



# 1) 조건문과 반복문



매 강의 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

## [수업 목표]

- C#에서 조건문과 반복문을 이해하고 활용할 수 있다.
- 다양한 조건문과 반복문을 사용하여 프로그램의 제어 흐름을 변경할 수 있다.

## [목차]

01. 조건문

02. 조건문 심화 실습

03. 반복문

04. 반복문 심화 실습



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

## 01. 조건문



조건문은 주어진 조건식의 결과에 따라 프로그램의 제어 흐름을 변경하는 제어문

### ▼ 1) if 문

- if 문은 조건식의 결과에 따라 실행 여부를 결정하는 조건문입니다.

```
# 실행할 코드가 여러줄인 경우
if (조건식)
{
    // 조건식이 참일 경우 실행할 코드
}

# 실행할 코드가 한줄인 경우
if (조건식)
    // 조건식이 참일 경우 실행할 코드
```

### ▼ [코드스니펫] if문 기초

```
int playerScore = 80;

if (playerScore >= 70)
{
    Console.WriteLine("플레이어의 점수는 70점 이상입니다. 합격입니다!");
}
Console.WriteLine("프로그램이 종료됩니다.");
```

### ▼ 2) else 문

- else 문은 if 문에서 조건식이 거짓일 경우 실행할 코드를 지정하는 조건문입니다.

```
if (조건식)
{
    // 조건식이 참일 경우 실행할 코드
}
else
{
    // 조건식이 거짓일 경우 실행할 코드
}
```

## ▼ [코드스니펫]else문 기초

```
int itemCount = 5;
string itemName = "HP 포션";

if (itemCount > 0)
{
    Console.WriteLine($"보유한 {itemName}의 수량: {itemCount}");
}
else
{
    Console.WriteLine($"보유한 {itemName}이 없습니다.");
}
```

## ▼ 3) else if 문

- else if 문은 if 문의 조건식이 거짓일 때, 새로운 조건식을 사용하여 실행 여부를 결정하는 조건문입니다.
- else 문은 생략이 가능하다.

```
if (조건식1)
{
    // 조건식1이 참일 경우 실행할 코드
}
else if (조건식2)
{
    // 조건식2가 참일 경우 실행할 코드
}
else
{
    // 모든 조건식이 거짓일 경우 실행할 코드
}
```

- 다음은 예시입니다.

```
int playerScore = 100;
string playerRank = "";

if (playerScore >= 90)
{
    playerRank = "Diamond";
}
else if (playerScore >= 80)
{
    playerRank = "Platinum";
}
```

```

}
else if (playerScore >= 70)
{
    playerRank = "Gold";
}
else if (playerScore >= 60)
{
    playerRank = "Silver";
}
else
{
    playerRank = "Bronze";
}

Console.WriteLine("플레이어의 등급은 " + playerRank + "입니다.");

```

#### ▼ 4) 중첩 조건문

- 하나의 조건문 안에 또 다른 조건문이 포함된 형태의 조건문입니다.

```

int itemLevel = 3; // 아이템 레벨
string itemType = "Weapon"; // 아이템 종류

if (itemType == "Weapon")
{
    if (itemLevel == 1)
    {
        // 레벨 1 무기 효과
        Console.WriteLine("공격력이 10 증가했습니다.");
    }
    else if (itemLevel == 2)
    {
        // 레벨 2 무기 효과
        Console.WriteLine("공격력이 20 증가했습니다.");
    }
    else
    {
        // 그 외 무기 레벨
        Console.WriteLine("잘못된 아이템 레벨입니다.");
    }
}
else if (itemType == "Armor")
{
    if (itemLevel == 1)
    {
        // 레벨 1 방어구 효과
        Console.WriteLine("방어력이 10 증가했습니다.");
    }
    else if (itemLevel == 2)
    {
        // 레벨 2 방어구 효과
        Console.WriteLine("방어력이 20 증가했습니다.");
    }
}

```

```

        else
        {
            // 그 외 방어구 레벨
            Console.WriteLine("잘못된 아이템 레벨입니다.");
        }
    }
else
{
    // 그 외 아이템 종류
    Console.WriteLine("잘못된 아이템 종류입니다.");
}
}

```

## ▼ 5) switch 문

- switch문은 변수나 식의 결과에 따라 다른 코드 블록을 실행하는 제어문입니다.
- case문을 사용하여 변수나 식의 결과에 따라 실행할 코드를 지정합니다.

```

switch (변수나 식)
{
    case 값1:
        // 값1이 나온 경우 실행되는 코드
        break;
    case 값2:
        // 값2가 나온 경우 실행되는 코드
        break;
    // ...
    default:
        // 모든 case문에 해당하지 않는 경우 실행되는 코드
        break;
}

```

## ▼ [코드스니펫] switch문 기초

```

Console.WriteLine("게임을 시작합니다.");
Console.WriteLine("1: 전사 / 2: 마법사 / 3: 궁수");
Console.Write("직업을 선택하세요: ");
string job = Console.ReadLine();

switch (job)
{
    // ...

}

Console.WriteLine("게임을 종료합니다.");

```

```

Console.WriteLine("게임을 시작합니다.");
Console.WriteLine("1: 전사 / 2: 마법사 / 3: 궁수");
Console.Write("직업을 선택하세요: ");
string job = Console.ReadLine();

switch (job)
{
    case "1":
        Console.WriteLine("전사를 선택하셨습니다.");
        break;
    case "2":
        Console.WriteLine("마법사를 선택하셨습니다.");
        break;
    case "3":
        Console.WriteLine("궁수를 선택하셨습니다.");
        break;
    default:
        Console.WriteLine("올바른 값을 입력해주세요.");
        break;
}

Console.WriteLine("게임을 종료합니다.");

```

## ▼ 6) 3항 연산자

- 3항 연산자는 if 문의 간단한 형태로, 조건식의 결과에 따라 두 값을 선택하는 연산자입니다.

(조건식) ? 참일 경우 값 : 거짓일 경우 값;

- 다음은 예시입니다.

```

int currentExp = 1200;
int requiredExp = 2000;

# 삼항 연산자
string result = (currentExp >= requiredExp) ? "레벨업 가능" : "레벨업 불가능";
Console.WriteLine(result);

# if else 문
if (currentExp >= requiredExp)
{
    Console.WriteLine("레벨업 가능");
}
else
{

```

```
Console.WriteLine("레벨업 불가능");  
}
```

## 02. 조건문 심화 실습



실습을 통해 조금더 자세히 알아보시다

### ▼ 1) 홀수 / 짝수 구분하기

- % 나머지 연산을 이용하여 짝수 구분

```
Console.Write("번호를 입력하세요: ");  
int number = int.Parse(Console.ReadLine());  
  
if (number % 2 == 0)  
{  
    Console.WriteLine("짝수입니다.");  
}  
else  
{  
    Console.WriteLine("홀수입니다.");  
}
```

### ▼ 2) 등급 출력

- else if 문에서 사용된 예시를 switch 문으로 변경

```
int playerScore = 100;  
string playerRank = "";  
  
switch (playerScore / 10)  
{  
    case 10:  
    case 9:  
        playerRank = "Diamond";  
        break;  
    case 8:  
        playerRank = "Platinum";  
        break;  
}
```

```

    case 7:
        playerRank = "Gold";
        break;
    case 6:
        playerRank = "Silver";
        break;
    default:
        playerRank = "Bronze";
        break;
}

Console.WriteLine("플레이어의 등급은 " + playerRank + "입니다.");

```

### ▼ 3) 로그인 프로그램

- && 와 || 논리연산자에 대해 다시 확인해봅시다.

```

string id = "myid";
string password = "mypassword";

Console.Write("아이디를 입력하세요: ");
string inputId = Console.ReadLine();
Console.Write("비밀번호를 입력하세요: ");
string inputPassword = Console.ReadLine();

if (inputId == id && inputPassword == password)
{
    Console.WriteLine("로그인 성공!");
}
else
{
    Console.WriteLine("로그인 실패...");
}

```

### ▼ 4) 알파벳 판별 프로그램

- 입력한 문자가 알파벳인지 여부를 판단

```

Console.Write("문자를 입력하세요: ");
char input = Console.ReadLine()[0];

if (input >= 'a' && input <= 'z' || input >= 'A' && input <= 'Z')
{
    Console.WriteLine("알파벳입니다.");
}
else
{
    Console.WriteLine("알파벳이 아닙니다.");
}

```



### 03. 반복문



일련의 명령문을 반복해서 실행하는 제어문

#### ▼ 1) for 문

- for문은 초기식, 조건식, 증감식을 사용하여 반복문을 작성합니다.
- 초기식은 반복문이 시작될 때 단 한 번 실행되고,  
조건식은 반복문이 실행될 때마다 평가되며, 참(true)인 경우 반복문이 계속 실행됩니다.  
증감식은 반복문이 실행될 때마다 실행되는 식입니다.

```
for (초기식; 조건식; 증감식)
{
    // 조건식이 참인 경우 실행되는 코드
}
```

#### ▼ [코드스니펫] for 문 기초

```
for (int i = 0; i < 10; i++)
{
    Console.WriteLine(i);
}
```

- 다음은 기본적인 for문의 예시입니다.

```
for (int i = 0; i < 10; i++)
{
    Console.WriteLine(i);
}

int i = 0;
for (i = 0 ; i < 10 ; i++)
{
    Console.WriteLine(i);
}
```

- 다음은 0 ~ 10 사이의 짝수를 출력하는 예제입니다.

```
for (int i = 0; i < 10; i += 2 )
{
    Console.WriteLine(i);
}
```

## ▼ 2) while 문

- while문은 조건식이 참(true)인 동안 코드 블록을 반복적으로 실행합니다.

```
while (조건식)
{
    // 조건식이 참인 경우 실행되는 코드
}
```

## ▼ [코드스니펫] while문 기초

```
int i = 0;
while (i < 10)
{
    Console.WriteLine(i);
    i++;
}
```

- 다음은 while문 예시입니다.

```
int i = 0;
while (i < 10)
{
    Console.WriteLine(i);
    i++;
}
```

```
int count = 0;
while (count < 10)
{
    Console.WriteLine("적을 처치했습니다! 남은 적 수: " + (10 - count - 1));
    count++;
}

Console.WriteLine("축하합니다! 게임에서 승리하셨습니다!");
```

### ▼ 3) for 문 vs while 문

```
int sum = 0;

for (int i = 1; i <= 5; i++)
{
    sum += i;
}

Console.WriteLine("1부터 5까지의 합은 " + sum + "입니다.");
```

```
int sum = 0;
int i = 1;

while (i <= 5)
{
    sum += i;
    i++;
}

Console.WriteLine("1부터 5까지의 합은 " + sum + "입니다.");
```

- for 문은 반복 횟수를 직관적으로 알 수 있고, 반복 조건을 한 눈에 확인할 수 있어 가독성이 좋습니다.
- while 문은 반복 조건에 따라 조건문의 실행 횟수가 유동적이며, 이에 따라 코드가 더 간결할 수 있습니다.
- 따라서 어떤 반복문을 사용할지는 코드의 흐름에 따라 상황에 맞게 선택하는 것이 좋습니다.

#### ▼ 4) do-while 문

- do-while문은 while문과 비슷하지만, 조건식을 검사하기 전에 먼저 코드 블록을 한번 실행하는 점이 다릅니다.

```
do
{
    // 조건식이 참인 경우 실행되는 코드
}
while (조건식);
```

```
int sum = 0;
int num;

do
{
    Console.Write("숫자를 입력하세요 (0 입력 시 종료): ");
    num = int.Parse(Console.ReadLine());
    sum += num;
} while (num != 0);

Console.WriteLine("합계: " + sum);
```

#### ▼ 5) foreach문

- foreach문은 배열이나 컬렉션에 대한 반복문을 작성할 때 사용합니다.

```
foreach (자료형 변수 in 배열 또는 컬렉션)
{
    // 배열 또는 컬렉션의 모든 요소에 대해 반복적으로 실행되는 코드
}
```

```
string[] inventory = { "검", "방패", "활", "화살", "물약" };

foreach (string item in inventory)
{
```

```
    Console.WriteLine(item);  
}
```

## ▼ 6) 중첩반복문

### • 이차원 반복문

```
for (int i = 0; i < 5; i++)  
{  
    for (int j = 0; j < 3; j++)  
    {  
        Console.WriteLine("i: {0}, j: {1}", i, j);  
    }  
}
```

### ▼ [코드스니펫] 구구단 출력

```
for (int i = 2; i <= 9; i++)  
{  
    for (int j = 1; j <= 9; j++)  
    {  
        Console.WriteLine(i + " x " + j + " = " + (i * j));  
    }  
}
```

## ▼ 7) Break & Continue

- **break** 는 반복문을 중지시키는 역할
- **continue** 는 현재 반복을 중지하고 다음 반복을 진행하는 역할

### ▼ [코드스니펫] break와 continue 기초

```
for (int i = 1; i <= 10; i++)  
{  
    Console.WriteLine(i);  
}
```

```

for (int i = 1; i <= 10; i++)
{
    if (i % 3 == 0)
    {
        continue; // 3의 배수인 경우 다음 숫자로 넘어감
    }

    Console.WriteLine(i);
    if (i == 7)
    {
        break; // 7이 출력된 이후에는 반복문을 빠져나감
    }
}

```

### ▼ [코드스니펫] break와 continue 예제

```

int sum = 0;

while (true)
{
    Console.Write("숫자를 입력하세요: ");
    int input = int.Parse(Console.ReadLine());

}

Console.WriteLine("합계: " + sum);

```

```

int sum = 0;

while (true)
{
    Console.Write("숫자를 입력하세요: ");
    int input = int.Parse(Console.ReadLine());

    if (input == 0)
    {
        Console.WriteLine("프로그램을 종료합니다.");
        break;
    }

    if (input < 0)
    {
        Console.WriteLine("음수는 무시합니다.");
        continue;
    }

    sum += input;
}

```


```

        Console.WriteLine("현재까지의 합: " + sum);
    }

    Console.WriteLine("합계: " + sum);

```

## 04. 반복문 심화 실습

 실습을 통해 조금더 자세히 알아보시다

### ▼ 1) 가위바위보

```

string[] choices = { "가위", "바위", "보" };
string playerChoice = "";
string computerChoice = choices[new Random().Next(0, 3)];

while (playerChoice != computerChoice)
{
    Console.Write("가위, 바위, 보 중 하나를 선택하세요: ");
    playerChoice = Console.ReadLine();

    Console.WriteLine("컴퓨터: " + computerChoice);

    if (playerChoice == computerChoice)
    {
        Console.WriteLine("비겼습니다!");
    }
    else if ((playerChoice == "가위" && computerChoice == "보") ||
             (playerChoice == "바위" && computerChoice == "가위") ||
             (playerChoice == "보" && computerChoice == "바위"))
    {
        Console.WriteLine("플레이어 승리!");
    }
    else
    {
        Console.WriteLine("컴퓨터 승리!");
    }
}

```

### ▼ 2) 숫자 맞추기

```

int targetNumber = new Random().Next(1, 101); ;
int guess = 0;
int count = 0;

Console.WriteLine("1부터 100 사이의 숫자를 맞춰보세요.");

while (guess != targetNumber)
{
    Console.Write("추측한 숫자를 입력하세요: ");
    guess = int.Parse(Console.ReadLine());
    count++;

    if (guess < targetNumber)
    {
        Console.WriteLine("좀 더 큰 숫자를 입력하세요.");
    }
    else if (guess > targetNumber)
    {
        Console.WriteLine("좀 더 작은 숫자를 입력하세요.");
    }
    else
    {
        Console.WriteLine("축하합니다! 숫자를 맞추셨습니다.");
        Console.WriteLine("시도한 횟수: " + count);
    }
}

```

이전 강의

4강 연산자와 문자열 처리

다음 강의

6강 배열과 컬렉션

---

Copyright © TeamSparta All rights reserved.