

Prof. Mateus Mendelson
mateus.silva@projecao.br

Estrutura de Dados Heterogêneas **(*Structs* ou Registros)**

1. Structs

- Uma **struct** é uma coleção de campos, em que cada campo pode ser de um tipo de dado diferente. Por isso, são conhecidas como estruturas de dados heterogêneas.

1. Structs

- Uma **struct** é uma coleção de campos, em que cada campo pode ser de um tipo de dado diferente. Por isso, são conhecidas como estruturas de dados heterogêneas.
- Exemplo:

	0	1	2	3	4	5	6	7
NOTAS	10	5	8	4	2	9	3	1

 ➔ Vetor

1. Structs

- Uma **struct** é uma coleção de campos, em que cada campo pode ser de um tipo de dado diferente. Por isso, são conhecidas como estruturas de dados heterogêneas.
- Exemplo:

	0	1	2	3	4	5	6	7
NOTAS	10	5	8	4	2	9	3	1

 → Vetor

	Codigo	Nome	Salario
FUNCIONARIO	2234	Alexandre	1234.56

 → Struct

1. Structs

- Uma **struct** é uma coleção de campos, em que cada campo pode ser de um tipo de dado diferente. Por isso, são conhecidas como estruturas de dados heterogêneas.
- Exemplo:

Dados_de_Funcionario					
Codigo:	9182	Nome:	Joseph Climber	Sexo:	Masculino
Endereco:	Universidade de Brasília				
Cargo:	Professor	Salario:	R\$ 455,46		

1. Structs

Dados_de_Funcionario					
Codigo:	9182	Nome:	Joseph Climber	Sexo:	Masculino
Endereco:	Universidade de Brasília				
Cargo:	Professor	Salario:	R\$ 455,46		

- As variáveis Nome, Sexo, Endereço e Cargo são **strings**, Codigo é um **int** e Salario é um **float**.
- Essas variáveis e seus tipos, criam um novo tipo de dado: **Dados_de_Funcionario**.
- Da mesma forma que existem os tipos vetor de char, int e float, agora existe o tipo Dados_de_Funcionario.
- As variáveis que estão dentro da struct são chamadas de **membros da estrutura** (struct).

1. Structs

- Declaração de uma **struct** em C:

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct Dados_De_Funcionario{
    int codigo;
    float salario;
    char nome[50], sexo[10];
    char endereco[50], cargo[50];
};
```

```
int main() {

    struct Dados_De_Funcionario funcionario;

    system("PAUSE");
    return 0
}
```


1. Structs

- Acessando membros de uma ***struct***:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
struct Dados_De_Funcionario{
    int codigo;
    float salario;
    char nome[50], sexo[10];
    char endereco[50], cargo[50];
};
```

1. Structs

- Acessando membros de uma **struct**:

```
int main() {  
  
    struct Dados_De_Funcionario funcionario;  
  
    funcionario.codigo = 2345;  
    funcionario.salario = 123.4;  
    strcpy(funcionario.nome, "Joseph Climber");  
    strcpy(funcionario.sexo, "Masculino");  
    strcpy(funcionario.endereco, "UnB");  
    strcpy(funcionario.cargo, "Professor");  
  
    printf("%d \n", funcionario.codigo);  
    printf("%f \n", funcionario.salario);  
  
    system("PAUSE");  
    return 0;  
}
```

1. Structs

- Acessando membros de uma ***struct***:

```
int main() {  
  
    struct Dados_De_Funcionario funcionario;  
  
    printf("Digite o codigo do funcionario:");  
    scanf("%d", &funcionario.codigo);  
    printf("Digite o salario do funcionario:");  
    scanf("%f", &funcionario.salario);  
    getchar();  
    printf("Digite o nome:");  
    gets(funcionario.nome);  
    printf("Digite o sexo:");  
    gets(funcionario.sexo);  
    printf("Digite o endereco:");  
    gets(funcionario.endereco);  
    printf("Digite o cargo:");  
    gets(funcionario.cargo);
```

1. Structs

- Acessando membros de uma **struct**:

```
printf("\\n\\n===== \\n");  
printf("Dados digitados \\n");  
printf("===== \\n\\n");  
printf("Codigo: %d \\n", funcionario.codigo);  
printf("Salario: %f \\n", funcionario.salario);  
printf("Nome:");  
puts(funcionario.nome);  
printf("Sexo:");  
puts(funcionario.sexo);  
printf("Endereco:");  
puts(funcionario.endereco);  
printf("Cargo:");  
puts(funcionario.cargo);  
  
system("PAUSE");  
return 0;  
}
```

1. Structs

- Declarando structs utilizando **typedef**:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
typedef struct Dados_De_Funcionario {

    int codigo;
    float salario;
    char nome[50], sexo[10];
    char endereco[50], cargo[50];

}dfunc;
```

1. Structs

- Declarando structs utilizando **typedef**:

```
int main() {  
  
    dfunc funcionario;  
  
        funcionario.codigo = 2345;  
        funcionario.salario = 123.4;  
    strcpy(funcionario.nome, "Joseph Climber");  
    strcpy(funcionario.sexo, "Masculino");  
    strcpy(funcionario.endereco, "UnB");  
    strcpy(funcionario.sexo, "Professor");  
  
    printf("%d \n", funcionario.codigo);  
    printf("%f \n", funcionario.salario);  
  
    system("PAUSE");  
    return 0;  
}
```

1. Structs

- Se precisar apenas de uma única variável:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
int main() {
```

```
struct {
```

```
    int codigo;
```

```
    float salario;
```

```
    char nome[50], sexo[10];
```

```
    char endereco[50], cargo[50];
```

```
}funcionario;
```

1. Structs

- Se precisar apenas de uma única variável:

```
funcionario.codigo = 2345;
funcionario.salario = 123.4;
strcpy(funcionario.nome, "Joseph Climber");
strcpy(funcionario.sexo, "Masculino");
strcpy(funcionario.endereco, "UnB");
strcpy(funcionario.sexo, "Professor");

printf("%d \n", funcionario.codigo);
printf("%f \n", funcionario.salario);

system("PAUSE");
return 0;
}
```


1. Structs

- Vetor de ***struct***:

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct teste{
    int a;
    float b;
};
```

1. Structs

- Vetor de ***struct***:

```
int main() {  
  
    struct teste x[2];  
  
    x[0].a = 10;  
    x[0].b = 11.5;  
    x[1].a = 12;  
    x[1].b = 13.5;  
  
    printf("%d \n", x[0].a);  
    printf("%.2f \n", x[0].b);  
    printf("%d \n", x[1].a);  
    printf("%.2f \n", x[1].b);  
  
    system("PAUSE");  
    return 0;  
}
```