

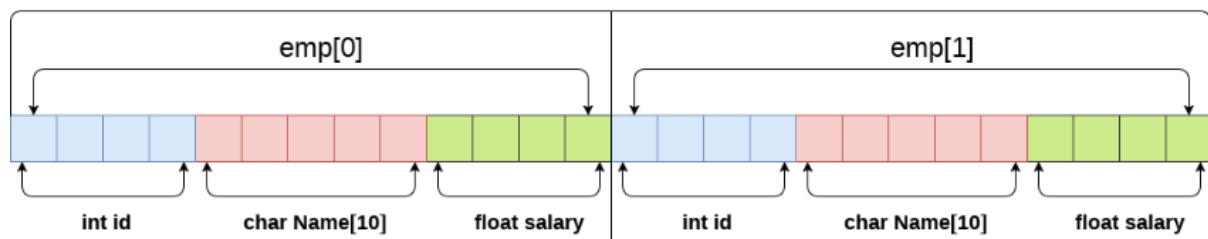
## Structures

It is a collection of different data types and we can use this collection as a single type.

```
struct employee
{
    char _name[10];
    int id[5];
    float salary;
};
```

### Array of structures

It is the collection of multiple structures variables where each variable contains information about different entities. It is used to store information about multiple entities of different data types. The array of structures is also known as the collection of structures.



### Pointers to structures

It is a pointer that points to a structure variable.

```
struct employee e1;
struct employee *ptr;
ptr=&e1;
```

### Arrow operator

It is used with a pointer variable pointing to a structure or union.

`(*ptr).id = ptr→id`

## Passing structure to function

//function prototype

```
void func(struct employee e1);
```

## typedef keyword

It is a keyword that is used to provide existing data types with a new name. It is used to redefine the name of already existing data types.

We can say it is used to create alias for data types.

We all have nicknames in our house. So, the use of typedef is to create nickname for datatypes.

```
//typedef existing_name alias_name;
```

```
typedef struct employee
```

```
{
```

```
    char name[100]
```

```
    int id;
```

```
    float salary;
```

```
} emp
```

```
emp employee1;
```