

Aula 03 - Operadores, controle de fluxo

Assuntos

1. Operadores
2. if... else if... else
3. Operador ternário
4. switch

Operadores de comparação e lógicos

operador	significado
>	maior
<	menor
==	igual
!=	diferente
>=	maior ou igual
<=	menor ou igual

operador	significado
&&	and
	or
!	not
> 0	verdadeiro
== 0	falso

Não temos tipos booleanos em C por padrão!

if... else if... else

```
int x;  
scanf("%d", &x);  
  
if (x > 0) {  
    printf("x eh positivo");  
} else if (x < 0) {  
    printf("x eh negativo");  
} else {  
    printf("x eh igual a 0");  
}
```

```
int x;  
scanf("%d", &x);  
  
if (x > 0)  
    printf("x eh positivo");  
else if (x < 0)  
    printf("x eh negativo");  
else  
    printf("x eh igual a 0");
```

Operador ternário

```
int x;  
scanf("%d", &x);  
  
int positivo = (x > 0) ? 1 : 0;  
  
if (positivo == 1)  
    printf("x eh positivo");  
else  
    printf("x nao eh positivo");
```

Operador ternário

```
int x;  
scanf("%d", &x);  
  
// evite de fazer isso...  
int sinal = (x > 0) ? 1 : (x < 0) ? -1 : 0 ;  
  
if (sinal == 1)  
    printf("x eh positivo");  
else if (sinal == -1)  
    printf("x eh negativo");  
else  
    printf("x eh igual a zero");
```

Switch, break e continue

- `break` : usado para quebrar a execução atual
- `continue` : para pular a próxima iteração (vamos ver junto com repetição)
- o `switch... case` é um dos operadores que usa `break` para controlar o fluxo

Switch

```
int x;  
scanf("%d", &x);  
  
switch(x){  
    case 1: printf("x = 1");  
    case 2: printf("x = 2");  
    default: printf("x nao eh 1 nem 2");  
}
```


Switch

```
int x;  
scanf("%d", &x);  
  
switch(x){  
    case 1: printf("x = 1");  
    case 2: printf("x = 2");  
    default: printf("x nao eh 1 nem 2");  
}
```

No caso do `switch` precisamos usar o `break`

Switch e break

```
scanf("%d", &x);  
switch(x){  
    case 1:  
        printf("x = 1");  
        break;  
    case 2:  
        printf("x = 2");  
        break;  
    default:  
        printf("x nao eh 1 nem 2");  
        break;  
}
```