JS JavaScript TUTORIAL

Lookahead

Summary: in this tutorial, you'll learn about JavaScript regex lookahead to match `X` but if only it is followed by `Y` .

Introduction to JavaScript regex lookahead

In regular expressions, a lookahead allows you to match `X` but only if it is followed by `Y` .

Here's the syntax of the lookahead:

```
X(?=Y)
```

In this syntax, the regex engine searches for `X` and only matches if it is followed by `Y` .

For example, suppose you have the following string:

```
const s = '1 car is 15 feet long';
```

And you want to match the number 15 followed by a space and the literal string `feet` , not the number `1` . To do that, you can use a lookahead in a regular expression:

```
/\d+(?=\s*feet)/
```

In this regular expression:

- `\d+` matches one or more digits
- `?=` is the lookahead syntax
- `\s*` matches zero or more whitespaces
- `feet` matches the literal string `feet`

The following code uses the above pattern to match the number that is followed by zero or more spaces and the literal string `feet` :

```
const s = '1 car is 15 feet long';
const pattern = /\d+(?=\s*feet)/;

const match = s.match(pattern);
console.log(match);
```

Output:

```
15
```

Regex multiple lookaheads

It's possible to have multiple lookaheads in a regular expression with the following syntax:

```
X(?=Y)(?=Z)
```

In this syntax, the regex engine performs the following steps:

1. Search for X
2. Check if Y is immediately after X, skip if it isn't.
3. Check if Z is also immediately after Y; skip if it isn't.
4. If both tests pass, return X as a match; otherwise, search for the next match.

Therefore, the `X(?=Y)(?=Z)` matches `X` followed by `Y` and `Z` simultaneously.

Regex negative lookaheads

Suppose you want to match the number `1` but not the number `15` in the following string:

```
const s = '1 car is 15 feet long';
```

To do that, you use a negative lookahead. Here’s the syntax of the negative lookahead:

```
X(?!Y)
```

In this syntax, the regex engine matches `X` only if it is not followed by `Y`.

The following example illustrates how to use a negative lookahead:

```
const s = '1 car is 15 feet long';
const pattern = /\d+(?!\s*feet)/;

const match = s.match(pattern);
console.log(match);
```

Output:

```
1
```

Summary

- A regex lookahead `X(?=Y)` matches `X` only if it is followed by `Y`.
- A negative regex lookahead `X(?!Y)` matches `X` only if it is not followed by `Y`.