

Regular Expression: Sets and Ranges

Summary: in this tutorial, you will learn about the sets and ranges in regular expressions.

Sets

The square brackets search for any character in a set. For example, `[aeiou]` matches any of the five characters: `'a'` , `'e'` , `'i'` , `'o'` and `'u'` . The `[...]` is called a set.

For example, the regular expression `/[cbr]ats/g` matches `cats` , `bats` , and `rats` :

```
let str = 'How cats, rats, and bats became Halloween animals';
let re = /[cbr]ats/g;
let results = str.match(re);

console.log(results);
```

Output:

```
["cats", "rats", "bats"]
```

Ranges

The square brackets can contain character ranges. For example, `[a-z]` is a character range from `a` to `z` . And `[0-9]` is a digit from 0 to 9.

The `[a-zA-Z0-9_]` is the same as `\w` . The `[0-9]` is the same as `\d` .

Excluding ranges

To negate a range, you use the excluding range like: `[^...]` .

For example, `[^0-9]` matches any character except a digit. It is the same as `\D`.

Or, the `[^aeiou]` matches any character except `'a'`, `'e'`, `'i'`, `'o'` and `'u'`.

Escape special characters

Typically, you use a backslash to escape a special character e.g., `\.`. However, in square brackets, you don't need to escape most of the special characters except they have a meaning for the square brackets.

For example, if the caret (`^`) is at the beginning of a string, you need to escape it:

```
[\\^#$]
```

If the caret is not at the beginning of a string (`^`), you do not need to escape:

```
[#^$]
```

The flag u

If a set has surrogate pair, you need to add the flag `u` to the regular expression to make it work correctly:

```
let result = 'It is 🍏'.match(/[🍏🍏🍓]/);  
  
console.log(result);
```

Output:

```
["🍏"]
```

In this example, the `[🍏🍏🍓]` has six characters, not three:

```
let str = '🍏🍏🍓';
```

```
for(let i=0; i<str.length; i++) {  
    console.log(str.charCodeAt(i));  
}
```

Output:

```
55356  
57166  
55356  
57157  
55356  
57171
```

If you add the flag `u`, then the behavior will be correct:

```
let result = 'It is 🍎'.match(/[🍎🍌🍓]/u);  
  
console.log(result);
```

Output:

```
["🍎"]
```

Summary

- Use `[...]` to construct a set to match any character in it.
- Use the `-` inside a set to construct a range to match any character in the range.
- Use the `^` to negate a range.