

- A) Matches any digit
- B) Matches any non-digit
- C) Matches any word character

Question: In the regular expression `/^\d{3}-\d{2}-\d{4}$/`, what does the pattern represent?

- A) Social Security Number (SSN) format
- B) Date format
- C) Phone number format

Question: What does the `matchAll()` method return?

- A) The first match of a pattern
- B) An array of all matches of a pattern, including capturing groups
- C) The index of the first occurrence of a pattern

Question: What does the `\S` character class represent in a regular expression?

- A) Matches any whitespace character
- B) Matches any non-whitespace character
- C) Matches any word character

Question: In the regular expression `/(\w+)\s(\w+)/`, what do the capturing groups `(\w+)` represent?

- A) Matches any word character
- B) Captures the first and last names
- C) Creates a non-capturing group

Question: Which character is used to escape a metacharacter in a regular expression?

- A) !
- B) \
- C) |

Question: What is the purpose of the `search()` method in JavaScript when used with a regular expression?

- A) Searches for a pattern in a string and returns the index of the first match
- B) Replaces the first occurrence of a pattern in a string
- C) Tests a string for a match against a regular expression

Question: Which method splits a string into an array of substrings based on a regular expression?

- A) split()
- B) slice()
- C) splice()

Question: What is the purpose of the `\w` character class in a regular expression?

- A) Matches any whitespace character
- B) Matches any word character
- C) Matches any non-word character

Question: In the regular expression `/(\d{2})\(/(\d{2})\(/(\d{4})/`, what do the capturing groups `(\d{2})`, `(\d{2})`, and `(\d{4})` represent?

- A) Captures the day, month, and year in a date
- B) Matches any two digits
- C) Creates non-capturing groups

Question: What is the purpose of the `\W` character class in a regular expression?

- A) Matches any whitespace character
- B) Matches any non-word character
- C) Matches any word character

Question: In the regular expression `/(\b\w+\b)\s\1/`, what does `\1` represent?

- A) Matches the same word as captured in the first capturing group
- B) Matches any word boundary
- C) Creates a non-capturing group

Question: What does the `\b` anchor do in a regular expression?

- A) Matches a word boundary
- B) Matches any non-word character
- C) Matches any word character

Question: Which quantifier matches one or more occurrences of the preceding element?

- A) ?
- B) *
- C) +

Question: In the regular expression `/^[A-Z][a-z]*$/`, what does the pattern represent?

- A) Matches a sentence
- B) Matches a capitalized word
- C) Matches an email address

Question: What is the purpose of the `\n` escape sequence in a regular expression?

- A) Matches a newline character
- B) Matches a non-word character
- C) Matches a word character

Question: In the regular expression `/(\d+)\s(?:years?|yrs?)`, what does `(?:years?|yrs?)` represent?

- A) Captures the number of years
- B) Creates a non-capturing group for "years" or "yrs"
- C) Matches the literal characters "years" or "yrs"

Question: What is the purpose of the `\s` character class in a regular expression?

- A) Matches any whitespace character
- B) Matches any non-whitespace character
- C) Matches any word character

Question: In the regular expression `/(\d{3})-(\d{2})-(\d{4})`, what does the capturing group `(\d{3})` represent?

- A) Captures the area code in a phone number
- B) Matches any three digits
- C) Creates a non-capturing group

Question: Which method is used to test if a string contains a pattern in a regular expression?

- A) contains()
- B) includes()
- C) test()

Question: What does the `\S+` pattern represent in a regular expression?

- A) Matches one or more occurrences of any character
- B) Matches one or more occurrences of any non-whitespace character
- C) Creates a non-capturing group

Question: In the regular expression `/^[a-zA-Z]\w*$/, what does the pattern represent?`

- A) Matches an email address
- B) Matches a variable name in JavaScript
- C) Matches a sentence

Question: What is the purpose of the \ character in a regular expression?

- A) Represents an escape character
- B) Matches any character
- C) Creates a non-capturing group

Question: In the regular expression `/[^\aeiou]/`, what does the character class `[\aeiou]` represent?

- A) Matches any vowel
- B) Matches any non-vowel
- C) Matches any digit

Question: What does the `flags` property of a RegExp object return?

- A) The source pattern of the regular expression
- B) The index of the first match
- C) A string containing the flags used in the regular expression

Question: What is the purpose of the \$ anchor in a regular expression?

- A) Matches the end of a string
- B) Matches the start of a string
- C) Matches any character

Question: In the regular expression `/([a-z+)\s(\d+)/`, what do the capturing groups `([a-z]+)` and `(\d+)` represent?

- A) Captures a word and a number in a string
- B) Matches any lowercase letter and any digit
- C) Creates non-capturing groups

Quiz Answers

1. Answer: C) A pattern describing a certain amount of text
2. Answer: C) `RegExp`
3. Answer: B) Tests a string for a match against a regular expression
4. Answer: A) `/`
5. Answer: A) An array containing matched text
6. Answer: B) `*`
7. Answer: B) Matches the start of a string

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

8. Answer: A) `\d`
9. Answer: C) Matches one or more occurrences of the character "b"
10. Answer: C) Groups expressions without capturing the matched result
11. Answer: A) Matches zero or one occurrence of the preceding element
12. Answer: B) Negative Lookahead (`?!`)
13. Answer: A) Global match (find all matches)
14. Answer: A) .
15. Answer: A) Matches any vowel
16. Answer: A) `{3}`
17. Answer: B) Replaces text in a string with a specified value
18. Answer: C) `matchAll()`
19. Answer: C) Case-insensitive match
20. Answer: A) Matches a word boundary
21. Answer: B) `match()`
22. Answer: B) Matches any non-digit
23. Answer: A) `{2,4}`
24. Answer: A) Matches any digit
25. Answer: A) Social Security Number (SSN) format
26. Answer: B) An array of all matches of a pattern, including capturing groups
27. Answer: B) Matches any non-whitespace character
28. Answer: B) Captures the first and last names
29. Answer: B) `\`
30. Answer: A) Searches for a pattern in a string and returns the index of the first match
31. Answer: A) `split()`
32. Answer: B) Matches any word character
33. Answer: A) Captures the day, month, and year in a date
34. Answer: B) Matches any non-word character

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- 35. Answer: A) Matches the same word as captured in the first capturing group
- 36. Answer: A) Matches a word boundary
- 37. Answer: C) +
- 38. Answer: B) Matches a capitalized word
- 39. Answer: A) Matches a newline character
- 40. Answer: B) Creates a non-capturing group for "years" or "yrs"
- 41. Answer: A) Matches any whitespace character
- 42. Answer: B) Matches any three digits
- 43. Answer: C) `test()`
- 44. Answer: B) Matches one or more occurrences of any non-whitespace character
- 45. Answer: B) Matches a variable name in JavaScript
- 46. Answer: A) Represents an escape character
- 47. Answer: B) Matches any non-vowel
- 48. Answer: C) A string containing the flags used in the regular expression
- 49. Answer: A) Matches the end of a string
- 50. Answer: A) Captures a word and a number in a string