



Lógica de Programação com JS

Laços Condicionais e numéricos

Sobre o que iremos falar

let vs const, laços...

- Recadinhos (Reposição de Sexta e Entregas)
 - Kahoot
 - autocorreção (sorteio 5 exercícios)
 - let / const
 - funções nativas * (Array, map, filter, reduce...)
-
- Spoilers *
-
- Laços numéricos (FOR, FOR-IN, FOR-OF)
 - Laços condicionais (WHILE, DO WHILE)
 - Desafios
-



Categorização de Parceiros

Uma empresa de engenharia precisa fazer a separação de seus parceiros/sócios em duas modalidades, parceiros PF, pessoas físicas no qual possuem CPF e PJ pessoas jurídicas no qual possuem CNPJ, dito isso temos a seguinte lista:

parceirosId	nome
19660156627897	Fernanda Santos
23998058019370	Rafael Souza
92291338611	Maria Silva
55443795656	Maria Souza
77743761456	Ana Costa
47202302326	Maria Ferreira
58017232567	Sofia Costa
16733009491247	Lucas Silva
63351859919	Rafael Souza
84297701780	Carlos Oliveira

O resultado final desse agrupamento deverá ser algo como:

```
const parceirosAgrupados = {
  PF: [
    { parceirosId: '12345678901', nome: 'Carlos Antonio' },
    { parceirosId: '12345678901', nome: 'Informática' },
  ],
  PJ: [
    { parceirosId: '12345678901234', nome: 'Empresa Xpto Ltda.' },
    { parceirosId: '12345678901235', nome: 'Construtora' },
  ]
}
```

Dicas:

- Utilize o conteúdo aprendido na aula de hoje.
- Leve em conta o parceirosId para fazer a diferenciação.
- Gere um Array de Objetos a partir da tabela apresentada.

Links úteis...

Para saber mais.

→ [Reduce](#)

→ [Hoisting](#)

Mais sobre hoisting / JS por trás dos panos:

<https://www.youtube.com/watch?v=M1Qvz7cRsq0>

<https://www.youtube.com/watch?v=q-iKinPQks>

■ ■ ■ ■ .map(■ → ●) → ● ● ● ●

■ ■ ● ■ .filter(■) → ■ ■ ■

● ● ■ ■ .find(■) → ■

● ● ● ■ .indexOf(■) → 3

■ ■ ■ ■ .fill(1, ●) → ■ ● ● ●

● ■ ■ ● .some(■) → true

■ ■ ■ ● .every(■) → false