```
//BREAK E CONTINUE:
//O break e o continue são palavras reservadas do Javascript que têm uma função parecida com palavras "go to" das linguagens de progr
amação mais antigas, que tinham a missão de desviar o fluxo do código para outro lugar. No Javascript o break e o continue agem da se
guinte forma:
//  break: pode ser aplicado somente sobre 3 estruturas de bloco, que são: "switch", "for" e "while". Sua missão é causar uma saída p
ara fora do bloco ao qual ele está definido;
//  continue: só pode ser aplicado sobre as estruturas: "for" e "while". Sua missão é pular uma determinada repetição de acordo com u
m desvio condicional qualquer que afete as repetições de alguma forma;
//OBSERVAÇÕES: ESSE É UM MÉTODO PRATICAMENTE EM DESUSO.
//Vejamos alguns exemplos:

//USANDO BREAK NUM LAÇO FOR:
const nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

for (let x in nums) { //Perceba que vamos iterar sobre o array acima...
    if (x == 5) { //Mas o if traz uma condição, se o x chegar no índice 5 o break irá causar uma saída para fora do laço "for"...
        console.log("O break aconteceu aqui!!");
        break //Note que embora o break esteja sobre o bloco de if, ele não causa uma saída do if, em vez disso ele vai procurar uma
estrutura de bloco "for", "while" ou "switch" e vai sair de dentro destas estruturas...
    }
    console.log(`1) ${x} = ${nums[x]}`);
}
console.log('\n');



//USANDO CONTINUE NUM LAÇO FOR:
for (let x in nums) { //Perceba que vamos iterar sobre o array acima...
    if (x == 5) { //Mas o if traz uma condição, se o x chegar no índice 5 o continue irá pular a impressão desse índice...
        console.log("Cade o 5? Percebe que faltou?");
        continue //Note que embora o continue esteja sobre o bloco de if, ele não causa um pulo na execução do if, em vez disso ele v
ai procurar uma estrutura de bloco "for" ou "while" mais próxima e vai fazê-la pular segundo as condições do desvio condicional...
    }
    console.log(`2) ${x} = ${nums[x]}`);
}
console.log('\n');
```

```
//USANDO BREAK NUM LAÇO WHILE:
n = 0;

while (n != 10) { //Perceba que o laço deveria contar até 10...
    n++;
    if (n == 5){ //Mas o if possuí um break que interage com o laço "while" fazendo-o sair do laço quando o valor chega a "5"...
        console.log('O break aconteceu!')
        break;
    }
    console.log('3)', n);
}
console.log('\n');



//USANDO CONTINUE NUM LAÇO WHILE:
m = 0;

while (m != 10) { //Perceba que o laço deveria contar até 10...
    m++;
    if (m == 5){ //Mas o if possuí um break que interage com o laço "while" fazendo-o sair do laço quando o valor chega a "5"...
        console.log('Pulamos o 5...')
        continue;
    }
    console.log('4)', m);
}
console.log('\n');



//USO ANTEPASSADO DE BREAK COM RÓTULO:
//A missão desses laços for é contar de 0,0 até 2,2...
externo: for (a in nums) { //Perceba que rotulamos uma parte do código para que o break afetasse o for externo e não o for ao qual el
e foi definido...
```

```
    for (b in nums) {
        if (a == 2 && b == 3) break externo; //Perceba que usamos o rótulo logo após o break para ativá-lo...
        console.log(`5) Par = ${a},${b}`);
    }
}
```

RESULTADO NO CONSOLE:

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\060-Break_e_Continue.js"
1) 0 = 1
1) 1 = 2
1) 2 = 3
1) 3 = 4
1) 4 = 5
O break aconteceu aqui!!


2) 0 = 1
2) 1 = 2
2) 2 = 3
2) 3 = 4
2) 4 = 5
Cade o 5? Percebe que faltou?
2) 6 = 7
2) 7 = 8
2) 8 = 9
2) 9 = 10


3) 1
3) 2
3) 3
3) 4
O break aconteceu!


4) 1
```

```
4) 2
4) 3
4) 4
Pulamos o 5...
4) 6
4) 7
4) 8
4) 9
4) 10


5) Par = 0,0
5) Par = 0,1
5) Par = 0,2
5) Par = 0,3
5) Par = 0,4
5) Par = 0,5
5) Par = 0,6
5) Par = 0,7
5) Par = 0,8
5) Par = 0,9
5) Par = 1,0
5) Par = 1,1
5) Par = 1,2
5) Par = 1,3
5) Par = 1,4
5) Par = 1,5
5) Par = 1,6
5) Par = 1,7
5) Par = 1,8
5) Par = 1,9
5) Par = 2,0
5) Par = 2,1
5) Par = 2,2

[Done] exited with code=0 in 0.172 seconds
```