

### ตารางค่าความจริงนิเสธ (¬)

```
JS logic.js X
logic > JS logic.js > ...
1 let p = [true, false];
2 let not_p = []
3 for (let i = 0; i < p.length; i++) {
4   not_p.push(!p[i]); // not p
5 }
6 console.log(p);
7 console.log(not_p);
8

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>node
[ true, false ]
[ false, true ]

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>
```

### ตารางค่าความจริงและ (∧)

```
logic > JS logic.js > ...
1 let p=[true,true,false,false];
2 let q=[true,false,true,false];
3 let pAq = []
4 for (let i = 0; i < p.length; i++) {
5   pAq.push(p[i] && q[i]); //p and q
6 }
7 console.log(p);
8 console.log(q);
9 console.log(pAq);

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
[ true, true, false, false ]
[ true, false, true, false ]
[ true, false, false, false ]

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>
```

### ตารางค่าความจริงก็ต่อเมื่อ ( $\leftrightarrow$ )

```
JS logic.js X
logic > JS logic.js > ...
1 let p=[true,true,false,false];
2 let q=[true,false,true,false];
3 let pXq = []
4 for (let i = 0; i < p.length; i++) {
5   if(p[i]== q[i])
6     pXq.push(true);
7   else
8     pXq.push(false);
9 }
10 console.log(pXq);
11

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
[ true, false, false, true ]

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>
```

### การพิสูจน์ประพจน์ด้วยค่าความจริง

```
JS logic.js X
logic > JS logic.js > ...
1 let p= (2*(4 - 8) == 16);
2 let q= (3*(1 - 5) == 12);
3 console.log(p+' or '+q);
4 console.log(p||q); // p∨q
5

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
false or false
false

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>
```

JS logic.js X

logic > JS logic.js > ...

```
1 let x=3;
2 let p= ((x*x)==9);
3 let q= (x == 3);
4 let r= (x == -3);
5 console.log(p+' <-> ('+q+' and '+r+'));
6 if(p == (q&&r))
7     console.log(true);
8 else
9     console.log (false);
```

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\11 2 เทอม 1\Math> node "c:  
true <-> (true and false)  
false  
  
C:\Users\KEN\Desktop\Home work\11 2 เทอม 1\Math>

JS logic.js X

logic > JS logic.js > ...

```
1 let p= true;
2 let q =false; //answer
3 let pAq= false;
4 if((p&&q)==pAq)
5 console.log(p+' and '+q);
6
```

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\11 2 เทอม 1\Math>node  
true and false

```
JS logic.js X
logic > JS logic.js > ...
2 let q =true; //answer
3 let result= false;
4 if((p==q)==result)
5 console.log(p+' <-> '+q);
6

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
false <-> true
```

```
JS logic.js X
logic > JS logic.js > ...
1 let p= false; //answer
2 let q = false;
3 let result= true;
4 if( (p!=true) ==result)
5 console.log(p+' -> '+q);
6

OUTPUT JUPYTER SQL CONSOLE DEBUG CONSOLE TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
false -> false

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>
```

JS logic.js X

logic > JS logic.js > ...

```
1 let p= false; //answer
2 let q =false;
3 let result= false;
4 if((p||q)==result)
5 console.log(p+' or '+q);
6
```

OUTPUT

JUPYTER


SQL CONSOLE

DEBUG CONSOLE

TERMINAL

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node  
false or false

C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>



```
1 let p = [true, true, false, false];
2 let q = [true, false, true, false];
3 let pRq = [];
4 let NpRq = [];
5 let qAp = [];
6 let result = [];
7 for (let i = 0; i < p.length; i++) {
8     pRq.push(p[i] || q[i]); //p or q }
9     console.log(pRq);
10    for (let i = 0; i < pRq.length; i++) {
11        NpRq.push(!pRq[i]); //not p or q }
12        console.log(NpRq);
13    }
14 }
15 for (let i = 0; i < q.length; i++) {
16     qAp.push(q[i] && p[i]); //q and p
17 }
18 console.log(qAp);
19 for (let i = 0; i < NpRq.length; i++) { //(2) -> (3)
20     if ((NpRq[i] == true) && (qAp[i] == false))
21         result.push(false);
22     else {
23         result.push(true);
24     }
25     console.log(result);
26 }
```



```
1 [ true ]
2 [ false ]
3 [ true, true ]
4 [ false, false ]
5 [ false, false, false ]
6 [ true, true, true ]
7 [ false, false, false, false ]
8 [ false, false, false, false, false ]
9 [ false, false, false, false, false, false ]
10 [ true, true, true, false ]
11 [
12     false, false,
13     false, false,
14     false, false,
15     false
16 ]
17 [
18     false, false,
19     false, false,
20     false, false,
21     false, false
22 ]
23 [
24     false, false,
25     false, false,
26     false, false,
27     false, false,
28     false
29 ]
30 [
31     false, false, false,
32     false, false, false,
33     false, false, false,
34     true
35 ]
36 [
37     true, true, true,
38     true, true, true,
39     true, true
40 ]
41 [
42     true, true, true,
43     true, true, true,
44     true, true, true
45 ]
46 [
47     true, true, true,
48     true, true, true,
49     true, true, true,
50     true
51 ]
```



```
1  let p = [true, true, false, false];
2  let q = [true, false, true, false];
3  let Np = [];
4  let NpTq = [];
5  let pRq = [];
6  let result = [];
7  for (let i = 0; i < p.length; i++) {
8      Np.push(!p[i]); // not p
9  }
10 console.log(Np);
11 for (let i = 0; i < Np.length; i++) {
12     if ((Np[i] == true) && (q[i] == false))
13         NpTq.push(false);
14     else
15         NpTq.push(true);
16 }
17 console.log(NpTq);
18 for (let i = 0; i < p.length; i++) {
19     pRq.push(p[i] || q[i]); // q and p
20 }
21 console.log(pRq);
22 for (let i = 0; i < NpTq.length; i++) {
23     //(2) <-> (3)
24     if ((NpTq[i] == pRq[i])) // สมมูลหรือไม่
25         result.push(true);
26     else
27         result.push(false);
28 }
29 console.log(result);
30
```






```
1 [ false, false, true, true ]
2 [ true, true, true, false ]
3 [ true, true, true, false ]
4 [ true, true, true, true ]
```



```
1 let p = [true, true, true, true, false, false, false, false];
2 let q = [true, true, false, false, true, true, false, false];
3 let r = [true, false, true, false, true, false, true, false];
4 let pTq = [];
5 let qTr = [];
6 let pTqAqTr = [];
7 let pTr = [];
8 let result = [];
9 for (let i = 0; i < p.length; i++) {
10     if ((p[i] == true) && (q[i] == false)) //p ->q
11         pTq.push(false);
12     else
13         pTq.push(true);
14 }
15 for (let i = 0; i < q.length; i++) {
16     if ((q[i] == true) && (r[i] == false)) //q ->r
17         qTr.push(false);
18     else
19         qTr.push(true);
20 }
21 for (let i = 0; i < pTq.length; i++) {
22     pTqAqTr.push(pTq[i] && qTr[i]);
23 }
24 for (let i = 0; i < p.length; i++) {
25     if ((p[i] == true) && (r[i] == false)) //p ->r
26         pTr.push(false);
27     else
28         pTr.push(true);
29 }
30 for (let i = 0; i < pTqAqTr.length; i++) {
31     if ((pTqAqTr[i] == true) && (pTr[i] == false))
32         result.push(false);
33     else
34         result.push(true);
35 }
36 console.log(result);
37
```



```
1  [  
2      true, true, true,  
3      true, true, true,  
4      true, true  
5  ]
```



```
1  let p = [true, true, false, false];
2  let q = [true, false, true, false];
3  let Np = [];
4  let Nq = [];
5  let pTq = [];
6  let pTqANq = [];
7  let result = [];
8  for (let i = 0; i < p.length; i++) {
9      Np.push(!p[i]); // not p
10     Nq.push(!q[i]); // not q
11     if ((p[i] == true) && (q[i] == false)) //p -> q
12         pTq.push(false);
13     else
14         pTq.push(true);
15
16     pTqANq.push(pTq && Nq); // and
17     if ((pTqANq[i] == true) && (Np[i] == false))
18         result.push(false);
19     else
20         result.push(true);
21 }
22 console.log(Np);
23 console.log(Nq);
24 console.log(pTq);
25 console.log(pTqANq);
26 console.log(result);
27
```



```
1  [ false, false, true, true ]
2  [ false, true, false, true ]
3  [ true, false, true, true ]
4  [
5      [ false, true, false, true ],
6      [ false, true, false, true ],
7      [ false, true, false, true ],
8      [ false, true, false, true ]
9  ]
10 [ true, true, true, true ]
```