

ACH2001 - Introdução à Programação

Condicionais e laços

Prof. Flávio Luiz Coutinho
flcoutinho@usp.br

Condicionais (if)

```
if( /* expressão booleana */ ){  
    // bloco de código que executa quando  
    // a expressão booleana é verdadeira.  
}
```

Condicionais (if/else)

```
if( /* expressão booleana */ ){  
    // bloco de código que executa quando  
    // a expressão booleana é verdadeira.  
}  
  
else {  
    // bloco de código que executa quando  
    // a expressão booleana é falsa.  
}
```

Laços (while)

```
while( /* expressão booleana */ ){  
    // Este bloco de código que executa se a  
    // expressão booleana for verdadeira. Após  
    // a execução do bloco a expressão é novamente  
    // avaliada, e se continuar verdadeira, o  
    // bloco de código é novamente executado. Isso  
    // se repete até que a expressão seja falsa.  
}
```

Laços (while): exemplo

```
int main() {  
    int i = 0;  
    while( i < 10 ){  
        printf("i = %d\n", i);  
        i = i + 1;  
    }  
    return 0;  
}
```

Laços (do-while)

```
do {  
    // Este bloco de código que executa ao menos  
    // uma vez, e sua execução se repete enquanto  
    // a expressão booleana for verdadeira.  
} while ( /* expressão booleana */ );
```

Laços (do-while)

```
int main() {  
    int i = 0;  
  
    do {  
        printf("i = %d\n", i);  
        i = i + 1;  
    } while ( i < 10 );  
  
    return 0;  
}
```

Laços (for)

```
for( /*inicialização*/ ; /*condição*/ ; /*incremento*/ ) {  
    // Este bloco de código executa enquanto a condição  
    // (expressão booleana) for verdadeira. A inicialização  
    // é executada antes da primeira execução do bloco  
    // (apenas uma vez), e ao término de cada execução  
    // do bloco o incremento é executado.  
};
```


Laços (for - while equivalente)

```
for( /*inicialização*/ ; /*condição*/ ; /*incremento*/ ) {  
    // bloco de código  
}
```

```
/* inicialização */  
while( /* condição */ ){  
    // bloco de código  
    /* incremento */  
}
```

Laços (for): exemplo

```
int main() {  
    int i;  
    for( i = 0; i < 10; i++ ) {  
        printf("i = %d\n", i);  
    }  
    return 0;  
}
```

Expressões booleanas

Qualquer expressão que possa ser avaliada como **verdadeira** ou **falsa** pode ser usada como expressão booleana em uma condicional ou laço.

Podem ser expressões simples (envolvendo apenas o uso de um operador relacional) ou complexas (envolvendo operadores lógicos que combinam outras expressões booleanas menores)

Na realidade, qualquer valor (literal, ou armazenado em variável) pode ser usado como expressão booleana. Se o valor for zero, é avaliado como falso, caso seja diferente de zero, é avaliado como verdadeiro.

Lembrete: blocos de códigos aninhados

Um bloco de código pode ser composto por outras condicionais e/ou laços. Por ex:

```
int main(){  
    int i, j;  
    for( i = 0; i < 10; i++ ) {  
        for( j = 0; j < 10; j++ ) {  
            printf("(%d, %d) ", i, j); // O bloco de código em azul será  
                                     // executado um total de 10 vezes,  
                                     // com i variando de 0 a 9. Cada  
                                     // execução deste bloco, por sua  
                                     // vez, faz, para cada valor de i,  
                                     // 10 chamadas ao printf sublinhado,  
                                     // com j variando de 0 a 9.  
        }  
        printf("\n");  
    }  
    return 0;  
}
```