

# 1-DEFININDO TIPOS

- STRUCT é uma estrutura de dados. E com ele, criamos nossa propria estrutura.
- Uma das formas de utilizar struct, eh ja no momento da declaração dela, declaramos uma ou mais variaveis que queremos fazer a utilização.

```
✓ struct st_aluno{  
    char matricula[10];  
    char nome[100];  
    char curso[50];  
    int ano_nascimento;  
};
```

```
1  #include <stdio.h>  
2  #include <string.h>  
3  
4  struct st_aluno{  
5      char matricula[10];  
6      char nome[100];  
7      char curso[50];  
8      int ano_nascimento;  
9  };  
10  
11 int main(){  
12  
13     struct st_aluno aluno1;  
14  
15     printf("Informe a matricula do aluno:\n");  
16     fgets(aluno1.matricula,10,stdin);  
17  
18     printf("Informe o nome do aluno:\n");  
19     fgets(aluno1.nome,100,stdin);  
20  
21     printf("Informe o curso do aluno:\n");  
22     fgets(aluno1.curso,50,stdin);  
23  
24     printf("Informe o ano de nascimento do aluno:\n");  
25     scanf("%d",&aluno1.ano_nascimento);  
26  
27     printf("-----Dados do aluno-----\n");  
28     printf("Matricula :%s\n", aluno1.matricula);  
29     printf("Nome do Aluno:%s\n", aluno1.nome);  
30     printf("Curso :%s\n", aluno1.curso);  
31     printf("Ano de Nascimento:%d\n", aluno1.ano_nascimento);  
32  
33     return 0;  
34 }
```

- A partir do struct temos acesso a uma estrutura completa para se trabalhar, a partir de uma variavel fazemos acesso tanto para colocar dados quanto para fazer leitura de dados.

## DECLARANDO VARIABEL NA PROPRIA STRUCT

```
struct st_aluno{  
    char matricula[10];  
    char nome[100];  
    char curso[50];  
    int ano_nascimento;  
}aluno1,aluno2;
```

- Em alguns casos podemos precisar de varios alunos, então podemos declarar um array de alunos, a declaração pode ser feita tanto na struct quanto fora dela.

```
struct st_aluno{
    char matricula[10];
    char nome[100];
    char curso[50];
    int ano_nascimento;
}alunos[5];
```

```
int main(){
    struct st_aluno alunos[5];
```

- Para o programa poder funcionar precisamos aplicar uma repetição.  
- Sempre que temos um scanf e depois eh seguido de uma string, temos o problema de receber o enter, resolvemos isso colocando getchar(); depois do scanf.

```
struct st_aluno{
    char matricula[10];
    char nome[100];
    char curso[50];
    int ano_nascimento;
}alunos[5]; //array 0...4

int main(){

    //struct st_aluno alunos[5];

    for(int i = 0; i < 5; i++){
        printf("Informe a matricula do aluno:\n");
        fgets(alunos[i].matricula,10,stdin);

        printf("Informe o nome do aluno:\n");
        fgets(alunos[i].nome,100,stdin);

        printf("Informe o curso do aluno:\n");
        fgets(alunos[i].curso,50,stdin);

        printf("Informe o ano de nascimento do aluno:\n");
        scanf("%d",&alunos[i].ano_nascimento);
        getchar();
    } //fim_for

    for(int i = 0; i < 5; i++){
        printf("=====Dados do aluno  %d===== \n", (i+1));
        printf("Matricula :%s\n", alunos[i].matricula);
        printf("Nome do Aluno:%s\n", alunos[i].nome);
        printf("Curso :%s\n", alunos[i].curso);
        printf("Ano de Nascimento:%d\n", alunos[i].ano_nascimento);
    }

    return 0;
```

---

- Podemos criar structs de structs  
- Vamos montar uma estrutura para contato, e outra estrutura com um array de 100 contatos.

```

struct st_contato{
    char nome[100];
    int ano_nascimento;
    char telefone[20];
    char email[100];
}; //array 0...4

struct st_agenda{
    struct st_contato contatos[100];
}agenda;

int main(){
    //struct st_aluno alunos[5];

    for(int i = 0; i < 3; i++){
        printf("Informe o nome:\n");
        fgets(agenda.contatos[i].nome,100,stdin);

        printf("Informe o ano de nascimento:\n");
        scanf("%d",&agenda.contatos[i].ano_nascimento);
        getchar();

        printf("Informe o telefone:\n");
        fgets(agenda.contatos[i].telefone,50,stdin);

        printf("Informe O EMAIL:\n");
        fgets(agenda.contatos[i].email,100,stdin);

    } //fim_for

    printf("=====AGENDA DE CONTATO=====\\n");

    for(int i = 0; i < 3; i++){
        printf("=====CONTATO: %d=====\\n",(i+1));
        printf("Nome :%s\\n", agenda.contatos[i].nome);
        printf("Telefone :%s\\n", agenda.contatos[i].telefone);
        printf("Email:%s\\n", agenda.contatos[i].email);
        printf("Ano de Nascimento:%d\\n", agenda.contatos[i].ano_nascimento);
    }

    return 0;
}

```

- o fgets faz com que na hora de salvar o que eh pedido se salve tbm o enter depois da pessoa digitar, por isso o espaço na resposta, para corrigir isso usamos um comando na impressão.

strtok

- Faz parte da biblioteca string e a função ele eh, se no primeiro printf tiver um enter(\\n), removemos ele e imprime sem o \\n