

Loops

domingo, 1 de junho de 2025 21:35

Definição: é uma sequência de repetições que acontecem até que a condição seja verdadeira.

Estrutura em C++:

```
while (condição){  
//código a ser executado;  
}
```

OBS: // --> significa comentário, o compilador(responsável por ler o seu código traduzir para assembly e depois bits) vai ignorar tudo que vier após o // na linha.

Quando e por que usariam loops?

Usamos loopings para evitar a repetição de códigos.

Ex: Se eu quero printar uma sequência de números de 1 a 10. É muito mais prático usar um loop do que dar cout em 1,2,3,4,5...,10.

```
int contador = 1;  
while (contador <= 5) {  
    cout << contador << endl;  
    contador++;  
}
```

Se eu tirar esse =, quantas vezes executaria? Resposta: 4.

Perguntas:

- 1) Como faço se quiser printar os números de 1 até 10?

Resposta:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int i = 1;  
    while (i <= 10) {  
        cout << i << " ";  
        i++;  
    }  
    return 0;  
}
```

2) Printar os pares de 1 até 20:

```
#include <iostream>
using namespace std;

int main() {
    int i = 2;
    while (i <= 20) {
        cout << i << " ";
        i += 2;
    }
    return 0;
}
```

3) Contar de 10 até 1 de forma decrescente?

```
#include <iostream>
using namespace std;

int main() {
    int i = 10;
    while (i >= 1) {
        cout << i << " ";
        i--;
    }
    return 0;
}
```

4) Mostrar de um até o número pedido pelo usuário?

```
#include <iostream>
using namespace std;

int main() {
    int n, i = 1;
    cin >> n;
    while (i <= n) {
        cout << i << " ";
        i++;
    }
    return 0;
}
```

5) Pedir senha até acertar:

```
#include <iostream>
using namespace std;

int main() {
    int senha;
    cout << "Digite a senha: ";
    cin >> senha;

    while (senha != 1234) {
        cout << "Senha incorreta, tente novamente: ";
        cin >> senha;
    }

    cout << "Acesso permitido!";
    return 0;
}
```

6) Somar 5 números digitados pelo usuário:

```
#include <iostream>
using namespace std;

int main() {
    int i = 1, num, soma = 0;

    while (i <= 5) {
        cin >> num;
        soma += num;
        i++;
    }

    cout << "Soma = " << soma;
    return 0;
}
```

- 7) Somar de um até 100:

```
#include <iostream>
using namespace std;

int main() {
    int i = 1, soma = 0;

    while (i <= 100) {
        soma += i;
        i++;
    }

    cout << "Soma = " << soma;
    return 0;
}
```

- 8) Somar os números digitados pelo usuário até 0:

```
#include <iostream>
using namespace std;

int main() {
    int num, soma = 0;
    cin >> num;

    while (num != 0) {
        soma += num;
        cin >> num;
    }

    cout << "Soma = " << soma;
    return 0;
}
```

- 9) Tabuada de um número digitada pelo usuário:(No formato 1 x 2)

```
#include <iostream>
using namespace std;

int main() {
    int n, i = 1;
    cin >> n;

    while (i <= 10) {
        cout << n << " x " << i << " = " << n * i << endl;
        i++;
    }
    return 0;
}
```

- 10) Fatorial de x:

```

#include <iostream>
using namespace std;

int main() {
    int n, f = 1, i = 1;
    cin >> n;

    while (i <= n) {
        f *= i;
        i++;
    }

    cout << "Fatorial = " << f;
    return 0;
}

```

- 11) Maior número digitado pelo usuário, parando quando ele digitar -1:

```

#include <iostream>
using namespace std;

int main() {
    int i=0, num, maior;
    cin >> num;

    while (num!=-1) {
        if(i==0){
            maior = num;
        }
        else if (num > maior) {
            maior = num;
        }
        cin >> num;
        i++;
    }

    cout << "Maior numero = " << maior;
    return 0;
}

```

- 12) Em uma dada quantidade de vezes(digitadas pelo usuário), contar a quantidade de pares e ímpares que o usuário digitou:

```

#include <iostream>
using namespace std;

int main() {
    int num, qtd, pares = 0, impares = 0, i=0;
    cin >> qtd;

    while (i < qtd) {
        cin >> num;
        if (num % 2 == 0)
            pares++;
        else
            impares++;
        i++;
    }

    cout << "Pares = " << pares << endl;
    cout << "Impares = " << impares;
    return 0;
}

```

13) Verificar número primo:

```

#include <iostream>
using namespace std;

int main() {
    int n, i = 2;
    bool primo = true;

    cin >> n;

    while (i < n && primo) {
        if (n % i == 0) {
            primo = false;
        }
        i++;
    }

    if (primo && n > 1)
        cout << "Primo";
    else
        cout << "Nao primo";

    return 0;
}

```

14) Jogo de adivinhação:

```

#include <iostream>
using namespace std;

int main() {
    int segredo = 7, chute;
    cout << "Adivinhe o numero secreto: ";
    cin >> chute;

    while (chute != segredo) {
        cout << "Errou! Tente de novo: ";
        cin >> chute;
    }

    cout << "Acertou!";
    return 0;
}

```

- 15) Desafio: Peça um intervalo para o usuário e printe a quantidade de números primos dados esse intervalo:
 Dicas: Crie a variável início e fim do intervalo.

```

#include <iostream>
using namespace std;

int main() {
    int inicio, fim, num;
    //criando uma variavel para verificar se há primos
    bool primos0 = false;
    cout << "Digite o inicio: ";
    cin >> inicio;
    cout << "Digite o fim: ";
    cin >> fim;

    num = inicio;

    while (num <= fim) {
        // Verifica se 'num' é primo
        int i = 2;
        bool primo = true;

        if (num < 2) {
            primo = false; // Números menores que 2 não são primos
        } else {
            while (i < num && primo) {
                if (num % i == 0) {
                    primo = false;
                }
                i++;
            }
        }

        if (primo) {
            cout << num << " é primo" << endl;
            primos0 = true;
        }
        num++;
    }
    if (!primos0){
        cout << 0 << " primos" << endl;
    }
    return 0;
}

```

- 16) Desafio mesclando essa aula com a passada:

16) Desafio mesclando essa aula com a passada:

```
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Quantos termos da sequencia de Fibonacci voce quer? ";
    cin >> n;

    int a = 0, b = 1, i = 1;

1.    while (i <= n) {
        cout << a << " ";
        int prox = a + b;
        a = b;
        b = prox;
        i++;
    }

    return 0;
}
```

Como funciona a sequência de Fibonacci?

- Ela começa sempre com **0 e 1**.
- Cada número seguinte é a **soma dos dois anteriores**.

Exemplo passo a passo:

0, 1 → já são os dois primeiros
0 + 1 = 1 → terceiro número
1 + 1 = 2 → quarto número
1 + 2 = 3 → quinto número
2 + 3 = 5 → sexto número
3 + 5 = 8 → sétimo número

Como pensar no algoritmo

1. O usuário escolhe **quantos termos** quer ver (N).
2. Guardamos as duas primeiras variáveis:
 - a = 0
 - b = 1
3. Vamos repetir N vezes:
 - Mostrar a (o primeiro número da vez).
 - Calcular o **próximo** número: prox = a + b.
 - Atualizar os valores:
 - a recebe b (anda uma casa para frente).
 - b recebe prox (anda outra casa para frente).

Representação em tabela

Se n = 7, a execução fica assim:

Passo (i) a (imprime) b prox = a+b

1	0	1	1
2	1	1	2
3	1	2	3
4	2	3	5
5	3	5	8
6	5	8	13
7	8	13	21

🖨 Resultado impresso: **0 1 1 2 3 5 8**

Ligando com o while

- Criamos um contador $i = 1$.
- Enquanto $i \leq n$, repetimos:
 - Mostrar a.
 - Atualizar variáveis (a, b, prox).
 - Incrementar i.

Ou seja, o truque é usar duas variáveis (a e b) que sempre guardam os últimos dois números da sequência.

Roteiro de aula

quarta-feira, 20 de agosto de 2025 22:12

Começar perguntando oq é um loop?

Definição: é uma sequência de repetições que acontecem até que a condição seja verdadeira.

Por que eu ia querer um loop?

Quando eu quero fazer mais de uma coisa igual, por ex: se eu quero printar do 1 até o 5. 10, 100...

Existem dois comandos para fazer loopings o for e o while. Nessa aula exploraremos o while.

Para usar o while:

```
while (condição){  
//código a ser executado  
}
```