

# Revisão da Linguagem C

Eduardo G. R. Miranda

Agosto 2023

O intuito deste documento é ajudar a relembrar a sintaxe da linguagem C que será utilizada durante as próximas aulas, além de ensinar como compilar e executar os códigos gerados.

1. Exemplo de como declarar a função principal, declarar e ler valores para variáveis e mostrar informações na tela.

Listing 1: Introdução

```
#include <stdio.h>

int main() {
    int idade;
    float altura = 1.75;
    char genero = 'M';

    printf("Digite a sua idade:\n");
    scanf("%d", &idade);

    printf("Idade: %d\nAltura: %.2f\nGenero: %c\n", idade,
        altura, genero);

    return 0;
}
```

2. Exemplo de como utilizar condicionais.

Listing 2: Condicionais

```
#include <stdio.h>

int main() {
    int numero;

    printf("Digite um numero: ");
    scanf("%d", &numero);

    if (numero % 2 == 0) {
        printf("%d e um numero par.\n", numero);
    } else {
        printf("%d e um numero impar.\n", numero);
    }
}
```

```

    }

    return 0;
}

```

3. Exemplo de como utilizar laços de repetição.

Listing 3: Laços de repetição

```

#include <stdio.h>

int main() {
    int i;

    for (i = 1; i <= 10; i++) {
        printf("%d ", i);
    }
    printf("\n");

    return 0;
}

```

4. Exemplo de como utilizar vetores.

Listing 4: Vetores

```

#include <stdio.h>

int main() {
    int numeros[5] = {2, 4, 6, 8, 10};
    int i;

    for (i = 0; i < 5; i++) {
        printf("%d ", numeros[i]);
    }
    printf("\n");

    return 0;
}

```

5. Exemplo de como utilizar matrizes.

Listing 5: Matrizes

```

#include <stdio.h>

int main() {
    int matriz[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int i, j;
}

```

```

    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

## 6. Exemplo de como utilizar structs.

Listing 6: Structs

```

#include <stdio.h>

struct Pessoa {
    char nome[50];
    int idade;
};

int main() {
    struct Pessoa pessoa;

    printf("Digite o nome: ");
    scanf("%s", pessoa.nome);

    printf("Digite a idade: ");
    scanf("%d", &pessoa.idade);

    printf("Nome: %s\nIdade: %d\n", pessoa.nome, pessoa.idade);
    ;

    return 0;
}

```

## 7. Exemplo de como utilizar funções.

Listing 7: Funções

```

#include <stdio.h>

void porValor(int x) {
    x = x * 2;
}

void porReferencia(int *x) {
    *x = *x * 2;
}

int main() {
    int numero = 5;

    porValor(numero);
    printf("Valor apos porValor: %d\n", numero);
}

```

```

    porReferencia(&numero);
    printf("Valor apos porReferencia: %d\n", numero);

    return 0;
}

```

8. Exemplo de como utilizar ponteiros.

Listing 8: Ponteiros

```

#include <stdio.h>

int main() {
    int numero = 10;
    int *ponteiro = &numero;

    printf("Valor de numero: %d\n", numero);
    printf("Endereco de numero: %p\n", &numero);
    printf("Valor apontado pelo ponteiro: %d\n", *ponteiro);
    printf("Endereco armazenado no ponteiro: %p\n", ponteiro);

    return 0;
}

```

A forma de se compilar um programa é:

```
gcc <nomePrograma.c> <nomeExecutavel>
```

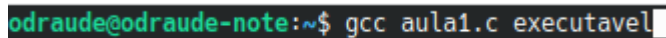


Figure 1: Exemplo de compilação do programa pelo terminal "aula1.c"

A forma de se executar o programa compilado é:

```
.\<nome-executavel>
```

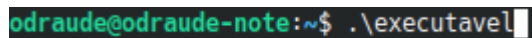


Figure 2: Exemplo de execução do programa "executavel" pelo terminal