

Loop com array:

Você se lembra do tipo de dado “Array”? Isso mesmo, aquele que usamos para criar listas, ou colocar várias informações dentro de uma única variável! Exemplo para recordar:

```
var alimentos = [ "Pão", "Leite", "Café", "Queijo"]
```

Já sabemos que elas são muito úteis em várias situações, mas podemos nos deparar com uma situação um pouco complicada: trabalhar com seus valores. Por isso, quero que pense na seguinte tarefa: **para cada informação contida dentro da array**, queremos imprimir em uma frase assim:

“Esse alimento se chama: NOME”

Se fôssemos usar a primeira posição do array do exemplo anterior, seria algo como: **“Esse alimento se chama: Pão”** e assim por diante.

No código ficaria assim:

```
var frase = "Esse alimento se chama: " + alimentos[0]  
console.log(frase)//imprime "Esse alimento se chama: Pão"
```

Observe que há um espaço entre **:** e **"**

E, se quisermos que esse mesmo processo se **repita**, basta continuar fazendo a mesma coisa para as outras posições do nosso array:

```
var frase = "Esse alimento se chama: " + alimentos[0]  
console.log(frase)//imprime "Esse alimento se chama: Pão"  
  
var fraseDois = "Esse alimento se chama: " + alimentos[1]  
console.log(fraseDois)//imprime "Esse alimento se chama: Leite"  
  
var fraseTres = "Esse alimento se chama: " + alimentos[2]
```

```
console.log(fraseTres)//imprime "Esse alimento se chama: Café"
```

Enquanto tivermos apenas 3 ou 5 informações isso não será um problema, não é? Mas, se tivéssemos 100 alimentos para imprimir, seria um trabalho **repetitivo**?

Certamente! É por isso que podemos usar um loop para nos resolver esse problema. Afinal, todo o processo é igual. O que muda são as **posições** que queremos imprimir.

Vejamos passo-a-passo:

Estrutura do for:

```
for(contador; condição de parada; incremento){  
    //ação a ser repetida  
}
```

Essa estrutura já conhecemos, certo? Aqui será muito semelhante a exemplos passados, porém teremos que **dizer ao nosso loop que ele deve realizar essa ação em nossa array**. Qual ação?

Ele deverá **passar em cada valor**, independente de quantos sejam, para que assim possamos **trabalhar com cada um** de forma específica. A essa ação chamamos de **"percorrer um array"**.

Para isso, iremos indicar que nosso **contador comece do 0**, afinal esse é o valor/posição inicial da nossa array, e é por lá que nosso loop deve começar:

```
// geralmente definimos a variável i, que significa índice ou index  
for(var i = 0; condição de parada; incremento){  
    //ação a ser repetida  
}
```

Show! Nosso loop já sabe por onde ele começa, por isso agora precisamos dizer onde ele deve **terminar**. A ideia é a seguinte: o for irá fazer x repetições, e nós devemos indicar que, **para cada repetição, ele deve ir para um próximo valor da nossa array**.

Assim, quando ele chegar ao final das posições do nosso array, ele deve parar!

Para isso, precisamos saber quantas posições/valores existem em nossa array. Então, conta comigo: 1... 2...Brincadeira! Não precisamos contar, porque o próprio JavaScript já resolveu esse problema.

Para saber quantas posições existem em um array, devemos acessar uma informação chamada `length` (comprimento em português). Veja só:

```
var numeroDePosicoes = alimentos.length //Será 4
```

Bom, agora nos resta montar uma condição em nosso for, que faça ele parar quando chegar no final do array:

```
//Estamos dizendo que o for irá parar quando o contador chegar ao número total da nossa array
for(var i=0; i < alimentos.length; incremento){
    //ação a ser repetida
}
```

Com isso finalizamos, dando poder ao nosso contador de ir mudando. Caso contrário, o contador ficaria sempre em zero. Agora é hora de definirmos nosso incremento!

```
// Usamos o ++ para ir de valor em valor
for(let i=0; i < alimentos.length; i++){
    //ação a ser repetida
}
```

O que será repetido

Show, né? Agora que já configuramos o for para percorrer o nosso array, precisamos dizer **o que ele irá fazer!** Iremos definir no bloco de código, dentro das chaves do for:

```
// Usamos o ++ para ir de valor em valor
for(let i=0; i < alimentos.length; i++){
    console.log('Esse alimento se chama: ' + alimento[i])
}

//Esse loop irá imprimir:

// Esse alimento se chama: Pão           || que é o valor da posição 0
// Esse alimento se chama: Leite         || que é o valor da posição 1
// Esse alimento se chama: Café          || que é o valor da posição 2
// Esse alimento se chama: Queijo        || que é o valor da posição 3
```

Percebemos que no lugar da posição do array, colocamos a variável *i* na hora de imprimir a frase. A gente não poderia colocar uma posição fixa, porque assim ele iria imprimir sempre a mesma coisa. Como o foco aqui é trabalhar com todas as informações do array, a nossa variável *i* deve ser dinâmica, mudando de valor até chegar ao final do nosso array!

Agora não importa se nosso array tiver 5 ou 1000 itens, porque nosso código irá percorrer todos eles!