

Estruturas de repetição: while

emanoelim@utfpr.edu.br

While

- Repete um conjunto de instruções enquanto uma condição for verdadeira;
- Quando a condição se tornar falsa, as instruções param de ser repetidas (o código irá “sair” do while);

While

- Estrutura do while:

```
while (condição) {  
    instrução 1;  
    instrução 2;  
    instrução 3;  
    ...  
}
```

- Condição: teste lógico que irá indicar quando o loop deve terminar. Chamamos de **condição de parada** do loop;
- É fundamental estabelecer uma condição de parada que seja atingível;
- Assim como em if ou else if, **não vai ponto e vírgula** após a condição;
- Assim como foi visto em outras estruturas de controle, quando existe **mais de uma instrução** para ser executada, **é preciso usar chaves {}**;

While

- Esta estrutura pode ser usada quando:
 - Sabemos exatamente o número vezes um conjunto de instruções precisa ser executado;
 - Quando não sabemos exatamente o número de vez que um conjunto de instruções precisa ser executado, mas podemos estabelecer uma condição de parada para o laço.

While

- Exemplo:

```
// programa que mostra números de 0 até 10 com while
#include <stdio.h>

main() {
    int x = 0;

    while (x <= 10) {
        printf("Numero: %d\n", x);
        x++;
    }
}
```

- Sabemos que o laço precisa ser executado 10 vezes.
- Criamos um contador para guardar o número de vezes em que já passamos pelo laço;
- Devemos lembrar de inicializar o contador;
- O contador é testado pela condição de parada.

While

- Exemplo:

```
#include <stdio.h>

int main(void) {
    int x = 100; // x inicia menor que y
    int y = 1000;

    while(x <= y) { // enquanto x não ultrapassar y
        x += 50;
        y += 10;
    }
    printf("x: %d e y:%d", x, y);

    return 0;
}
```

- Neste caso, queremos atualizar os valores de x e y enquanto o valor de x não ultrapassar o valor de y.
- Não sabemos quantas repetições serão necessárias, mas podemos definir uma condição de parada bastante clara.

While

- O while é bem conveniente para criar programas que permitam ao usuário executar o programa várias vezes;
- Facilita muito para testarmos nossos programas usando diferentes entradas;
- Exemplo:

While

```
...  
int main(void) {  
    char opc = 's';  
    int num;  
    while(opc == 's') {  
        printf("Informe um número: ");  
        scanf("%d", &num);  
        if(num % 2 == 0)  
            printf("O número %d é par.\n", num);  
        else  
            printf("O número %d é ímpar.\n", num);  
        __fpurge(stdin);  
        printf("Continuar (s ou n)? ");  
        scanf("%c", &opc);  
    }  
    return 0;  
}
```