

# Loop for:

O Javascript tem diversos comandos para criar loop. Agora, vamos aprender um deles: o **for**!

O **for** utiliza um **contador** para definir quantas vezes o código deve ser repetido. O mais legal é que podemos decidir **em qual** número o contador deve **iniciar**, e também qual o **limite** em que deve **parar**.

## Sintaxe do for:

Esse comando é composto por **três partes principais** que ficam **entre parênteses** e são separadas por ponto e vírgula. As três partes são formadas por:

- **variável que vai guardar o número inicial do contador;**
- **condição de parada;**
- **incrementador.**

*Exemplo:*

```
for (inicio; condição de parada; incrementador) {  
  
}
```

*Vejamos cada uma dessas partes:*

**início:** determina em **qual número** o contador deve iniciar. Não se esqueça de criar uma variável para guardar esse valor!

**condição de parada:** determina em qual número o contador **deve parar**. Devemos criar uma **condicional**. Exemplo: contador < 10

**incrementador:** para cada vez que o **loop é executado**, nós **alteramos** o número do contador, **adicionando** +1. Por padrão utilizamos o operador ++.

Em um exemplo prático ficaria assim:

```
for (var contador = 1; contador < 10; contador++) {  
    // código que deve se repetir  
}
```

Dessa forma, o contador inicia com valor 1; para cada repetição ele deve adicionar +1(incremento); e continua repetindo enquanto a condição (contador < 10) for verdadeira.

Ufa, agora podemos melhorar aquele código de contar de 1 à 10.

Vamos ver como fica:

```
for (var contador = 1; contador < 10; contador++) {  
    console.log(1)  
}
```

Conforme o exemplo, observamos que o console.log é o comando que se repete, já que ele está dentro do loop. Entretanto, da maneira que está escrito ele irá repetir sempre o número 1, certo? Podemos resolver isso pedindo para que imprima o valor de uma variável no lugar de um valor fixo, dessa forma:

```
for (var contador = 1; contador < 10; contador++) {  
    console.log(contador)  
}
```

Dessa maneira a variável **contador** resolve nosso problema, porque ela vai ser incrementada a cada execução. Assim, o valor muda sempre, contando de 1 até 10 :)