FUNÇÕES JS

Conheça o que faz filter, map, e reduce

FILTER()

```
home > gustavo > Área de Trabalho > JS funcoes.js > ...
1  const pets = [
2      {nome: 'Cindy', tipo: 'cão', idade: 11},
3      {nome: 'Git', tipo: 'gato', idade: 4},
4      {nome: 'Python', tipo: 'cobra', idade: 1} ];
5  const newPets = pets.filter(
6      (pet) => {return pet.idade < 5} );
7  console.log(newPets);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[Running] node "/home/gustavo/Área de Trabalho/funcoes.js"
[
  { nome: 'Git', tipo: 'gato', idade: 4 },
  { nome: 'Python', tipo: 'cobra', idade: 1 }
]
```

Cria um novo array com todos os elementos que passaram no teste da função fornecida.

MAP()

```
home > gustavo > Área de Trabalho > JS funcoes.js > ...
1  const pets = [
2      {nome: 'Cindy', tipo: 'cão', idade: 11},
3      {nome: 'Git', tipo: 'gato', idade: 4},
4      {nome: 'Python', tipo: 'cobra', idade: 3} ];
5  const petNames = pets.map(
6      (pet) => {return pet.nome})
7  console.log(petNames);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[Running] node "/home/gustavo/Área de Trabalho/funcoes.js"
[ 'Cindy', 'Git', 'Python' ]
```

Percorre todo array (callback) e retorna um novo array apenas com a informação desejada como "nome" e com a mesma quantidade de elementos que o inicial.

REDUCE()

```
home > gustavo > Área de Trabalho > JS funcoes.js > pets
1  const pets = [
2      {nome: 'Cindy', tipo: 'cão', idade: 11, peso: 9},
3      {nome: 'Git', tipo: 'gato', idade: 4, peso: 5},
4      {nome: 'Python', tipo: 'cobra', idade: 3, peso: 2} ];
5  const totalPeso = pets.reduce(
6      (total, pet) => { return total + pet.peso; }, 0)
7  console.log(totalPeso);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[Running] node "/home/gustavo/Área de Trabalho/funcoes.js"
16

[Done] exited with code=0 in 0.12 seconds
```

Executa uma função para cada elemento retornando um único valor de retorno.
Supondo que precisa somar o peso de todos os animais e apenas mostrar o total.

LinkedIn & GitHub: @gustavohcsantos