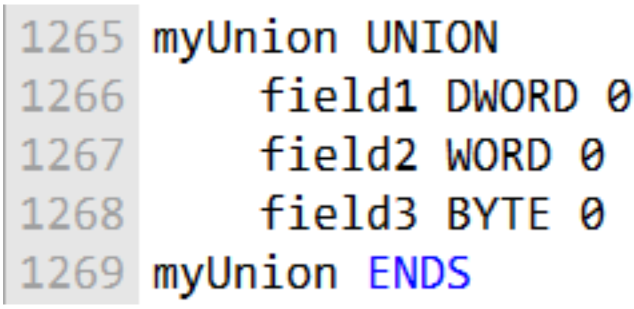
Unions in ASM

In assembly language, unions are used to group fields together at the same offset.

The size of a union is determined by its largest field.

To declare a union, you use the **UNION and ENDS directives**, like this:



```
myUnion UNION
    field1 DWORD 0
    field2 WORD 0
    field3 BYTE 0
myUnion ENDS
```

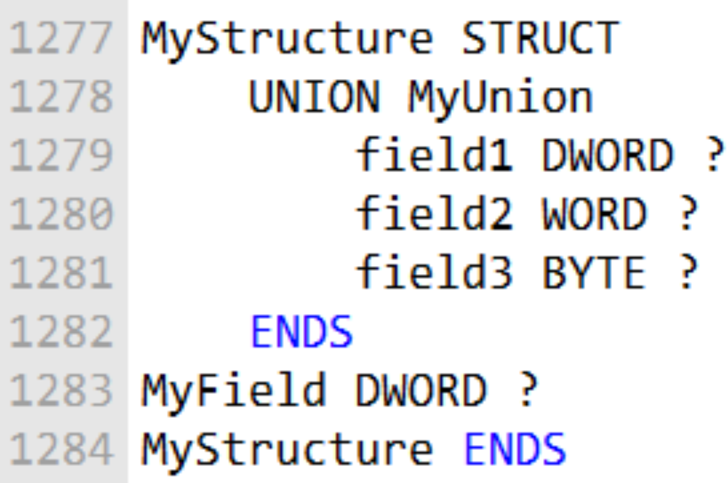
You don't have to indent.

Fields in a union follow the same rules as structures, but each field can have only one initializer.

Nesting Unions in Structures

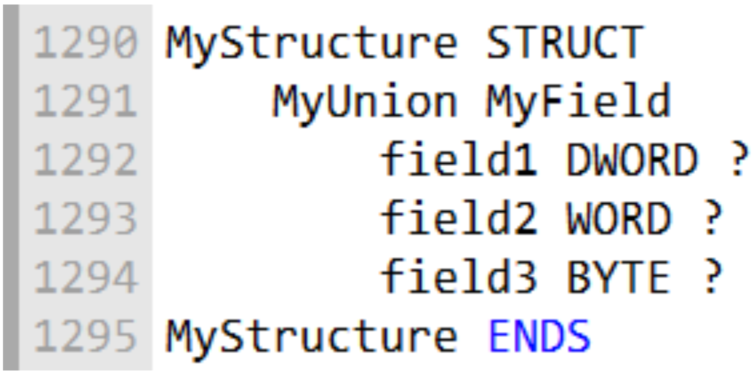
You can nest a union inside a structure in two ways:

Declare the union inside the structure:



```
MyStructure STRUCT
    UNION MyUnion
        field1 DWORD ?
        field2 WORD ?
        field3 BYTE ?
    ENDS
MyField DWORD ?
MyStructure ENDS
```

Use the union's name in a declaration inside the structure:

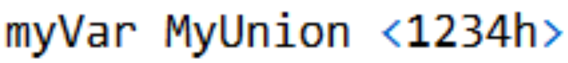


```
MyStructure STRUCT
    MyUnion MyField
        field1 DWORD ?
        field2 WORD ?
        field3 BYTE ?
MyStructure ENDS
```

Declaring and Using Union Variables:

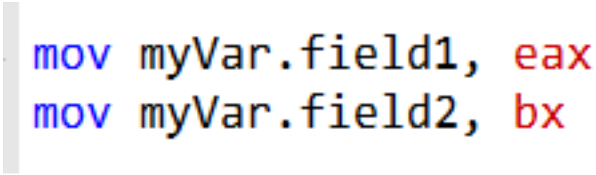
Declaring a union variable is similar to declaring a structure variable.

However, you can only have one initializer:



```
myVar MyUnion <1234h>
```

To use a specific field of a union variable, you supply the field's name. For example:



```
mov myVar.field1, eax
mov myVar.field2, bx
```

Unions Can Contain Structures

Unions can also contain structures.

For instance, you can have a union that can hold different types of structures based on the situation. Example - KEY_EVENT_RECORD:

In the Windows API, there's a structure called KEY_EVENT_RECORD that contains a union named uChar. It's used to handle keyboard input events.

Depending on the situation, it can contain either a Unicode character or an ASCII character.

The union lets you use the appropriate character based on the event type.

Unions and structures are helpful for organizing data in assembly language programs.

Unions allow you to store data of different sizes in the same memory space, which can be useful in various situations.