**1. 논리 연산자**

* **AND와 OR 연산자**

테이블이(가) 표시된 사진

자동 생성된 설명

* **NOT 연산자**

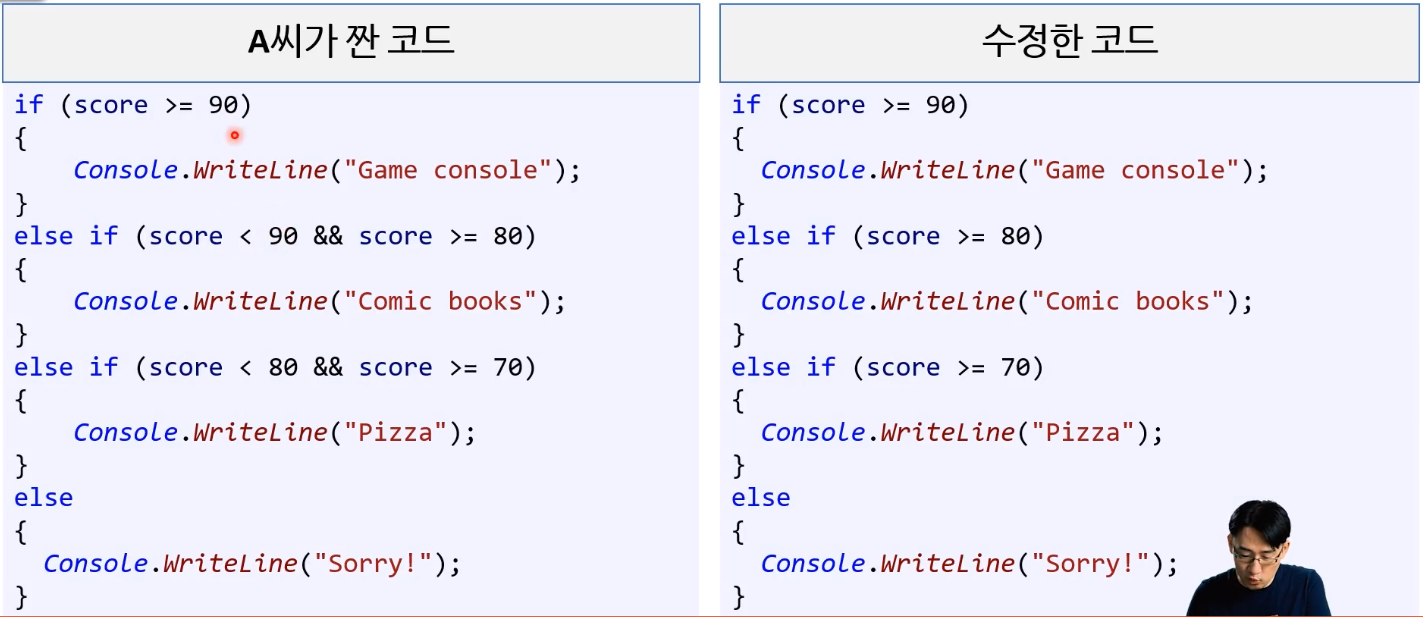


* **if문 안의 표현식들은 종종 평가되지 않을 때가 있음**
* 평가되지 않는다는 말은 실행되지 않는다는 것임
* **OR 논리 연산자와 표현식 평가**- ex)  
   텍스트이(가) 표시된 사진

  자동 생성된 설명
* 앞선(좌항) 표현식이 참이라면 || 연산자의 우항은 실행되지 않음  
  - 이미 해당 구문은 참이기 때문에 우항을 평가하지 않음
* 앞선 표현식이 거짓이라면 || 연산자의 우항도 평가해야 함  
  - 좌항만으로 진리값을 알 수 없으므로 우항도 평가해야 함
* **AND 논리 연산자와 표현식 평가**- ex)  
   텍스트이(가) 표시된 사진

  자동 생성된 설명
* 앞선(좌항) 표현식이 거짓이라면 && 연산자의 우항은 실행되지 않음  
  - 이미 해당 구문은 거짓이기 때문에 우항을 평가하지 않음
* 앞선 표현식이 참이라면 && 연산자의 우항도 평가해야 함  
  - 좌항만으로 진리값을 알 수 없으므로 우항도 평가해야 함

**2. if / else / else if 문과 논리**

* **AND 논리 연산자와 표현식 평가**- ex)  
  
* **올바른 로직을 짤 수 있는 능력이 필요함**- ex)  
   테이블이(가) 표시된 사진

  자동 생성된 설명
* 조건문의 순서가 올바른지 확인할 것
* 이미 배제된 로직을 굳이 다시 생각하지 않아도 됨  
  - 불필요하게 반복된 로직이 들어갔는지 확인
* 논리적으로 말이 되는지 확인
* 이러한 능력은 디버깅 할 때도 중요함

**3. 조건 연산자(삼항 연산자)**

* **삼항 연산자라고도 함**
* ?와 : 기호를 사용
* 불리언 표현식을 평가하여 참과 거짓일 때 서로 다른 반환값을 반환
* ex)  
  텍스트이(가) 표시된 사진

  자동 생성된 설명
* 매우 매우 간단한 비교를 할 때만 조건 연산자를 사용하는 것이 바람직함  
  - 이 경우 if / else 문을 사용하는 것보다 빠를 수 있다
* **특히, 남용하지 말 것**
* 조건 연산자 안에 조건 연산자를 쓸 경우 가독성이 떨어짐

**4. 연산자 우선순위**

* **수학과 마찬가지로 대부분 연산자 결합 순서는 왼쪽에서 오른쪽**
* **소수의 연산자만이 오른쪽에서 왼쪽**- 대입 연산자와 삼항 연산자가 대표적
* **일부는 반드시 알아야 함**
* 사칙연산 수준의 우선순위 정도는 알아야 함
* **일부는 반드시 알 필요는 없음**
* 이러한 경우에는 소괄호로 처리하면 됨
* **주의: 연산자 우선순위와 평가 순서는 다름**
* ex)  
  텍스트이(가) 표시된 사진

  자동 생성된 설명
* 우선순위는 &&가 ||보다 높음
* 하지만 다음 코드는 실행조차 되지 않음  
  - 3+2 == 5, 1+2 == 2
* **|| 및 && 연산자는 왼쪽에 있는 표현식의 평가를 강제**하기 때문임  
  - 왼쪽에서 오른쪽으로 평가(연산)된다고 기억하자
* 평가는 순서대로 진행되지만 연산자 우선법칙에 따라 && 연산 후 || 연산을 진행함