

for



```
for(int nota = 0; nota < 10; nota++)
```

Syntax

```
for(int i = 0; i < 10; i++) {  
  
}
```

Syntax

Inicialização

Condição

Incremento

```
for(int i = 0; i < 10; i++) {  
  
}
```

Syntax

Inicialização

Condição

Incremento

```
for(int i = 0; i < 10; i++) {  
  
}
```

Declaração de **uma ou mais** variáveis

Utilizadas apenas dentro do **for**

Exemplos de Inicialização

Todas as variáveis
são do **mesmo tipo**

São separadas por
vírgulas

```
for(int i = 0, j = 3, x = 2; i < 10; i++) {  
  
} // 3 variáveis do tipo int
```

```
for(char letra = 'A', numero = '0'; letra < 'Z'; letra++) {  
  
} // 2 variáveis do tipo char
```

```
for(double peso = 0.0; peso <= 90; peso++) {  
  
} // 1 variáveis do tipo double
```

Syntax

Inicialização

Condição

Incremento

```
for(int i = 0; i < 10; i++) {  
  
}
```

Expressão lógica, numérica e relacional

Syntax

Inicialização

Condição

Incremento

```
for(int i = 0; i < 10; i++) {  
  
}
```

Operações em variáveis: **incrementos e decrementos**

Syntax

Inicialização

Condição

Incremento

```
for(int i = 0; i < 10; i++) {  
  
}
```

Operações em variáveis: **incrementos e decrementos**

Uma ou mais operações

Exemplos de Incremento

São separadas por
vírgulas

```
for(int i = 0, j = 3; i < 10; i++, j *= 2) {  
  
}
```

```
for(int i = 0, j = 3; i < 10; i += 5, j /= 10) {  
  
}
```

```
for(int i = 0, j = 3; i < 10; i %= 2, j -= 5) {  
  
}
```

Operadores Contraídos

a 😊 = 😄

a = a 😊 😄

Operadores Contraídos

`a += 3`

`a = a + 3`

Operadores Contraídos

$a *= 10$

$a = a * 10$

Operadores Contraídos

$+=$ $/=$

$-=$ $\%=$

$*=$

Exemplos de Incremento

São separadas por
vírgulas

```
for(int i = 0, j = 3; i < 10; i++, j *= 2) {  
  
}
```

```
for(int i = 0, j = 3; i < 10; i += 5, j /= 10) {  
  
}
```

```
for(int i = 0, j = 3; i < 10; i %= 2, j -= 5) {  
  
}
```

**Como funciona o
for?**

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```


Início do loop

➡ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

Início do loop



Inicialização da variável i com o valor
0

➔ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

Início do loop



Inicialização da variável i com o valor
0

Condição é testada

➡ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

```
for(int i = 0; i < 3; i++) {
```



```
    printf("i = %d\n", i);
```

```
}
```

Imprimir: i = 0

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```



Fim do loop

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
    → }
```

Fim do loop

Execução do Incremento

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
    ➡ }
```

Fim do loop

Execução do Incremento

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);
```

➡ } // i++ ---> i = i + 1 ---> i = 1

2° loop

➡ `for(int i = 0; i < 3; i++) {
 printf("i = %d\n", i);
}`

2º loop

Condição é testada

➡ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

2º loop

Condição é testada

→ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

Imprimir: i = 1

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```



Fim do loop

Execução do Incremento

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);
```

➡ } // i++ ---> i = i + 1 ---> i = 2

3º loop

Condição é testada

➡ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

3º loop

Condição é testada

2 < 3

→ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

Imprimir: i = 2

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```



Fim do loop

Execução do Incremento

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);
```

➡ } // i++ ---> i = i + 1 ---> i = 3

4º loop

Condição é testada

➡ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

4º loop

Condição é testada

3 < 3

→ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

4º loop

Condição é testada

Condição é **falsa**

3 < 3

➔ `for(int i = 0; i < 3; i++) {`
 `printf("i = %d\n", i);`
}

Saiu do Loop

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```



**Coisas para se
notar**

Inicialização

Executada 1 vez no início do primeiro loop

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```

Condição

Testada toda vez no início de cada loop

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```


Incremento

Realizado ao fim de cada loop

```
for(int i = 0; i < 3; i++) {  
    printf("i = %d\n", i);  
}
```

break

break: Interrompe o loop

```
for(int i = 1; i <= 10; i++) {  
    if (i % 5 == 0) { // i multiplo de 5  
        break; // Interrompe o loop  
    }  
}
```

break: Interrompe o loop

```
for(int i = 1; i <= 10; i++) {  
    if (i % 5 == 0) {  
        break;  
    }  
}
```

Não checa nenhuma
condição

Sai do loop
imediatamente

continue

continue: Termina aquela iteração

```
for(int i = 1; i <= 15; i++) {  
    if (i % 5 == 0) { // i multiplo de 5  
        continue; // vai para a próxima iteração  
    }  
    printf("i = %d\n", i);  
}
```

continue: Termina aquela iteração

```
for(int i = 1; i <= 15; i++) {  
    if (i % 5 == 0) {  
        continue;  
    }  
    printf("i = %d\n", i);  
}
```

Termina o laço sem
executar as próximas
linhas

Realiza o incremento

Testa a condição

**Testem: Some todos os
números de 0 a N**

**Testem 2: Some todos
os números múltiplos
de 3 e 2, de 0 a N**

continue



Na próxima aula