

# Linguagem de Programação C

## Comandos de Iteração

Cristiano Lehrer

<http://www.ybadoo.com.br/>

## while (1/2)

```
while (condição)  
instrução;
```

- Funcionamento:
  - A condição é avaliada.
    - Se o resultado da avaliação for verdade (diferente de zero), é executada a instrução (ou bloco de instruções ali presente) associada ao `while`.
      - Volta-se ao ponto inicial.
    - Se o resultado da avaliação for falso (zero), o laço termina e o programa continua na instrução imediatamente depois do `while`.
- A instrução de um laço também é chamada de corpo do laço.
- Em um `while` coloca-se entre parênteses a condição que se tem que verificar para que a instrução ou bloco de instruções seja executado.

## while (2/2)

```
#include <stdio.h>
int main()
{
    int num;
    printf("Forneça o número desejado: ");
    scanf("%d", &num);
    printf("Tabuada do %d\n", num);
    int count = 0;
    while (count <= 10)
    {
        printf("%d * %2d = %2d\n", num, count, num * count);
        count = count + 1;
    }
    return 0;
}
```

## for (1/3)

**for** (cargas iniciais; condição; pós-instrução)  
instrução;

- Funcionamento:
  - O código presente em `cargas iniciais` é executado.
    - Normalmente aqui são iniciadas as variáveis presentes no laço. Esse componente do laço `for` é executado apenas uma única vez.
  - A `condição` é avaliada.
    - Se o resultado da avaliação for verdade (diferente de zero), então é executada a `instrução` (ou bloco de instruções) do laço.
      - Depois de executada a `instrução` presente no laço, é executada a `pós-condição`. Nesse componente do laço `for` são normalmente realizadas as alterações necessárias para passar à próxima iteração do laço (incremento ou decremento de variáveis).
      - Volta ao ponto inicial.
    - Se o resultado da avaliação for falso (zero), então o laço `for` termina e o programa continua na instrução imediatamente a seguir.

## for (2/3)

- O laço `for` identifica os seus três componentes, separando-os por ponto-e-vírgula (`;`).
- Assim, se for necessário realizar mais do que uma carga inicial ou mais do que uma pós-instrução, estas deverão ser separadas por vírgula (`,`).

```
for ( ; ; )
```

```
for (a = 0, b = 1; ; )
```

```
for ( ; ; a++, b--)
```

```
for (a = 0, b = 10; a < b; a++, b--)
```

## for (3/3)

```
#include <stdio.h>
int main()
{
    int n, num, soma, produto;
    printf("Forneça o numero desejado: ");
    scanf("%d", &num);
    for(soma = 0, n = produto = 1; n <= num; n = n + 1)
    {
        soma = soma + n;
        produto = produto * n;
    }
    printf("Soma = %d\nProduto = %d\n", soma, produto);
    return 0;
}
```

## do ... while (1/2)

```
do
    instrução;
while (condição);
```

- A `instrução` (ou bloco de instruções) é executada.
- A condição é avaliada.
  - Se o resultado da avaliação for verdade (diferente de zero), volta-se ao ponto inicial.
  - Se o resultado da avaliação for falso (zero), termina o laço e o programa continua na instrução seguinte ao laço.

## do ... while (2/2)

```
#include <stdio.h>
int main()
{
    int opcao;
    do
    {
        printf("\n\nMenu principal\n");
        printf("1\tClientes\n");
        printf("2\tFornecedores\n");
        printf("3\tSair\n");
        scanf("%d", &opcao);
        // Limpar o buffer do teclado
        fflush(stdin);
    }
```

```
switch (opcao)
{
    case 1:
        printf("Opção Clientes\n");
        break;
    case 2:
        printf("Opção Fornecedores\n");
        break;
    case 3: break;
    default:
        printf("Opção inválida!\n");
}
while (opcao != 3);
return 0;
}
```

## Comandos de Iteração – Resumo

	<b>while</b>	<b>for</b>	<b>do...while</b>
<b>Sintaxe</b>	<b>while</b> (condição) instrução;	<b>for</b> (carga inicial; condição; pós-instrução) instrução;	<b>do</b> instrução; <b>while</b> (condição)
<b>Executa a instrução</b>	zero ou mais vezes	zero ou mais vezes	uma ou mais vezes
<b>Testa a condição</b>	antes da instrução	antes da instrução	depois da instrução
<b>Utilização</b>	frequente	frequente	pouco frequente

## break

- A instrução `break` pode ser utilizada para terminar uma sequência de instruções dentro de um `switch` ou para terminar um laço.

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i < 100; i++)
    {
        if(i == 30)
            break;
        else
            printf("%2d\n", i);
    }
    return 0;
}
```

## continue

- A instrução `continue`, quando presente dentro de um laço, passa para a próxima iteração.
- A instrução `continue` só pode ser utilizada dentro de laços, enquanto o `break` pode ser utilizado em laços ou na instrução `switch`.

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i < 100; i++)
    {
        if(i == 30)
            continue;
        else
            printf("%2d\n", i);
    }
    return 0;
}
```