

조건이 하나일 때

문제

만약 18살 이상이면 "성인입니다"를 출력하고, 만약 18살 미만이라면 "미성년자입니다."를 출력해야 한다.

```
if (조건) {
    실행문; // 조건이 참일 때 실행되는 코드
}

public class 조건문 {
    public static void main(String[] args) {
        int age = 20; // 사용자 나이

        if (age >= 18) {
            System.out.println("성인입니다.");
        }
        if (age < 18) {
            System.out.println("미성년자입니다.");
        }
    }
}
```

else를 활용한 조건문

```
if (조건) {
    실행문; // 조건이 참일 때 실행되는 코드
} else {
    실행문; // 조건이 거짓일 때 실행되는 코드
}

public class 조건문 {
    public static void main(String[] args) {

        int age = 20; // 사용자 나이

        if (age >= 18) {
            System.out.println("성인입니다.");
        } else {
            System.out.println("미성년자입니다.");
        }
    }
}

결과 : 성인입니다.
```

조건이 여러 개일 때 : else if

문제

7세 이하일 경우: "미취학"

8세 이상 13세 이하일 경우: "초등학생"

14세 이상 16세 이하일 경우: "중학생"

17세 이상 19세 이하일 경우: "고등학생"

20세 이상일 경우: "성인"

```
public class elseif_조건문 {  
    public static void main(String[] args) {  
        int age = 14;  
        if(age <= 7) { //~7: 미취학  
            System.out.println("미취학");  
        } if(age >= 8 && age <= 13) { //~8~13: 초등학생  
            System.out.println("초등학생");  
        }  
        if(age >= 14 && age <= 16) { //~14~16: 중학생  
            System.out.println("중학생");  
        }  
        if(age >= 17 && age <= 19) { //~17~19: 고등학생  
            System.out.println("고등학생");  
        }  
        if(age >= 20) { //~20~: 성인  
            System.out.println("성인");  
        }  
    }  
}
```

결과 : 중학생

단점 1. 불필요한 조건 검사: 이미 조건을 만족해도 불필요한 다음 조건을 계속 검사한다. 이로 인해 시간 낭비가 된다.

단점 2. 코드 효율성: 예를 들어서 나이가 8살인 초등학생이라면 미취학을 체크하는 조건인 `age <= 7` 을 통해 나이가 이미 8살이 넘는다는 사실을 알 수 있다. 다음에 있는 초등학생을 체크하는 조건에서 `age >= 8 && age <= 13` 라는 2가지 조건을 모두 수행한다. 여기서 `age >= 8` 이라는 조건은 이미 앞의 `age <= 7` 이라는 조건과 관련이 있다. 결과적으로 조건을 중복 체크한 것이다.

✓ else if 를 사용하면 불필요한 조건 검사를 피하고 코드의 효율성을 향상시킬 수 있다.

```

if (조건) {
    실행문; // 조건1이 참일 때 실행되는 코드
} else if (조건) {
    실행문; // 조건1이 거짓이고, 조건2가 참일 때 실행되는 코드
} else if (조건) {
    실행문; // 조건2이 거짓이고, 조건3이 참일 때 실행되는 코드
} else {
    실행문; // 모든 조건이 거짓일 때 실행되는 코드
}

```

else if로 바꾼 조건문

```

public class elseif_조건문 {
    public static void main(String[] args) {
        int age = 14;

        if(age <= 7) { //~7: 미취학
            System.out.println("미취학");
        } else if(age <= 13) { //~8~13: 초등학생
            System.out.println("초등학생");
        } else if(age <= 16) { //~14~16: 중학생
            System.out.println("중학생");
        } else if(age <= 19) { //~17~19: 고등학생
            System.out.println("고등학생");
        } else { //~20~: 성인
            System.out.println("성인");
        }
    }
}

```

결과 : 중학생

if문과 else if문

if-else 사용: 서로 연관된 조건이어서, 하나로 묶을 때 사용

if 각각 사용: 독립 조건일 때 사용

예시1. if 각각 사용: 독립 조건일 때

```
int price = 10000; // 아이템 가격
int age = 10; // 나이
int discount = 0;
if (price >= 10000) {
    discount = discount + 1000;
    System.out.println("10000원 이상 구매, 1000원 할인");
}
if (age <= 10) {
    discount = discount + 1000;
    System.out.println("어린이 1000원 할인");
}
System.out.println("총 할인 금액: " + discount + "원");
```

결과

10000원 이상 구매, 1000원 할인

어린이 1000원 할인

총 할인 금액: 2000원

if-else문을 잘못 적용한 경우

```
int price = 10000; // 아이템 가격
int age = 10; // 나이
int discount = 0;
if (price >= 10000) {
    discount = discount + 1000;
    System.out.println("10000원 이상 구매, 1000원 할인");
} else if (age <= 10) {
    discount = discount + 1000;
    System.out.println("어린이 1000원 할인");
} else {
    System.out.println("할인 없음");
}
System.out.println("총 할인 금액: " + discount + "원");
```

결과

10000원 이상 구매, 1000원 할인

총 할인 금액: 1000원

첫 번째로 충족되는 조건인 1000원 할인만 적용되고, if 문을 빠져나온다. 따라서 사용자는 나머지 할인을 놓치게 된다.

switch문

```
switch (조건식) {  
    case value1:  
        // 조건식의 결과 값이 value1일 때 실행되는 코드  
        break;  
    case value2:  
        // 조건식의 결과 값이 value2일 때 실행되는 코드  
        break;  
    default:  
        // 조건식의 결과 값이 위의 어떤 값에도 해당하지 않을 때 실행되는 코드  
}
```

특징

1. break 문은 현재 실행 중인 코드를 끝내고 switch 문을 빠져나가게 하는 역할을 한다.
만약 break 문이 없으면, 일치하는 case 이후의 모든 case 코드들이 순서대로 실행된다.
2. default 는 조건식의 결과값이 모든 case 의 값과 일치하지 않을 때 실행된다. if 문의 else 와 같다.
default 구문은 선택이다.

예제

```
public class switch_조건문 {  
    public static void main(String[] args) {  
        //grade 1:1000, 2:2000, 3:3000, 나머지: 500  
        int grade = 2;  
        int coupon;  
        switch (grade) {  
            case 1:  
                coupon = 1000;  
                break;  
            case 2:  
                coupon = 2000;  
                break;  
            case 3:  
                coupon = 3000;  
                break;  
            default:  
                coupon = 500;  
        }  
        System.out.println("발급받은 쿠폰 " + coupon);  
    }  
}
```

결과 : 발급받은 쿠폰 2000

자바 14버전 새로운 switch문

```
public class 자바14_switch_조건문 {  
    public static void main(String[] args) {  
        //grade 1:1000, 2:2000, 3:3000, 나머지: 500  
        int grade = 2;  
        int coupon = switch (grade) {  
            case 1 -> 1000;  
            case 2 -> 2000;  
            case 3 -> 3000;  
            default -> 500;  
        };  
        System.out.println("발급받은 쿠폰 " + coupon);  
    }  
}
```

if문 vs switch문 비교

switch 문의 조건식을 넣는 부분을 보면 참 거짓의 결과가 나오는 조건이 아니라, 단순히 값만 넣을 수 있다.

switch 문은 조건식이 특정 case 와 같은지만 체크할 수 있다. 쉽게 이야기해서 값이 같은지 확인하는 연산만 가능하다. (문자도 가능)

반면에 if 문은 참 거짓의 결과가 나오는 조건식을 자유롭게 적을 수 있다. 예) $x > 10$, $x == 10$

정리하자면 switch 문 없이 if 문만 사용해도 된다. 하지만 특정 값에 따라 코드를 실행할 때는 switch 문을 사용하면 if문보다 간결한 코드를 작성할 수 있다.

삼항 연산자

(조건) ? 참 : 거짓

```
public class 삼항연산자 {  
    public static void main(String[] args) {  
        int age = 18;  
        String status;  
        if (age >= 18) {  
            status = "성인";  
        } else {  
            status = "미성년자";  
        }  
        System.out.println("age = " + age + " status = " + status);  
    }  
}
```

결과 : age = 18 status = 성인

삼항 연산자를 사용한 구문

```
public class 삼항연산자 {  
    public static void main(String[] args) {  
        int age = 18;  
        String status = (age >= 18) ? "성인" : "미성년자";  
        System.out.println("age = " + age + " status = " + status);  
    }  
}
```

결과 : age = 18 status = 성인

String status = (age >= 18) ? "성인" : "미성년자"; //age=18

String status = (true) ? "성인" : "미성년자"; //조건이 참이므로 참 표현식 부분이 선택된다.

String status = "성인"; //결과

삼항 연산자 없이 if 문만 사용해도 된다. 하지만 단순히 참과 거짓에 따라서 특정 값을 구하는 삼항 연산자를 사용하면 if 문보다 간결한 코드를 작성할 수 있습니다.