



JAVA

GU, DA HAE

조 건 문
제 어 문
보 조 제 어 문

J A V A

제 어 문

제어문

| Control Statement

구분	키워드
선택문	if, else if, else
	switch
반복문	while
	do while
	for
	foreach
보조 제어문	break
	continue

프로그램은 보통 위에서 아래로 순차적으로 실행되는데, 특정 블록이 수행되거나 수행되지 않도록 해야 할 경우나 반복 실행해야 하는 경우도 있다. 이런 경우에 사용하는 것이 제어문이다.

제어문은 프로그래머가 특정 문장을 건너뛰게도 하고 특정 문장을 반복 실행하게도 하면서 프로그램의 흐름을 조절하는 역할을 한다.

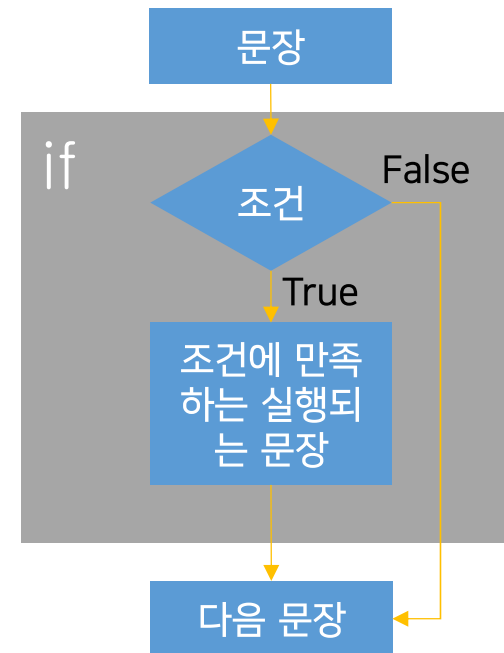
if 문

| Control Statement

주어진 조건을 만족하는 경우에만 특정 문장을 수행하도록 하는 제어문.

J A V A

```
if (조건식)
{
    코드
}
```



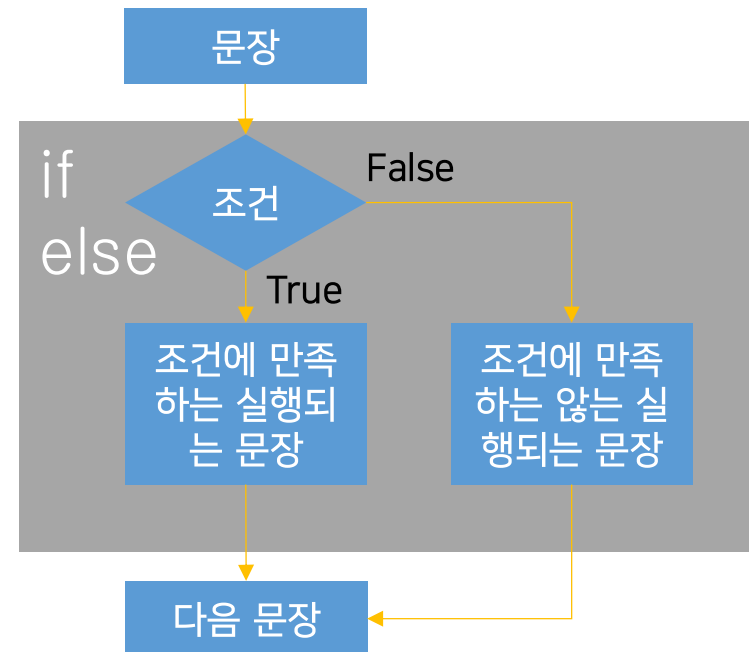
J A V A

else 문

| Control Statement

if문에 주어진 조건을 만족하지 않는 경우에만 특정 문장을 수행하도록 하는 제어문.

```
if (조건식)
{
    코드1
}
else
{
    코드2
}
```



J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        if(true){  
            System.out.println("TRUE!");  
        }  
        else{  
            System.out.println("FALSE!");  
        }  
    }  
}
```

J A V A

예제

```
public class Ex02{  
    public static void main(String[] args){  
        int num = 10;  
  
        if(num > 0){  
            System.out.println("num is more bigger than 0");  
        }  
        else{  
            System.out.println("num is less than or equal to 0");  
        }  
    }  
}
```

J A V A

예제

```
public class Ex03{  
    public static void main(String[] args){  
        int num = 10;  
  
        if(num < 0){  
            System.out.println("num is less than 0");  
        }else{  
            if(num < 10){  
                System.out.println("num is bigger than or equal to 0 and less than 10");  
            }else{  
                System.out.println("num is bigger than or equal to 10");  
            }  
        }  
    }  
}
```


else if 문

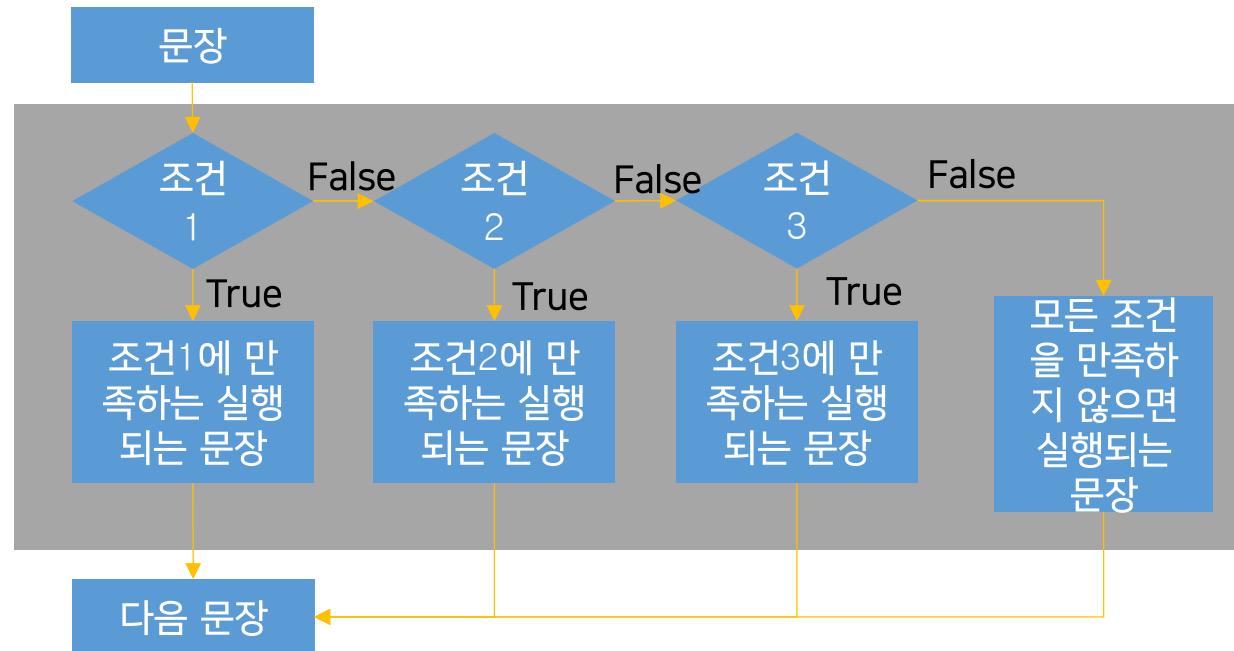
| Control Statement

else if는 else인 상태에서 조건식을 지정할 때 사용한다. 단, else if는 단독으로 사용할 수 없다.

```
if (조건식)
{
    코드1
}
else if (조건식)
{
    코드2
}
```

플로우 차트

J A V A



J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        int num = 10;  
  
        if(num < 0){  
            System.out.println("num is less than 0");  
        }  
        else if(num < 10){  
            System.out.println("num is bigger than or equal to 0 and less than 10");  
        }else{  
            System.out.println("num is bigger than or equal to 10");  
        }  
    }  
}
```

switch 문

| Control Statement

switch 문은 if 문에 비해 조건이 많아도 쉽게 처리할 수 있다.

정수만 처리할 수 있다.

```
switch (변수) {  
  case 숫자1:  
    코드1  
    break;  
  case 숫자2:  
    코드2  
    break;  
  default:  
    코드3;  
    break;  
}
```

J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        int num = 2;  
        switch(num){  
            case 1:  
                System.out.println("Hello!");  
                break;  
            case 2:  
                System.out.println("Hi!");  
                break;  
            default:  
                System.out.println("Bye!");  
        }  
    }  
}
```

해보기

switch문으로 아래와 같이 출력해보자.

변수 num의 범위가 $0 \leq \text{num} < 10$ 일 때) 0 이상 10 미만

변수 num의 범위가 $10 \leq \text{num} < 20$ 일 때) 10 이상 20 미만

그 외의 경우) 음수 또는 20 이상

J A V A

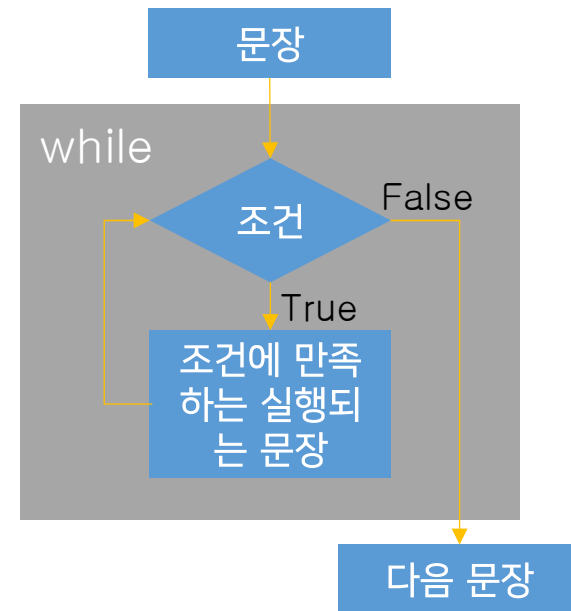
while 문

| Control Statement

주어진 조건을 만족하는 경우에만 특정 문장을 반복 수행하도록 하는 제어문.

조건식에 참(true)을 지정하면 영원히 끝나지 않는 무한 반복(루프)을 수행한다.

```
while (조건식)
{
    반복할 코드
}
```



J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        int i = 0;  
  
        while(i < 5){  
            System.out.println("LOOP : " + i);  
            i++;  
        }  
    }  
}
```


J A V A

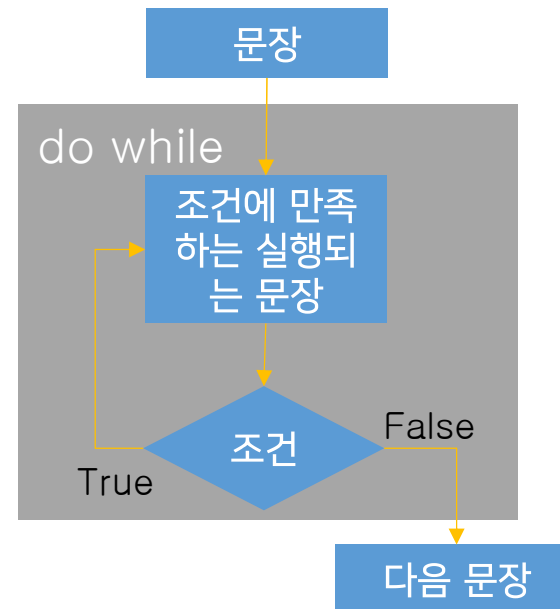
do while 문

| Control Statement

먼저 한 번 수행하고, 주어진 조건을 만족하는 경우에만 특정 문장을 반복 수행하도록 하는 제어문.

조건식에 참(1)을 지정하면 영원히 끝나지 않는 무한 반복(루프)을 수행한다.

```
do  
{  
    반복할 코드  
}while(조건식);
```



J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        int i = 0;  
  
        do{  
            System.out.println("LOOP : " + i);  
            i++;  
        }while(i < 5);  
    }  
}
```

J A V A

해보기

0과 500 사이의 숫자 중 3과 4의 공배수를 차례대로 출력하고, 출력한 숫자들의 합을 구해보자.

J A V A

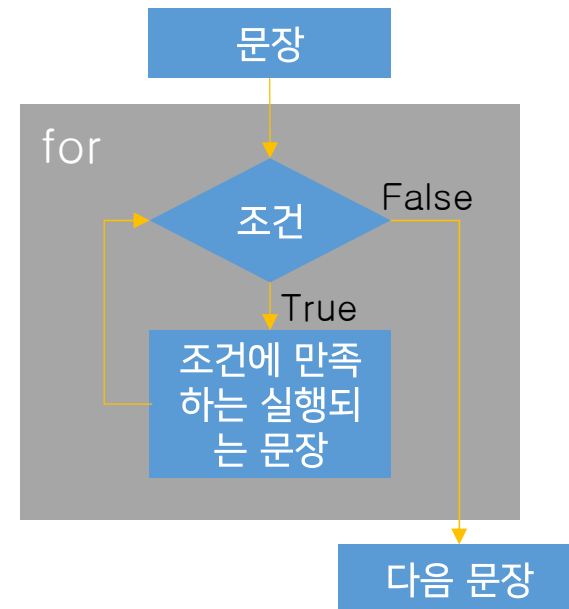
for 문

| Control Statement

주어진 조건을 만족하는 경우에만 특정 문장을 정해진 횟수만큼 반복 수행하도록 하는 제어문.

조건식에 참(true)을 지정하면 영원히 끝나지 않는 무한 반복(루프)을 수행한다.

```
for (초기식; 조건식; 변화식)
{
    반복할 코드
}
```



J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        for(int i = 0; i<5; i++){  
            System.out.println("LOOP : " + i);  
        }  
    }  
}
```

J A V A

예제

```
public class Ex02{  
    public static void main(String[] args){  
        for(int i = 0, j = 10; i<5; i++, j--){  
            System.out.println("LOOP : i-" + i + ", j-" + j);  
        }  
    }  
}
```

보조 제어문

| Control Statement

`break`는 for, while, do while, switch 문법에서 제어 흐름을 벗어나기 위해 사용한다.

`continue`는 제어 흐름(반복)을 유지한 상태에서 코드의 실행만 건너뛰는 역할을 한다.

J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        int i = 0;  
  
        while(true){  
            if(i == 10) break;  
  
            System.out.println("LOOP : " + i);  
        }  
    }  
}
```


J A V A

예제

```
public class Ex02{  
    public static void main(String[] args){  
        int i = 0;  
  
        while(true){  
            if(i % 2 == 0){  
                i++;  
                continue;  
            }  
  
            System.out.println("i : " + i);  
            i++;  
        }  
    }  
}
```

J A V A

해보기

1부터 시작하여 모든 3과 7의 배수를 차례대로 더한다. 그 합이 1000을 넘어가면 멈추고, 합과 마지막에 더한 값을 출력해보자.

중첩 반복문

| Control Statement

반복문 안에 반복문이 들어가는 형태를 **중첩 루프(다중 루프)**라고 한다.

반복문의 루프 인덱스 변수는 index의 첫 글자를 따서 i를 사용하는데 반복문 안에 반복문이 들어갈 때는 변수 이름을 i, j, k처럼 i부터 **알파벳 순서**대로 짓는다.

```
for(초기식i; 조건식i; 변화식i){  
    ...  
    for(초기식j; 조건식j; 변화식j){  
        ...  
    }  
    ...  
}
```

J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        for(int i = 0; i < 3; i++){  
            System.out.println("i : " + i);  
            for(int j = 0; j < 5; j++){  
                System.out.println("--- j : " + j);  
            }  
        }  
    }  
}
```

J A V A

예제

```
public class Ex02{  
    public static void main(String[] args){  
        for(int i = 0; i < 3; i++){  
            System.out.println("i : " + i);  
            for(int j = 0; j < 5; j++){  
                System.out.println("--- j : " + j);  
                if(j % 2 == 0)  
                    break;  
            }  
        }  
    }  
}
```

J A V A

예제

```
public class Ex03{  
    public static void main(String[] args){  
        LABEL :  
  
        for(int i = 0; i < 3; i++){  
            System.out.println("i : " + i);  
            for(int j = 0; j < 5; j++){  
                System.out.println("--- j : " + j);  
                if(j % 2 == 0)  
                    break LABEL;  
            }  
        }  
    }  
}
```

J A V A

해보기

구구단의 홀수단만 출력하며, 3단은 $3*2$ 까지, 5단은 $5*4$ 까지, 7단은 $7*6$ 까지 9단은 $9*8$ 까지 출력해보자.