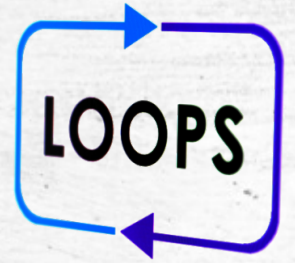


09

Comandos de Desvio

Fundamentos de Programação

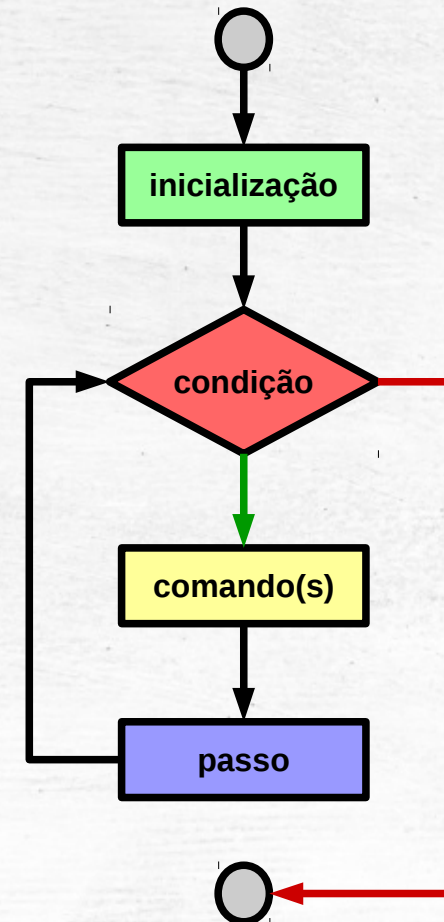
Revisão



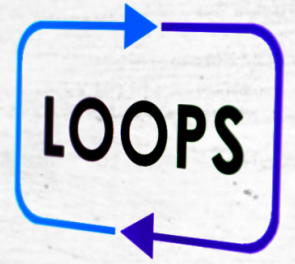
- Repetição com número definido de repetições

```
PARA inicialização ATÉ condição FAÇA passo  
INÍCIO  
    comando1  
    comando2  
    ...  
    comandon  
FIM
```

```
for(inicialização ; condição ; passo){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```



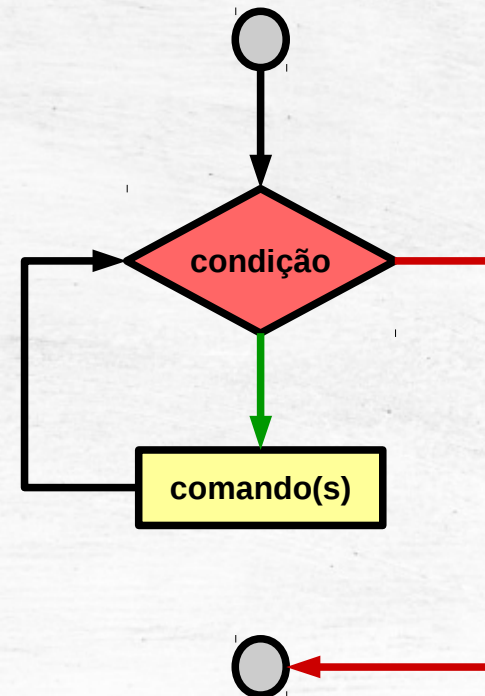
Revisão



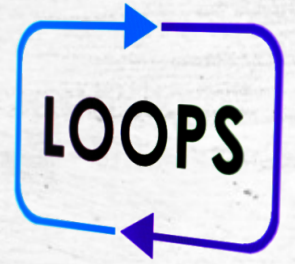
- Número Indefinido de Repetições e Teste no Início

```
ENQUANTO condição FAÇA  
INÍCIO  
    comando1;  
    comando2;  
    ...  
    comandon;  
FIM
```

```
while(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```



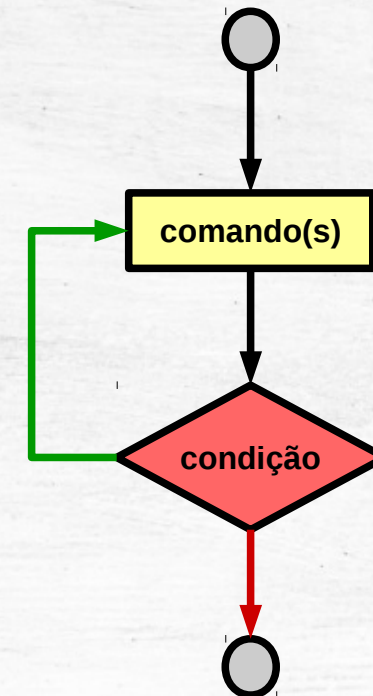
Revisão



- Número Indefinido de Repetições e Teste no Final

```
FAÇA  
INÍCIO  
    comando1;  
    comando2;  
    ...  
    comandon;  
FIM  
ENQUANTO condição
```

```
do{  
    comando1;  
    comando2;  
    ...  
    comandon;  
}while(condição);
```



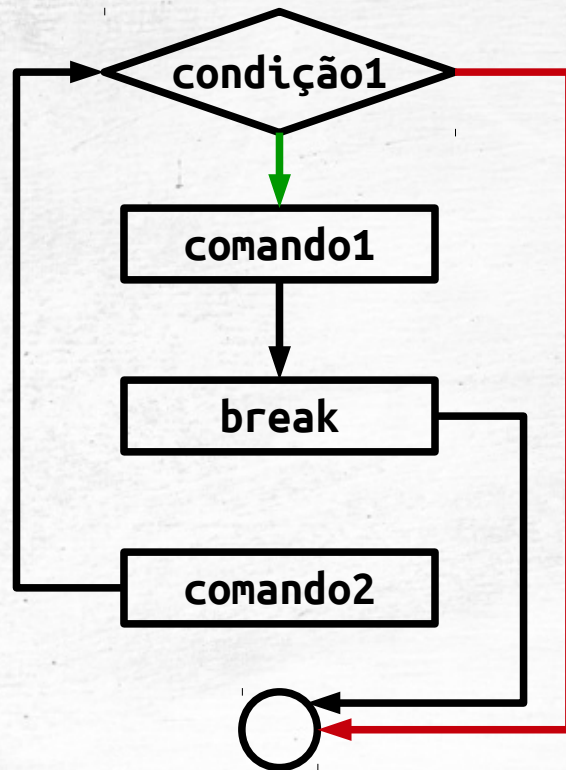
O comando **break**

```
switch(expressão){  
  case constante1: comando1; break;  
  case constante2: comando2; break;  
  ...  
  case constanten: comandon; break;  
  default:          comando; break;  
}
```

Impede comandos seguintes de executarem

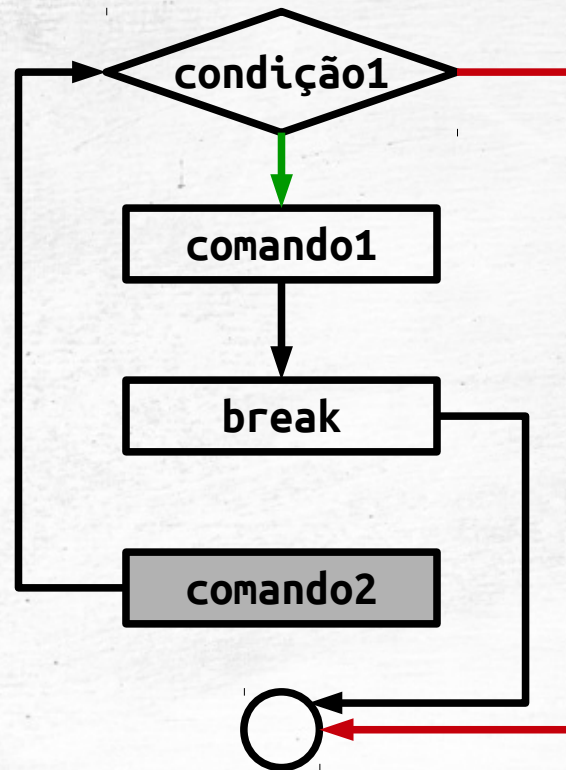
O comando **break**

- Interrompe a execução de um laço imediatamente



O comando break

- Interrompe a execução de um laço imediatamente



*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente

Memória

```
i =  
n =  
soma =
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    break;  
    soma = soma + n;  
}
```


O comando `break`

- Interrompe a execução de um laço imediatamente

Memória

```
i =  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1; i <= 10; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente

Memória

```
i = 1  
n =  
soma = 0
```

Tela

Numero 1:

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 : i <= 10 : i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```


O comando `break`

- Interrompe a execução de um laço imediatamente

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```

O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

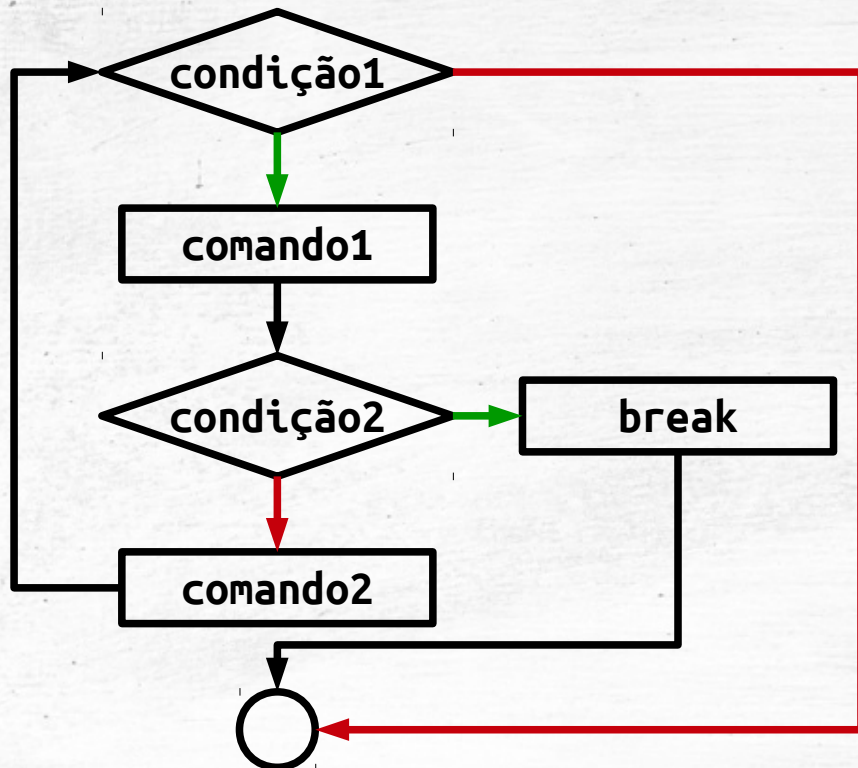
Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    break;  
    soma = soma + n;  
}
```


O comando break

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

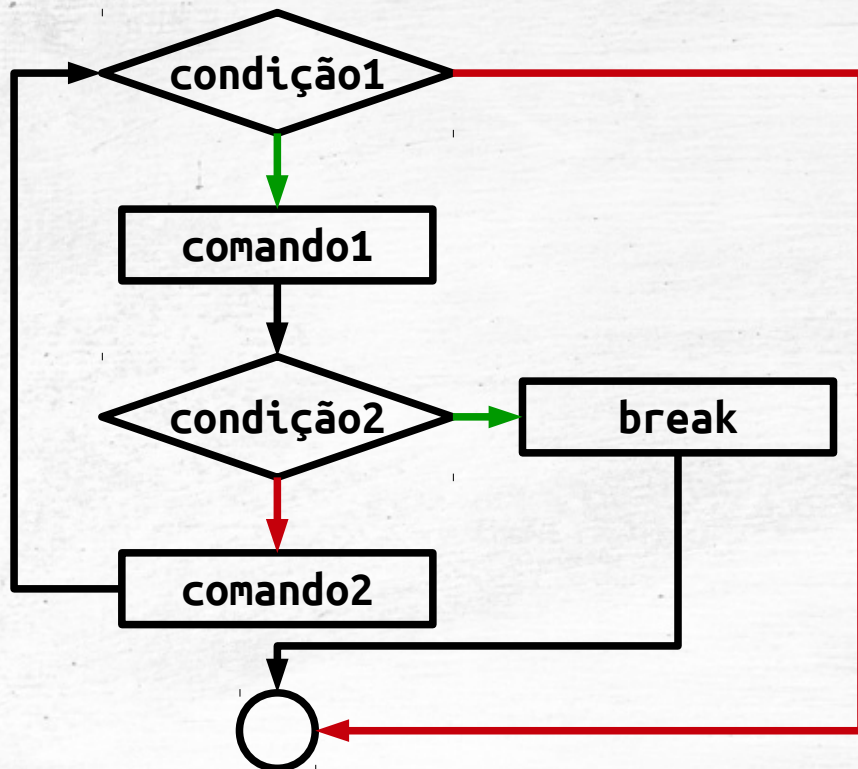


*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    break;  
    soma = soma + n;  
}
```

O comando break

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição



*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i =  
n =  
soma =
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i =  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1; i <= 10; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 1  
n =  
soma = 0
```

Tela

Numero 1:

Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 7
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 2  
n = 7  
soma = 7
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 2  
n = 7  
soma = 7
```

Tela

Numero 1: 7

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 2  
n = 7  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2:
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 2  
n = 4  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: 4
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 2  
n = 4  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: 4
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 2  
n = 4  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 3  
n = 4  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando `break`

- Interrompe a execução de um laço imediatamente
- Quase sempre, `break` está associado a alguma condição

Memória

```
i = 3  
n = 4  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 3  
n = 4  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3:
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 3  
n = 9  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 3  
n = 9  
soma = 11
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 3  
n = 9  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = 9  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = 9  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = 9  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9  
Numero 4:
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = -1  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9  
Numero 4: -1
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = -1  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9  
Numero 4: -1
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = -1  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9  
Numero 4: -1
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```


O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

Memória

```
i = 4  
n = -1  
soma = 20
```

Tela

```
Numero 1: 7  
Numero 2: 4  
Numero 3: 9  
Numero 4: -1
```

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

O comando **break**

- Interrompe a execução de um laço imediatamente
- Quase sempre, **break** está associado a alguma condição

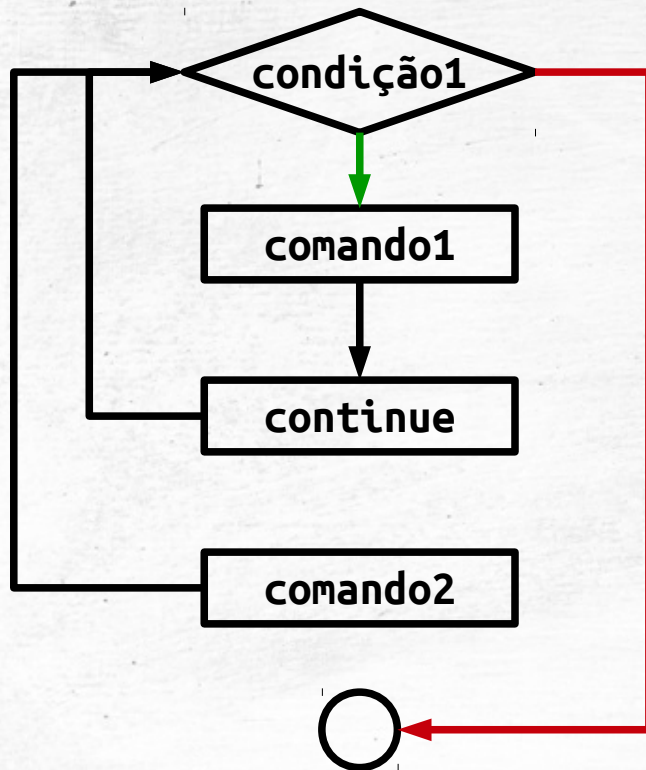
Normalmente se usa o comando **break** quando uma condição especial deve provocar uma interrupção imediata.

*Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

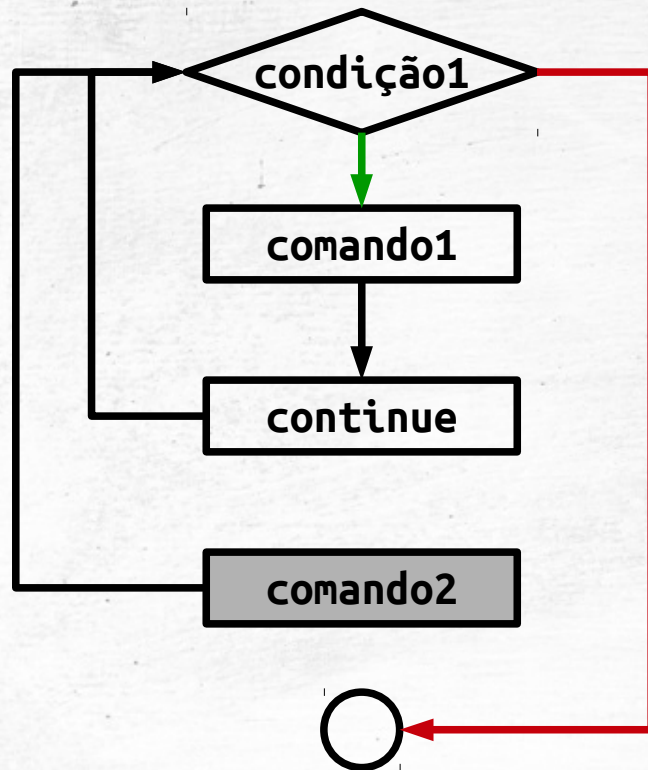

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois



O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois



*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i =  
n =  
soma =
```

Tela

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i =  
n =  
soma = 0
```

Tela

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1; i <= 5; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 1  
n =  
soma = 0
```

Tela

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 1  
n =  
soma = 0
```

Tela

Numero 1:

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 2  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 2  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 2  
n = 7  
soma = 0
```

Tela

```
Numero 1: 7  
Numero 2:
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois

Memória

```
i = 2  
n = 3  
soma = 0
```

Tela

```
Numero 1: 7  
Numero 2: 3
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 2  
n = 3  
soma = 0
```

Tela

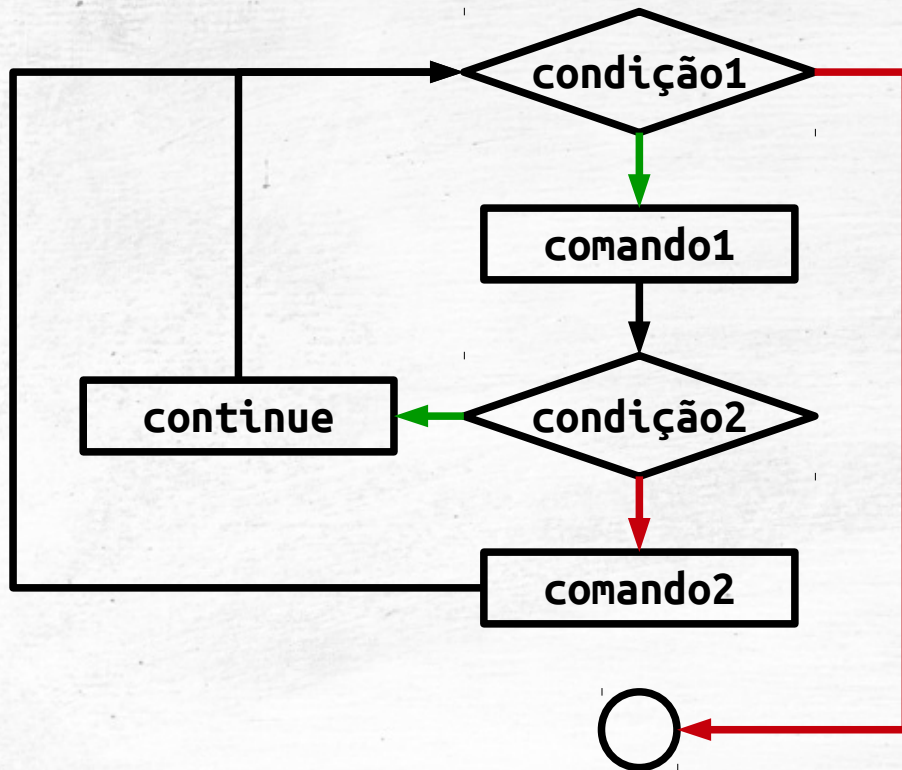
```
Numero 1: 7  
Numero 2: 3
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

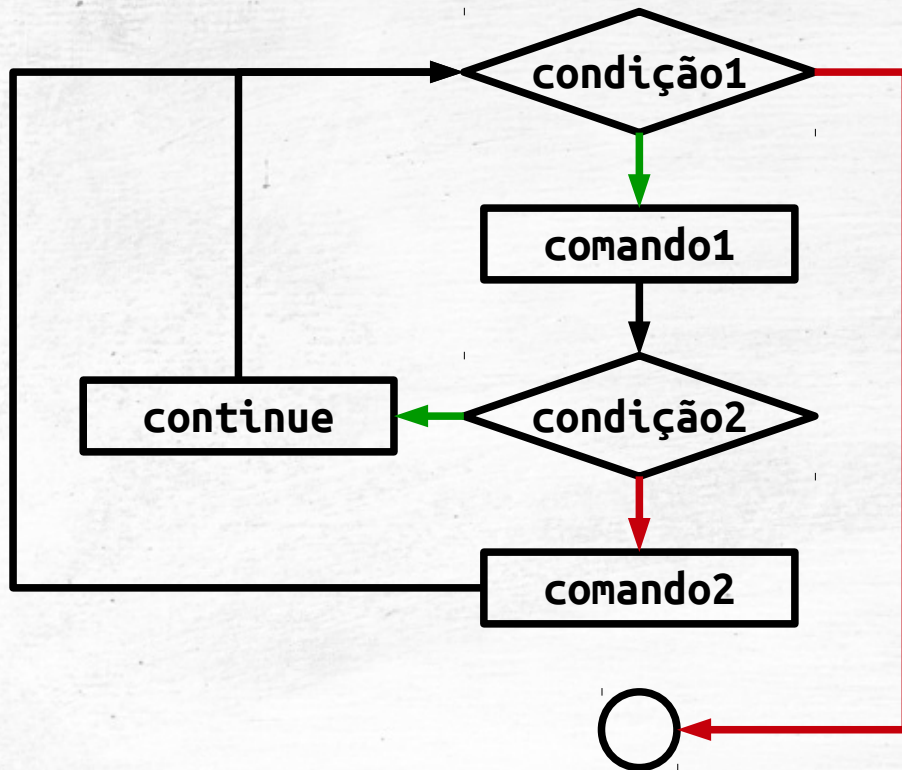


*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição



*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i =  
n =  
soma =
```

Tela

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 1  
n =  
soma = 0
```

Tela

Numero 1:

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 0
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 1  
n = 7  
soma = 7
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 2  
n = 7  
soma = 7
```

Tela

Numero 1: 7

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 2  
n = 7  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2:
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 2  
n = -2  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: -2
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 3  
n = -2  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: -2
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 3  
n = -2  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3:
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 3  
n = 3  
soma = 7
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 3  
n = 3  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 4  
n = 3  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 4  
n = 3  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4:
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 4  
n = -1  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 5  
n = -1  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 5  
n = -1  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1  
Numero 5:
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 5  
n = 5  
soma = 10
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1  
Numero 5: 5
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 5  
n = 5  
soma = 15
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1  
Numero 5: 5
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 6  
n = 5  
soma = 15
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1  
Numero 5: 5
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Memória

```
i = 6  
n = 5  
soma = 15
```

Tela

```
Numero 1: 7  
Numero 2: -2  
Numero 3: 3  
Numero 4: -1  
Numero 5: 5
```

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ", i);  
    scanf("%d", &n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```


O comando `continue`

- Força que ocorra a próxima iteração do laço, pulando código que vier depois
- Quase sempre, `continue` está associado a alguma condição

Normalmente se usa o comando `continue` quando deseja-se filtrar a execução do laço a uma condição específica

*Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!*

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

Conclusão

- Nenhum dos dois comandos é essencial

Calcule a soma de 5 numeros inteiros
Obs: Pule os números negativos!

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) continue;  
    soma = soma + n;  
}
```

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 5 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n >= 0)  
        soma = soma + n;  
}
```


Conclusão

- Nenhum dos dois comandos é essencial

Porém, ambos podem tornar os códigos mais simples de escrever e de entender.

Calcule a soma de 10 numeros naturais
Obs: Pare quando tiver um número negativo!

```
int i, n;  
int soma = 0;  
  
for(i = 1 ; i <= 10 ; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n < 0) break;  
    soma = soma + n;  
}
```

```
int i, n = 0;  
int soma = 0;  
  
for(i = 1 ; i <= 10 && n >= 0; i++){  
    printf("Numero %d: ",i);  
    scanf("%d",&n);  
    if(n >= 0)  
        soma = soma + n;  
}
```

