

06 true/false

부울이란?

변수에 담을 수 있는 값의 종류 중 하나입니다. 부울 값은 true(참) 또는 false(거짓)의 값을 가지게 됩니다. 이 부울 값은 후에 알고리즘을 위한 07 조건문, 08 반복문에서 사용하게 됩니다.

부울 사용하기

```
1| local bool1 = true
2| local bool2 = false
3| print( bool1 )
4| print( bool2 )
```

실행 결과

```
true
false
```

관계연산자와 논리연산자

부울 값은 보통 true/false라고 바로 입력해서 사용하기보다는 07 조건문, 08 반복문 등에서 조건식에 주로 사용하게 됩니다. 관계연산자와 논리 연산자를 이용해 조건식을 생성할 수 있고, 조건식은 해당 식이 참인지 거짓인지에 따라 true / false 값을 갖게 됩니다.

관계 연산자 (A, B는 정수 or 실수)

A > B	A가 B보다 크다
A < B	A가 B보다 작다
A >= B	A가 B보다 크거나 같다
A <= B	A가 B보다 작거나 같다
A == B	A와 B의 값이 같다
A ~= B	A와 B의 값이 같지 않다

논리 연산자 (A, B는 true or false)

A and B	A와 B가 모두 true면 true
A or B	A와 B 중 하나라도 true면 true
not A	A가 false면 true

부울 사용하기

```
1| local N = 10
2| local M = 20
3| print( N == M )
4| print( N > M or N < M )
```

실행 결과

```
false
true
```

07 조건문

조건문이란?

조건문은 해당 시점에 지정된 조건식이 true일 경우에 실행할 부분을 작성할 수 있게 해줍니다. 조건식은 if를 사용해서 작성하기 때문에 if문이라고도 합니다.

조건문 사용하기

+) 삽화 추가 예정

```
1| if (조건식) then
2|     -- 조건식이 true면 실행할 코드
3| end
```

```
1| local N = 10
2| local M = 10
3|
4| if ( N == M ) then
5|     print( "N과 M의 값이 같다." )
6| end
```

실행 결과

N과 M의 값이 같다.

조건식은 **if (조건식) then ~ end** 형식을 사용해서 작성합니다. then과 end 사이에 조건식이 true이면 실행할 코드들을 작성하면 됩니다.

```
1| if (조건식 A) then
2|     -- 조건식 A가 true일 때 실행할 코드
3| elseif (조건식 B) then
4|     -- 조건식 A가 false, 조건식 B가 true일 때 실행할 코드
5| else
6|     -- 조건식 A와 조건식 B가 모두 false일 때 실행할 코드
7| end
```

elseif를 사용해서 앞에 조건식이 false일 경우에 실행할 새로운 조건문을 작성할 수 있고, **else**를 이용해서 모든 조건식이 false일 때 실행할 코드들을 지정할 수 있습니다.

[0] [8] 반복문

반복문이란?

반복문은 조건식이 성립하는 동안 반복해서 실행할 코드를 작성할 수 있게 해줍니다.
반복문은 while문과 for문 두 가지 종류가 있습니다.

while문

```
1| while (조건식) do
2|     -- 조건식이 true일 동안 실행할 코드
3| end
```

while문은 간단히 while (조건식) do ~ end 형식을 해서 작성합니다. Do ~ end에 조건식이 true일 때 반복할 코드를 작성해주면 됩니다.

for문

```
1| for i = start, end, add do
2|     -- i가 end값에 도달할 때까지 반복할 코드
3| end
```

+) 삽화 추가 예정

for문은 while문보다 조금 복잡합니다. 변수 i가 start에서 시작해서 add씩 더해서 end에 도달할 때까지 돌아가는 반복문입니다. 한번 코드를 돌때마다 i에 add 값만큼 더해지게 되고, 이때 add가 1이라면 for문을 작성할 때 생략해도 됩니다.

N번 반복하는 반복문 예시

```
1| local i = 1
2| while i <= N do
3|     -- N번 실행할 코드
4|     i = i + A
5| end
```

```
1| for i = 1, N do
2|     -- N번 실행할 코드
3| end
```

add값이 1이므로 생략