10th October 2019

Word boundary: \b

A word boundary `\b` is a test, just like `^` and `$`.

When the regexp engine (program module that implements searching for regexps) comes across `\b`, it checks that the position in the string is a word boundary.

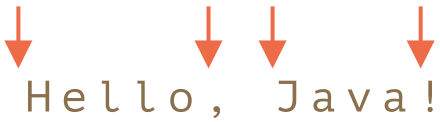
There are three different positions that qualify as word boundaries:

- At string start, if the first string character is a word character `\w`.
- Between two characters in the string, where one is a word character `\w` and the other is not.
- At string end, if the last string character is a word character `\w`.

For instance, regexp `\bJava\b` will be found in `Hello, Java!`, where `Java` is a standalone word, but not in `Hello, JavaScript!`.

```
1  alert( "Hello, Java!".match(/\bJava\b/) ); // Java
2  alert( "Hello, JavaScript!".match(/\bJava\b/) ); // null
```

In the string `Hello, Java!` following positions correspond to `\b`:



So, it matches the pattern `\bHello\b`, because:

1. At the beginning of the string matches the first test `\b`.
2. Then matches the word `Hello`.
3. Then the test `\b` matches again, as we're between `o` and a space.

The pattern `\bJava\b` would also match. But not `\bHell\b` (because there's no word boundary after `l`) and not `Java!\b` (because the exclamation sign is not a wordly character `\w`, so there's no word boundary after it).

```
1  alert( "Hello, Java!".match(/\bHello\b/) ); // Hello
2  alert( "Hello, Java!".match(/\bJava\b/) );  // Java
3  alert( "Hello, Java!".match(/\bHell\b/) );  // null (no match)
4  alert( "Hello, Java!".match(/\bJava!\b/) ); // null (no match)
```

We can use `\b` not only with words, but with digits as well.

For example, the pattern `\b\d\d\b` looks for standalone 2-digit numbers. In other words, it looks for 2-digit numbers that are surrounded by characters different from `\w`, such as spaces or punctuation (or text start/end).

```
1  alert( "1 23 456 78".match(/\b\d\d\b/g) ); // 23,78
2  alert( "12,34,56".match(/\b\d\d\b/g) ); // 12,34,56
```

Word boundary `\b` doesn't work for non-latin alphabets

The word boundary test `\b` checks that there should be `\w` on the one side from the position and "not `\w`" – on the other side.

But `\w` means a latin letter `a-z` (or a digit or an underscore), so the test doesn't work for other characters, e.g. cyrillic letters or hieroglyphs.

Tasks

Find the time

The time has a format: `hours:minutes`. Both hours and minutes has two digits, like `09:00`.

Make a regexp to find time in the string: `Breakfast at 09:00 in the room 123:456.`

P.S. In this task there's no need to check time correctness yet, so `25:99` can also be a valid result.

P.P.S. The regexp shouldn't match `123:456`.

solution

Previous lesson Next lesson

Share

Tutorial map

Comments

- If you have suggestions what to improve - please submit a GitHub issue or a pull request instead of commenting.
- If you can't understand something in the article – please elaborate.
- To insert a few words of code, use the `<code>` tag, for several lines – use `<pre>`, for more than 10 lines – use a sandbox (plnkr, JSBin, codepen…)