

03 - Expressões Regulares no Javascript

Flags

- **g** global
- **i** case insensitive
- **m** multiline
- **s** single line

Caracteres Especiais

- **^** início da linha
- **\$** fim da linha
- **|** ou
- **+** ocorrências em sequência
- **?** caractere opcional (antes do sinal)
- ***** ocorrências em sequência de forma opcional
- **.** caractere qualquer
- **** escapa um caractere especial
- **\w** qualquer caractere alfanumérico
- **\d** qualquer caractere numérico
- **\s** espaços
- **\W** qualquer caractere que não é alfanumérico
- **\D** qualquer caractere que não é numérico
- **\S** qualquer caractere que não é espaço
- **{ }** encontra uma quantidade de caracteres entre um mínimo e um máximo
- **[]** agrupamento de regras
- **()** agrupamento de captura

Look Ahead e Look Behind

- **(?=)** positive look ahead: encontra termos que precedem um determinado termo

- (?!) negative look ahead: encontra termos que não precedem um determinado termo
- (?<=) positive look behind: encontra termos que são precedidos por um determinado termo
- (?<!) negative look behind: encontra termos que não são precedidos por um determinado termo

Recomendação de site para testar expressões regulares:

<https://regex101.com/>

<https://regexr.com/>

Código da Aula

```
function PhoneNumber(phoneNumberString) { const fixedString =  
phoneNumberString.replace(/[\sa-zA-Z]/g, "") this.countryCode =  
fixedString.match(/(?<=+)\d{1,3}/)[0] this.ddd = fixedString.match(/(?  
<=\\(\\d+(?=\\))/)[0] this.number = fixedString.match(/(?<=\\).+/  
[0].replace(/-/g, "") } console.log(new PhoneNumber('+55 (22) 9 9876-  
5432')) console.log(new PhoneNumber('+1 (555) a555-999-8888'))
```