struct aula {

char conteudo[50000];

1. Como eu retorno dois ou mais valores na minha função?
2. Se eu quiser criar uma pokedex? Como eu represento um pokemon?

structs

1. Como eu retorno dois ou mais valores na minha função?
2. Se eu quiser criar uma pokedex? Como eu represento um pokemon?

O que são?
Onde elas vivem?
O que comem?

O que são?

- Uma composição de dados
- Estrutura para armazenamento de mais de um tipo de dado

O que são?

- Uma composição de dados
- Estrutura para armazenamento de mais de um tipo de dado

```
int numero;
char nome[100];
char tipo[100];
char descricao[500];
```

O que são?

- Uma composição de dados
- Estrutura para armazenamento de mais de um tipo de dado

```
pokemon {
    int numero;
    char nome[100];
    char tipo[100];
    char descricao[500];
};
```

O que são?

- Uma composição de dados
- Estrutura para armazenamento de mais de um tipo de dado

```
struct pokemon {
    int numero;
    char nome[100];
    char tipo[100];
    char descricao[500];
};
```

Syntax

```
struct NOME {
   // Campos
   tipo nome_do_campo1;
   tipo nome_do_campo2;
};
```

Syntax

```
struct Pessoa {
   // Campos
   tipo nome_do_campo1;
   tipo nome_do_campo2;
};
```

Syntax

```
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};
```

Syntax

```
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};
```

Linha do campo termina com ponto-e-vírgula

Após o fechamento das chaves tem ponto-e-vírgula

Syntax

```
struct Pessoa {
    char nome[100];
    int idade;
    char sexo;
};
```

```
struct pokemon {
    int numero;
    char nome[100];
    char tipo[100];
    char descricao[500];
};
```

Como eu uso?

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    return 0;
}
```

Declaração antes da *main*

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    return 0;
}
```

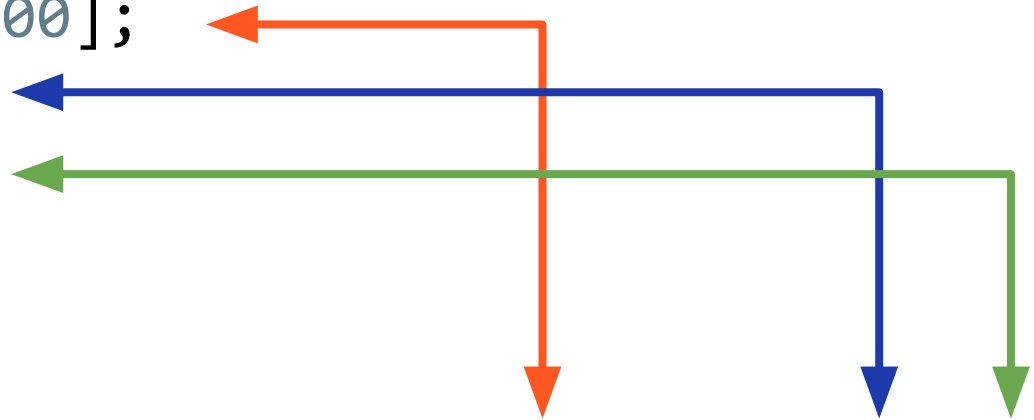
Criando **Variável** do tipo *struct Pessoa*

Preenchendo os campos da struct (Maneira 1)

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    return 0;
}
```

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno = {“Matheus”, 32, ‘M’};
    return 0;
}
```

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno = {“Matheus”, 32, ‘M’};
    return 0;
}
```



Preenchendo os campos da struct (Maneira 2)

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    return 0;
}
```

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    aluno.idade = 32;
    aluno.sexo = ‘M’;
    return 0;
}
```

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};
int main() {
    struct Pessoa aluno;
    for (int i = 0; i < tamanho_string; i++) {
        aluno.nome[i] = string[i];
    }
    return 0;
}
```

A string é mais complicada de inicializar

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    scanf(“%d”, &aluno.idade);
    return 0;
}
```

Utilizando no **scanf**

```c
struct Pessoa {
    // Campos
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno;
    aluno.sexo = ‘F’;
    printf(“%c\n”, aluno.sexo);
    return 0;
}
```

Utilizando no **printf**

```c
int main() {
    struct Pessoa aluno;
    aluno.idade = 32;

    if (aluno.idade < 18) {
        printf(“Di menor!\n”);
    }
    else {
        printf(“Adultão!\n”);
    }
    return 0;
}
```

Utilizando no **resto do código**

Comparar duas structs? Como?

```
struct Pessoa {
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno  = {“Matheus”, 32, ‘M’};
    struct Pessoa aluno2 = {“Matheus”, 32, ‘M’};

    if (aluno == aluno2) {
        ...
```

```
struct Pessoa {
    char n
    int id
    char s
};

int main(
    struct                                    'M'};
    struct                                    'M'};

    if (al
        ...
```

```c
struct Pessoa {
    char nome[100];
    int idade;
    char sexo;
};

int main() {
    struct Pessoa aluno  = {“Matheus”, 32, ‘M’};
    struct Pessoa aluno2 = {“Matheus”, 32, ‘M’};

    if (aluno.idade == aluno2.idade
        && aluno.sexo == aluno2.sexo) {
        ...
```

```
struct Pessoa {
    char n
    int id
    char s
};

int main(
    struct                                          'M'};
    struct                                          'M'};

    if (al
        &&
    ...
```

Que tal uma função?

Faça uma função que compare duas structs de pessoa.