Break e Continue:

Bem legal aprender a percorrer um array com um loop, né? Isso nos traz muitas possibilidades. Como, por exemplo, criar um sistema de busca por nome:

```
var listaDeNomes = ['Cesar', 'Pamela', 'Camila', 'Hendy']
var buscar = 'Cesar' //Nome que iremos buscar

for(var i =0; i < listaDeNomes.length; i++){
      if(listaDeNomes[i] == buscar){
            console.log('Encontrei o nome’)
      }
}
```

Viu, que legal? Aqui percorremos todo o nosso array, fazendo uma validação com o IF perguntando se naquela posição atual, o valor é igual ao da busca. Nesse caso, vemos que o nome “Cesar” estava na primeira posição. Mas, você percebeu que mesmo encontrando o valor na primeira posição, ele vai continuar o loop até chegar no final do array?

Puxa, seria um desperdício de esforço, não acha?

Realmente pode ser. Para resolvermos isso, quero te apresentar dois recursos que iremos aprender hoje: **Break** e **Continue**.

Começando com o break, ele nos ajudará justamente no problema acima. O break é um comando reservado que, ao ser processado pelo loop, irá parar todo o ciclo independente de quantos faltam. Olha só como ficaria:

```
for(var i =0; i < listaDeNomes.length; i++){
      if(listaDeNomes[i] == buscar){
            console.log('Encontrei o nome')
            break
      }
}
```

Com isso, o nosso loop irá parar assim que ele encontrar o nome que estamos esperando, evitando ficar percorrendo todo o nosso array. Legal, né? Dessa forma, ganhamos velocidade em encontrar os dados e o nosso computador não irá fazer ações desnecessárias.

Já o comando **continue** pula um ciclo! Vamos supor que queremos imprimir todos os nome contidos em nosso array, mas queremos pular o nome "Cesar". Veja só como ficaria no código:

```
var listaDeNomes = ['Cesar', 'Pamela', 'Camila', 'Hendy']
for(var i =0; i < listaDeNomes.length; i++){
      if(listaDeNomes[i] == 'Cesar'){
       continue
//Quando o nome for cesar, ele simplesmente passa para o próximo loop
      }
      console.log(listaDeNomes[i])
}

// Esse loop irá imprimir:

// Pamela
// Camila
// Hendy
```

Viu só? Isso é muito legal para os casos em que não queremos trabalhar com um determinado valor, passando para o próximo!