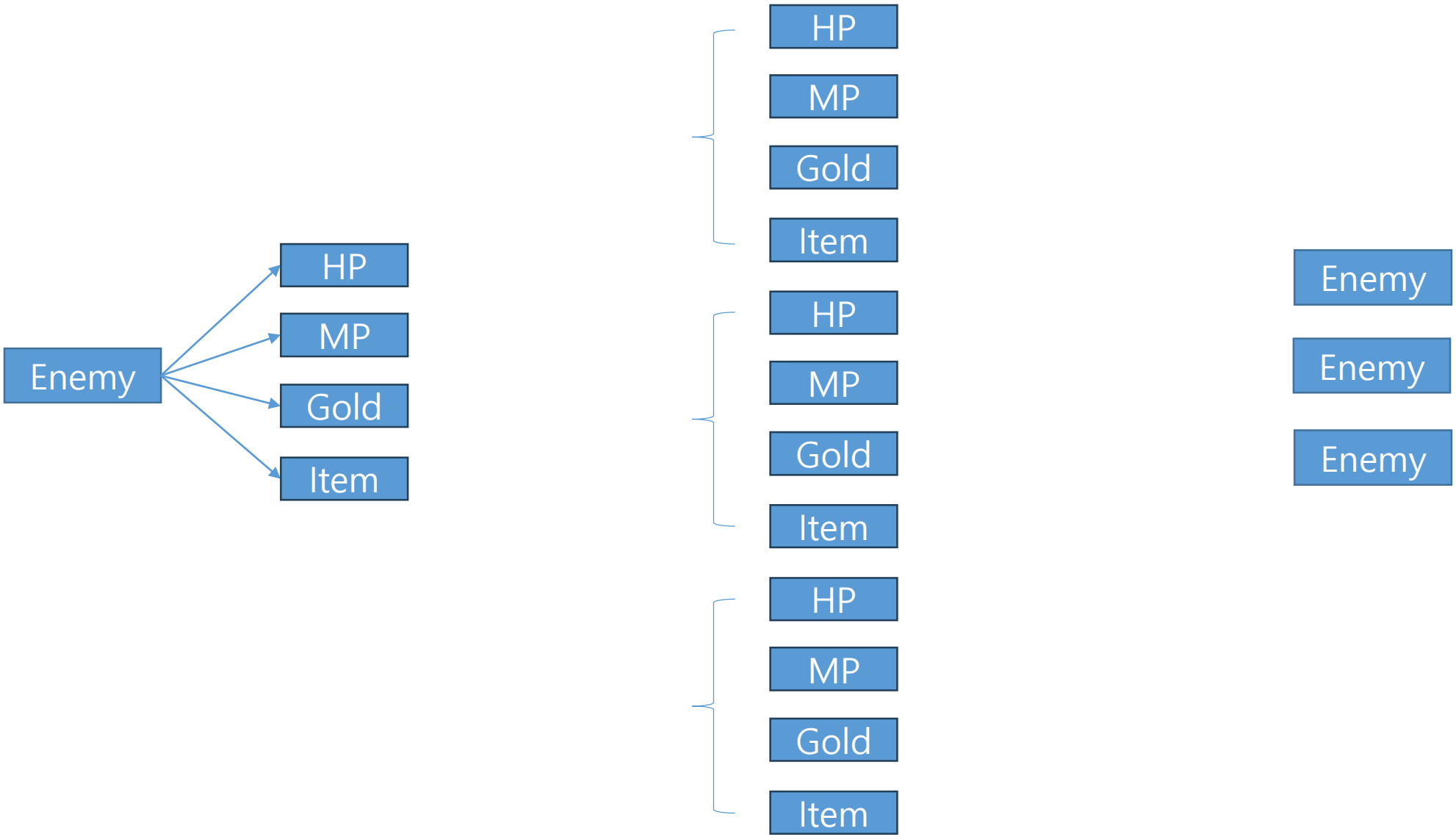
Struct & Union

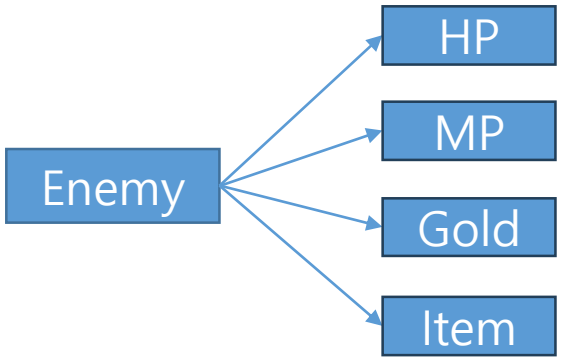
Jinwoo Choi

Struct(구조체) 란?


Enemy
HP
MP
Gold
Item
HP
MP
Gold
Item
HP
MP
Gold
Item
HP
MP
Gold
Item
Enemy
Enemy
Enemy

Struct

```
struct structName
{
    type variableName0;
    type variableName1;
    type variableName2;
        .
        .
        .
};
```



Member Variables

```
enum ItemType
{
    NONE,
    POTION,
    SWORD,
    SHIELD
};
struct Enemy
{
    int HP;
    int MP;
    int Gold;
    enum ItemType item;
};
```

Struct

```
enum ItemType
{
    NONE,
    POTION,
    SWORD,
    SHIELD
};
struct Enemy
{
    int HP;
    int MP;
    int Gold;
    enum ItemType item;
};
```

To access member variable:
structVariableName.memberVariableName

```
Enemy goblin;
goblin.HP = 100;
goblin.MP = 20;
goblin.Gold = 50;
goblin.item = POTION;
```

Struct

```
struct Enemy goblin =
        {100, 20, 50, POTION};
```

```
struct Enemy enemies[3] =
  {
      {100, 20, 50, POTION},
      {150, 30, 70, SWORD},
      {200, 40, 100, SHIELD}
  };
```

```
jinwoo@eulb:~$ ./a.out
Enemy 0 -> HP: 100, MP: 20, Gold: 50, Item: 1
Enemy 1 -> HP: 150, MP: 30, Gold: 70, Item: 2
Enemy 2 -> HP: 200, MP: 40, Gold: 100, Item: 3
```

```c
#include <stdio.h>

enum ItemType
{
    NONE,
    POTION,
    SWORD,
    SHIELD
};

struct Enemy
{
    int HP;
    int MP;
    int Gold;
    enum ItemType item;
};

int main(void)
{
    struct Enemy enemies[3] =
    {
        {100, 20, 50, POTION},
        {150, 30, 70, SWORD},
        {200, 40, 100, SHIELD}
    };

    for (int i = 0; i < 3; i++)
    {
        printf("Enemy %d -> HP: %d, MP: %d, Gold: %d, Item: %d\n",
               i, enemies[i].HP, enemies[i].MP, enemies[i].Gold, enemies[i].item);
    }

    return 0;
}
```

typedef

```
typedef enum enumName
{
    userDefinedName0,
    userDefinedName1
} newEnumName;
```

```
struct structName
{
    type memberVariable0;
    type memberVariable1;
};
typedef struct structName newStructName;
```

```
jinwoo@eulb:~$ ./a.out
Enemy 0 -> HP: 100, MP: 20, Gold: 50, Item: 1
Enemy 1 -> HP: 150, MP: 30, Gold: 70, Item: 2
Enemy 2 -> HP: 200, MP: 40, Gold: 100, Item: 3
```

```c
#include <stdio.h>

int main(void)
{

    enum ItemType
    {
        NONE,
        POTION,
        SWORD,
        SHIELD
    };
    typedef enum ItemType ITEMTYPE;

    typedef struct Enemy
    {
        int HP;
        int MP;
        int Gold;
        ITEMTYPE item;
    }ENEMY;


    ENEMY enemies[3] =
    {
        {100, 20, 50, POTION},
        {150, 30, 70, SWORD},
        {200, 40, 100, SHIELD}
    };

    for(int i = 0; i < 3; i++)
    {
        printf("Enemy %d -> HP: %d, MP: %d, Gold: %d, Item: %d\n",
                i, enemies[i].HP, enemies[i].MP, enemies[i].Gold, enemies[i].item);
    }
    return 0;
}
```

Struct with Pointer

structName *p_struct = &struct;

(*p_struct).memberVariable;
p_struct->memberVariable;

// -> is the general way

```c
#include <stdio.h>

int main(void)
{

    enum ItemType
    {
        NONE,
        POTION,
        SWORD,
        SHIELD
    };
    typedef enum ItemType ITEMTYPE;

    typedef struct Enemy
    {
        int HP;
        int MP;
        int Gold;
        ITEMTYPE item;
    }ENEMY;


    ENEMY enemy = {100, 20, 50, POTION};
    ENEMY *p_enemy = &enemy;

    printf("Enemy -> HP: %d, MP: %d, Gold: %d, Item: %d\n",
            (*p_enemy).HP, p_enemy->MP, enemy.Gold, p_enemy->item);

    return 0;
}
```

Union

```
union UnionName
{
    type variable0;
    type variable1;
    type variable2;
};
```

```
typedef union UnionName
{
    type variable0;
    type variable1;
    type variable2;
}newUnionName;
```

```c
#include <stdio.h>

typedef union
{
    int i;
    char bytes[4];
}Data;

int main()
{
    Data d;
    d.i = 0x12345678;  // 16진수 값 저장

    printf("Byte 0: %x\n", d.bytes[0]);
    printf("Byte 1: %x\n", d.bytes[1]);
    printf("Byte 2: %x\n", d.bytes[2]);
    printf("Byte 3: %x\n", d.bytes[3]);

    return 0;
}
```

```
jinwoo@DESKTOP-UEN32NR:~$ ./a.out
Byte 0: 78
Byte 1: 56
Byte 2: 34
Byte 3: 12
```

Address	Memory
0x10	01111000 (0x78)
0x11	01010110 (0x56)
0x12	00110100 (0x34)
0x13	00010010 (0x12)
0x14	
0x15	
0x16	
0x17	

4 byte

LAB – StudentInfo

- Create a file named 'StudentInfo_YourName.c'.
- Define a structure(struct Student)

  to store the following information:

  - **Name (string)**
  - **Age (integer)**
  - **Grade (floating point)**

- Implement the following functions:
  - **void inputStudent(struct Student *s);**
    - this function should prompt the user to enter student details
    - Use a pointer to modify the structure data
  - **void printStudent(const struct Student *s);**
    - This function should print the student details
- In the main() function:
  - Declare a variable of type struct Student
  - Call **inputStudent(&student);** to read student data
  - Call **printStudent(&student);** to display the student data