

6. 구조체



01



구조체

02



멤버 접근

03



자료구조



구조체

: 사용자가 C언어의 기본 타입을 가지고 새롭게 정의할 수 있는 구조화된 데이터

→ 사용자 정의 타입



구조체의 할당과 선언

```
1  #include <stdio.h>
2
3  struct Student {
4      int id;
5      char grade;
6      char * name;
7      char * gender;
8  };
9
10 int main() {
11     struct Student eunho = {2018, 'B', "Kang Eunho", "male"};
12     return 0;
13 }
```

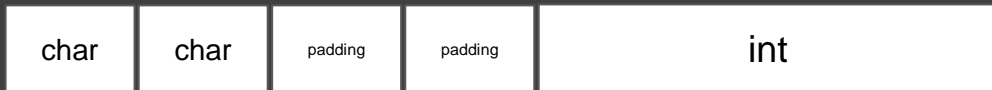
```
1  #include <stdio.h>
2
3  int main() {
4      struct Student {
5          int id;
6          char grade;
7          char * name;
8          char * gender;
9      }eunho = {2018, 'B', "Kang Eunho", "male"};
10     return 0;
11 }
```



실제로 일어나는 일

컴파일러는 구조체를 구성하는 멤버들을 가장 크기가 큰 멤버 자료형의 배수가 되도록 정렬한다.

→ 구조체 선언 방식에 따라 공간 효율에 차이가 생김



값으로 참조, 주소로 참조

```
1  #include <stdio.h>
2
3  struct Student {
4      int id;
5      char grade;
6      char * name;
7      char * gender;
8  };
9
10 int main() {
11     struct Student eunho = {2018, 'B', "Kang Eunho", "male"};
12
13     printf("%s's grade is %c\n", eunho.name, eunho.grade);
14
15     eunho.grade = 'A';
16
17     printf("%s's grade is %c\n", eunho.name, eunho.grade);
18
19     return 0;
20 }
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Student {
5      int id;
6      char grade;
7      char * name;
8      char * gender;
9  };
10
11 int main() {
12     struct Student * eunho = malloc(sizeof(struct Student));
13
14     eunho -> id = 2018;
15     eunho -> grade = 'B';
16     eunho -> name = "Eunho Kang";
17     eunho -> gender = "male";
18
19     printf("%s's grade is %c\n", (*eunho).name, (*eunho).grade);
20
21     eunho -> grade = 'A';
22
23     printf("%s's grade is %c\n", (*eunho).name, (*eunho).grade);
24
25     free(eunho);
26     return 0;
27 }
```

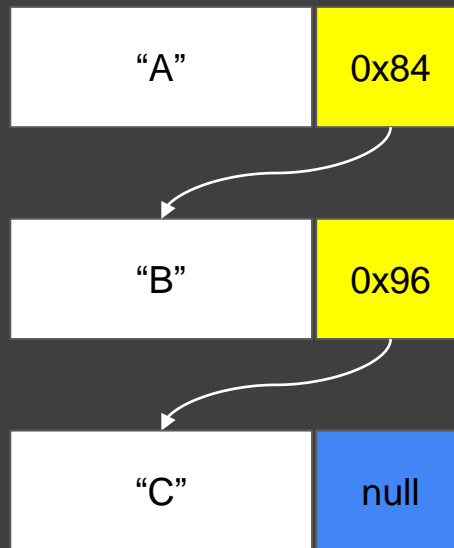
```
Kang Eunho's grade is B
Kang Eunho's grade is A
```



자기구조 참조체

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     char * item;
6     struct Node * next;
7 };
8
9 int main() {
10     struct Node * a, * b, * c;
11     a = malloc(sizeof(struct Node));
12     b = malloc(sizeof(struct Node));
13     c = malloc(sizeof(struct Node));
14
15     a -> item = "A";
16     b -> item = "B";
17     c -> item = "C";
18     a -> next = b;
19     b -> next = c;
20
21     printf("%s -> %s -> %s\n", a -> item, a -> next -> item, a -> next -> next -> item);
22     return 0;
23 }
```

A -> B -> C



자료구조

: 컴퓨터 과학에서 효율적인 접근 및 수정을 가능케 하는 자료의 조직, 관리, 저장을 의미한다.

→ 데이터 값의 모임, 또 데이터 간의 관계, 그리고 데이터에 적용할 수 있는 함수나 명령을 의미



자료구조를 쓰는 이유

“...이 질문에 대한 대답은 크게 두 가지로 나눌 수 있습니다. 바로 추상화와 최적화입니다. 사실 (거의) 모든 자료 구조는 이 두 가지 목적을 이루기 위해 고안된 것들입니다.”

-구종만, 알고리즘 문제해결전략 中

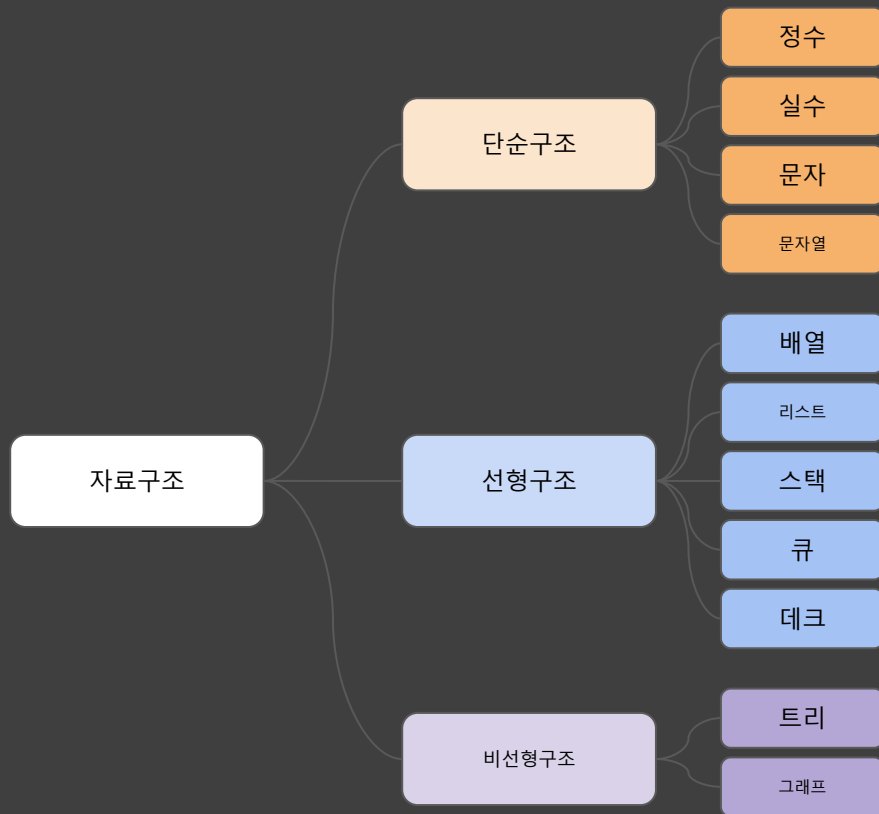


자료구조를 쓰는 이유

- 코드 상에서 자료를 좀 더 알아보기 쉽게 하기 위해
- 프로그래머가 서로 알려진 프로그래밍 지식으로 소통하기 위해
- 더 빠르고 메모리를 덜 쓰는 프로그램을 만들기 위해



자료구조의 종류



THANK YOU!

