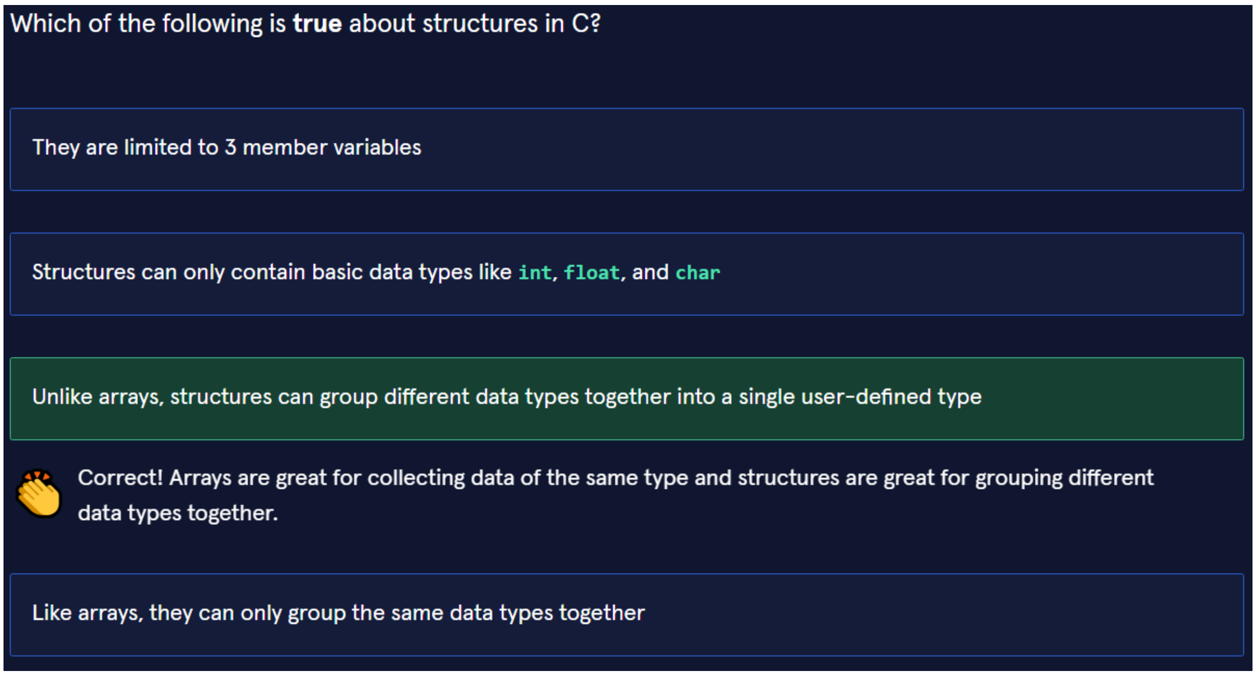
QUIZ



Which of the following is **true** about structures in C?

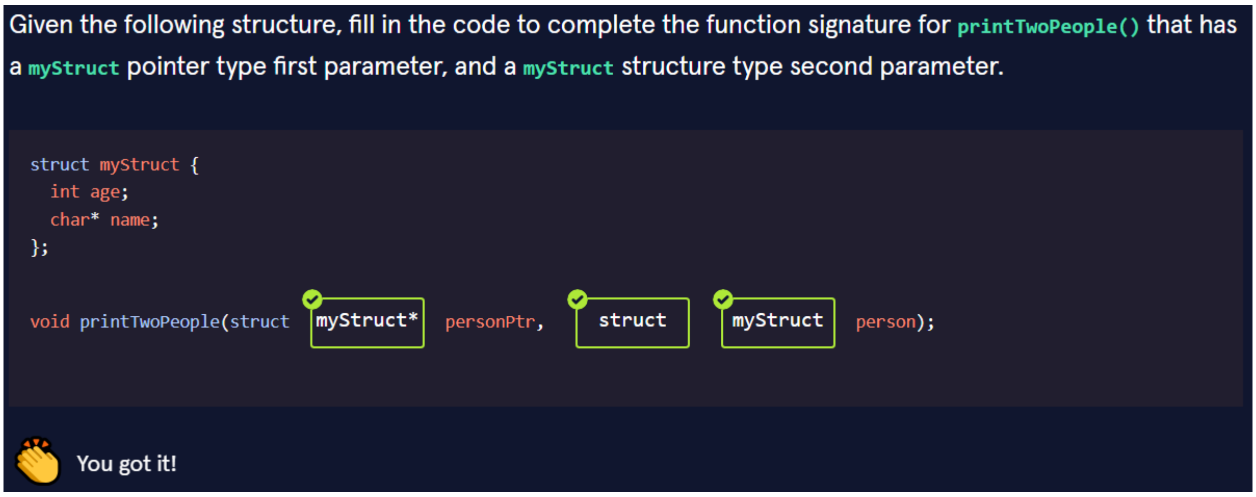
They are limited to 3 member variables

Structures can only contain basic data types like `int`, `float`, and `char`

Unlike arrays, structures can group different data types together into a single user-defined type

Correct! Arrays are great for collecting data of the same type and structures are great for grouping different data types together.

Like arrays, they can only group the same data types together

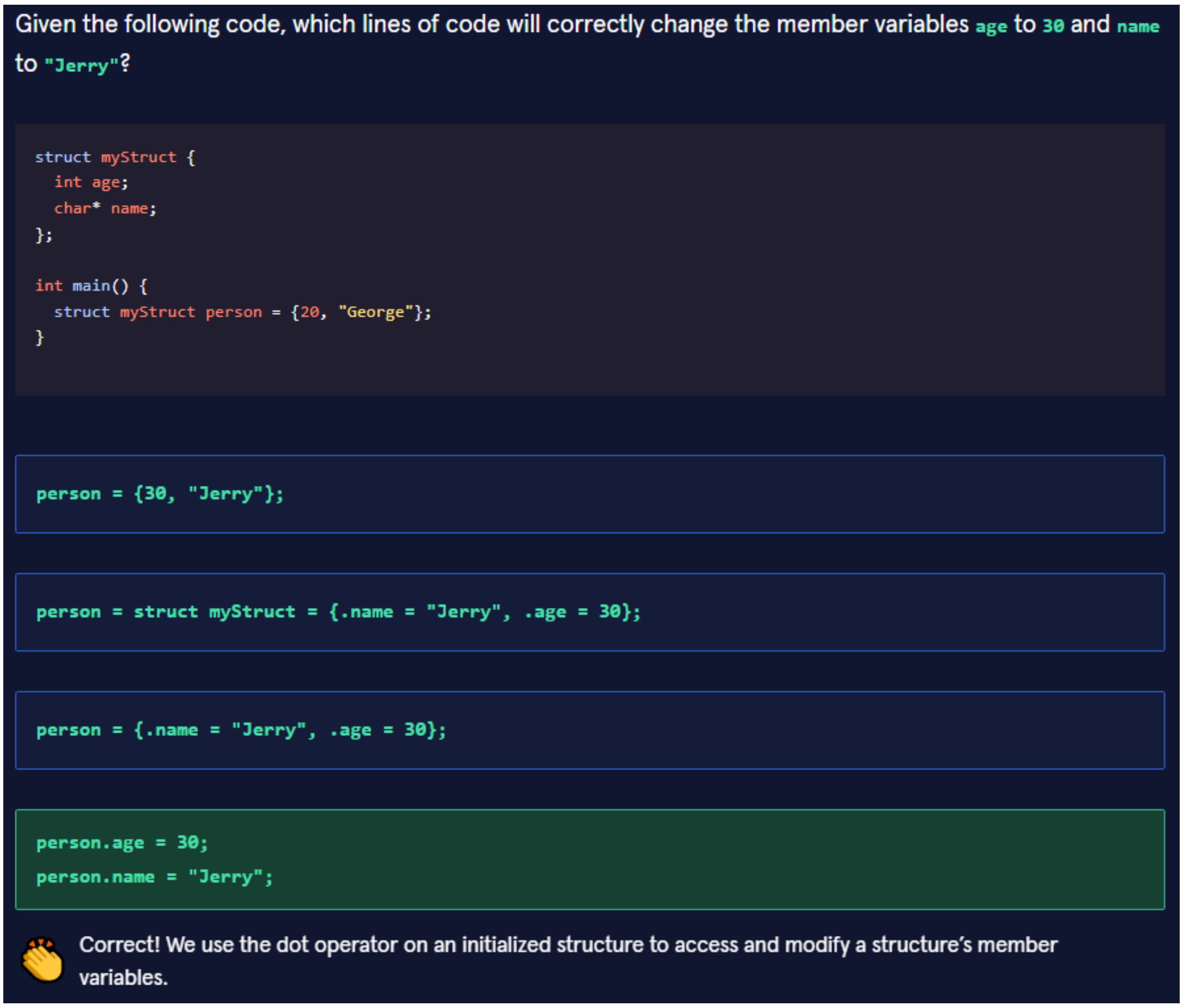


Given the following structure, fill in the code to complete the function signature for `printTwoPeople()` that has a `myStruct` pointer type first parameter, and a `myStruct` structure type second parameter.

```
struct myStruct {
  int age;
  char* name;
};

void printTwoPeople(struct  myStruct*  personPtr,  struct  myStruct  person);
```

You got it!



Given the following code, which lines of code will correctly change the member variables `age` to `30` and `name` to `"Jerry"`?

```c
struct myStruct {
  int age;
  char* name;
};

int main() {
  struct myStruct person = {20, "George"};
}
```

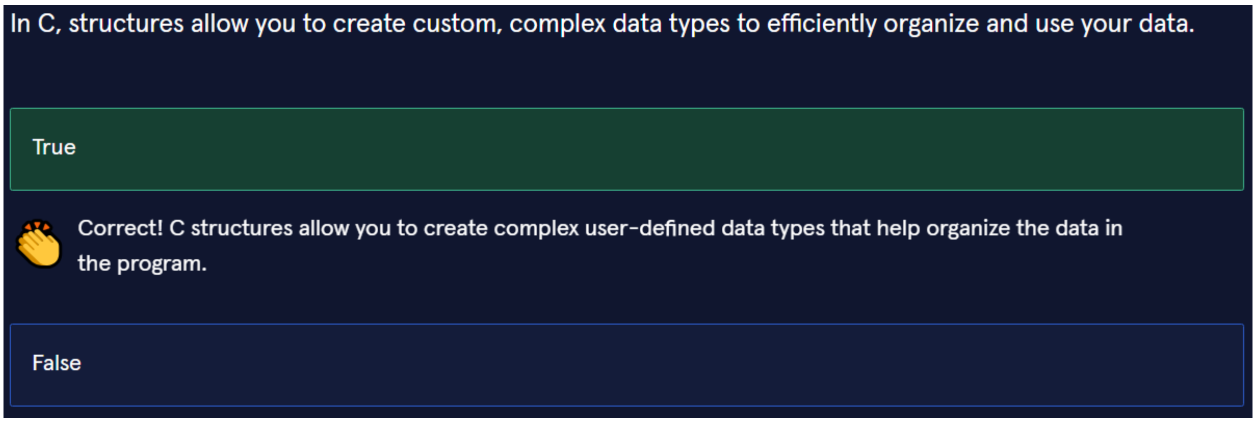
```c
person = {30, "Jerry"};
```

```c
person = struct myStruct = {.name = "Jerry", .age = 30};
```

```c
person = {.name = "Jerry", .age = 30};
```

[✓
```c
person.age = 30;
person.name = "Jerry";
```
]

Correct! We use the dot operator on an initialized structure to access and modify a structure's member variables.

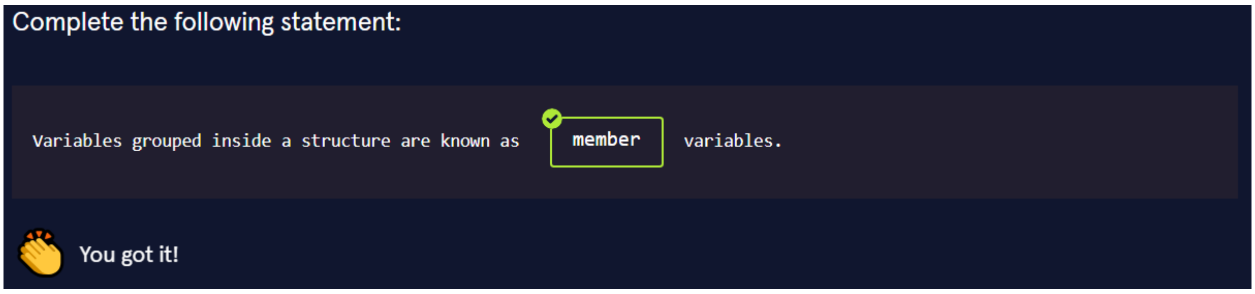


In C, structures allow you to create custom, complex data types to efficiently organize and use your data.

[✓ True]

Correct! C structures allow you to create complex user-defined data types that help organize the data in the program.

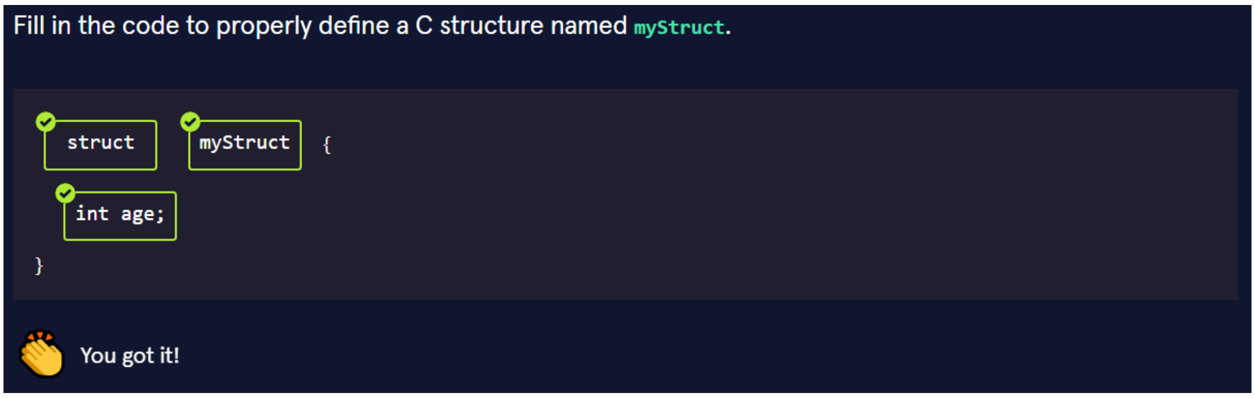
False



Complete the following statement:

`Variables grouped inside a structure are known as` ✓ `member` `variables.`

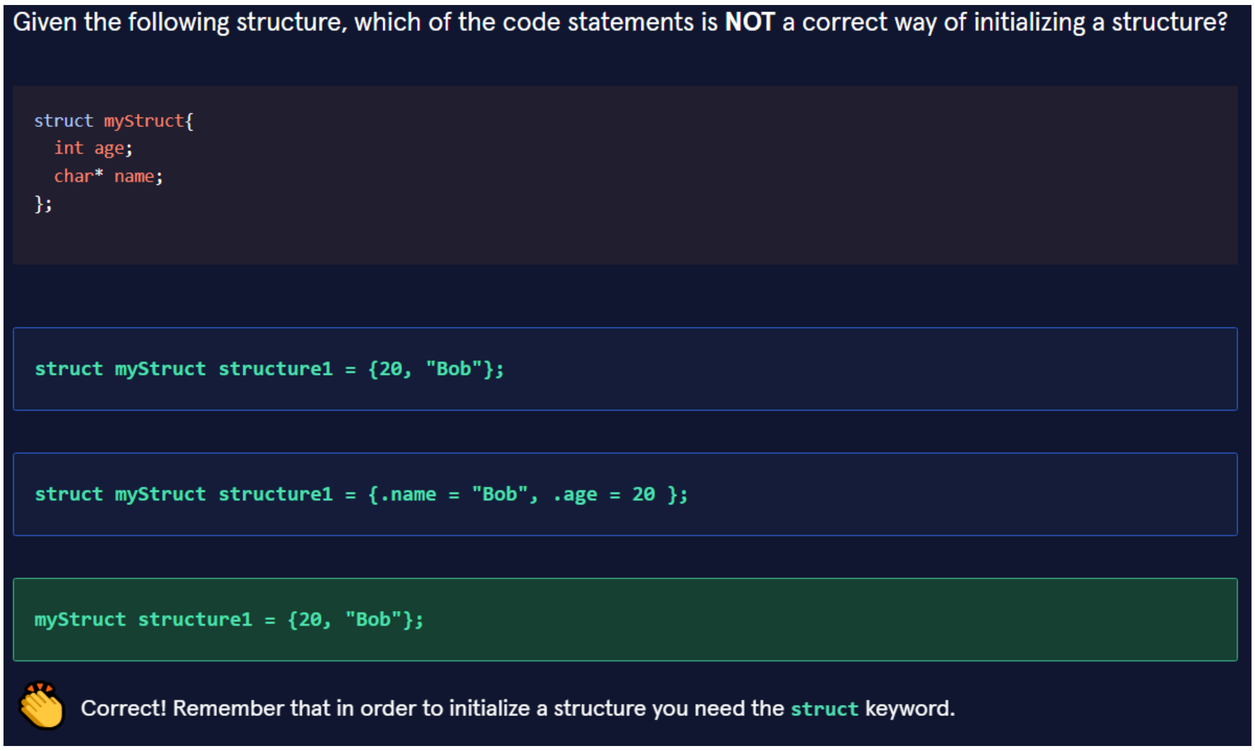
You got it!



Fill in the code to properly define a C structure named `myStruct`.

```
struct myStruct {
  int age;
}
```

You got it!



Given the following structure, which of the code statements is **NOT** a correct way of initializing a structure?

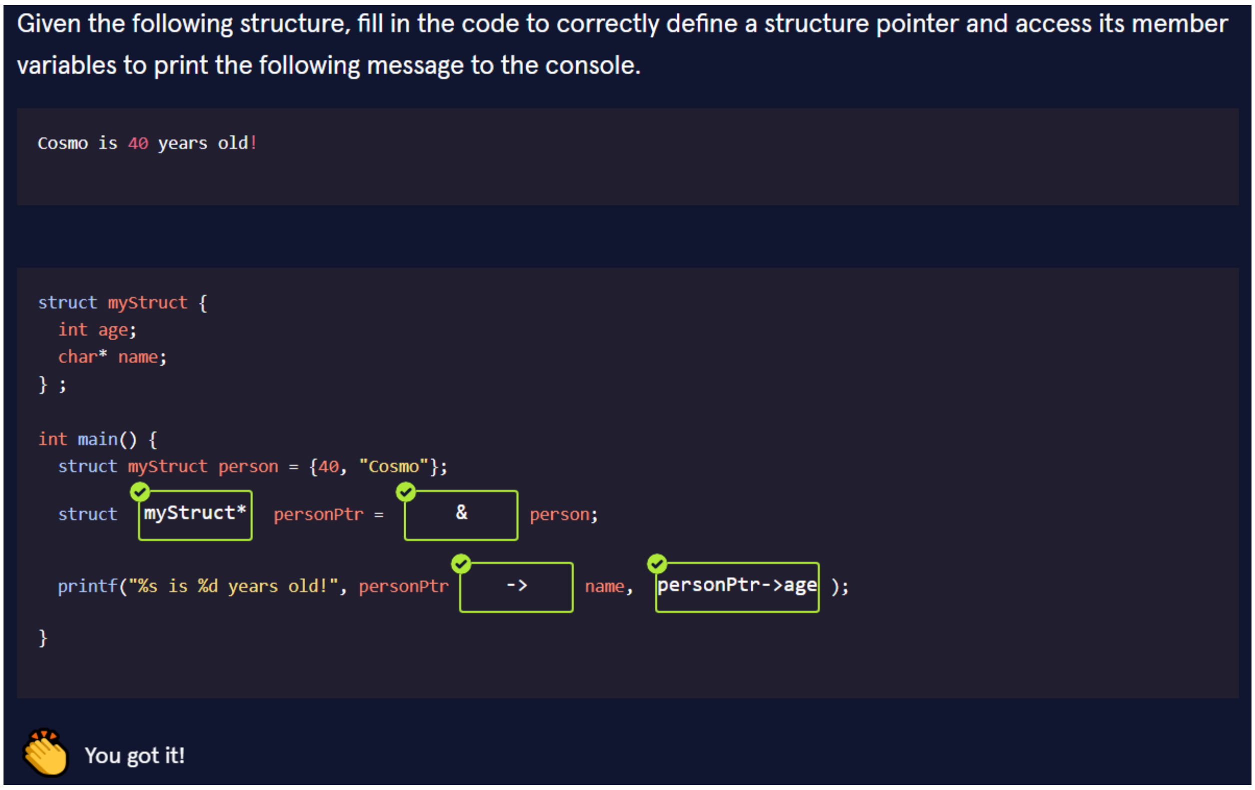
```
struct myStruct{
  int age;
  char* name;
};
```

`struct myStruct structure1 = {20, "Bob"};`

`struct myStruct structure1 = {.name = "Bob", .age = 20 };`

[✓ `myStruct structure1 = {20, "Bob"};`]

Correct! Remember that in order to initialize a structure you need the `struct` keyword.


Given the following structure, fill in the code to correctly define a structure pointer and access its member variables to print the following message to the console.

```
Cosmo is 40 years old!
```

```
struct myStruct {
  int age;
  char* name;
} ;

int main() {
  struct myStruct person = {40, "Cosmo"};
  struct myStruct* personPtr = & person;

  printf("%s is %d years old!", personPtr -> name, personPtr->age );

}
```

You got it!