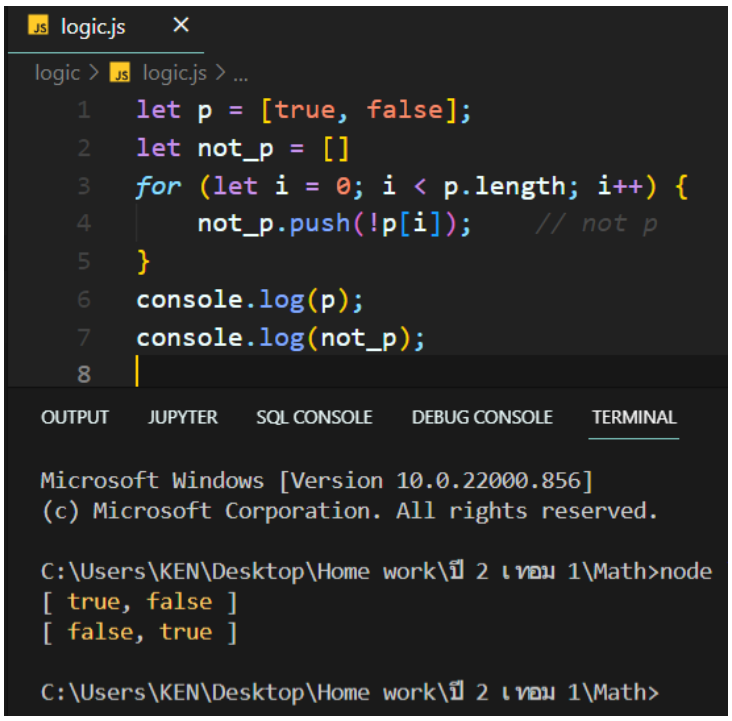
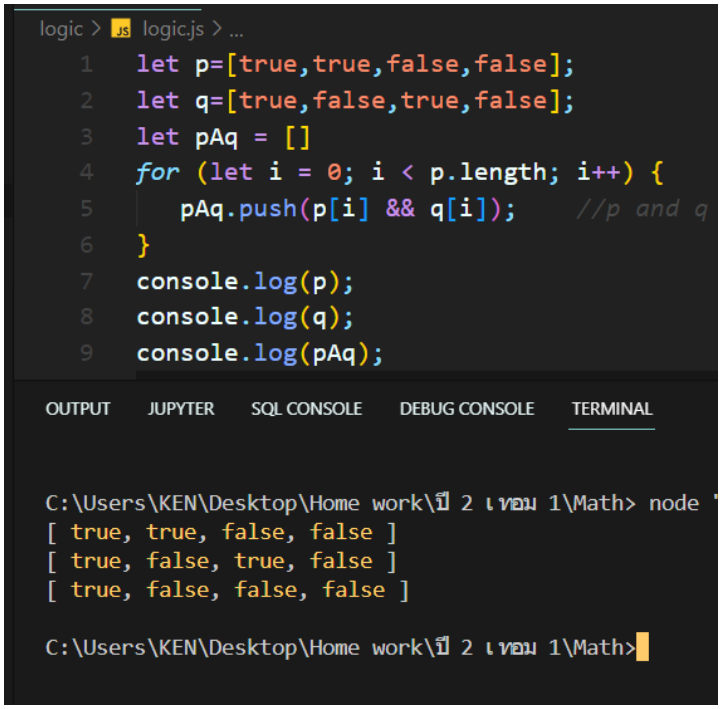
ตารางค่าความจริงนิเสธ (¬)



ตารางค่าความจริงและ (∧)



ตารางค่าความจริงก็ต่อเมื่อ (↔)

logic.js

logic > logic.js > ...

```
let p=[true,true,false,false];
let q=[true,false,true,false];
let pXq = []
for (let i = 0; i < p.length; i++) {
    if(p[i]== q[i])
        pXq.push(true);
    else
        pXq.push(false);
}
console.log(pXq);
```

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

logic.js

logic > logic.js > ...

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

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>
```

logic.js

logic > 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\ปี 2 เทอม 1\Math> node "c
true <-> (true and false)
false

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

logic.js

logic > 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\ปี 2 เทอม 1\Math>node
true and false
```

logic.js

logic > 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
```

logic.js

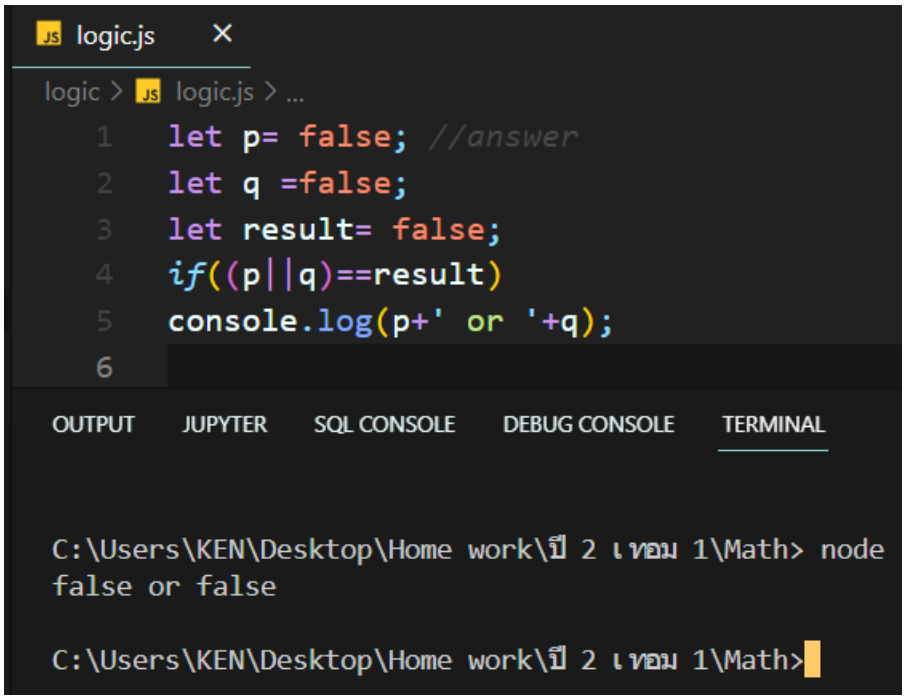
logic > logic.js > ...

```
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>
```

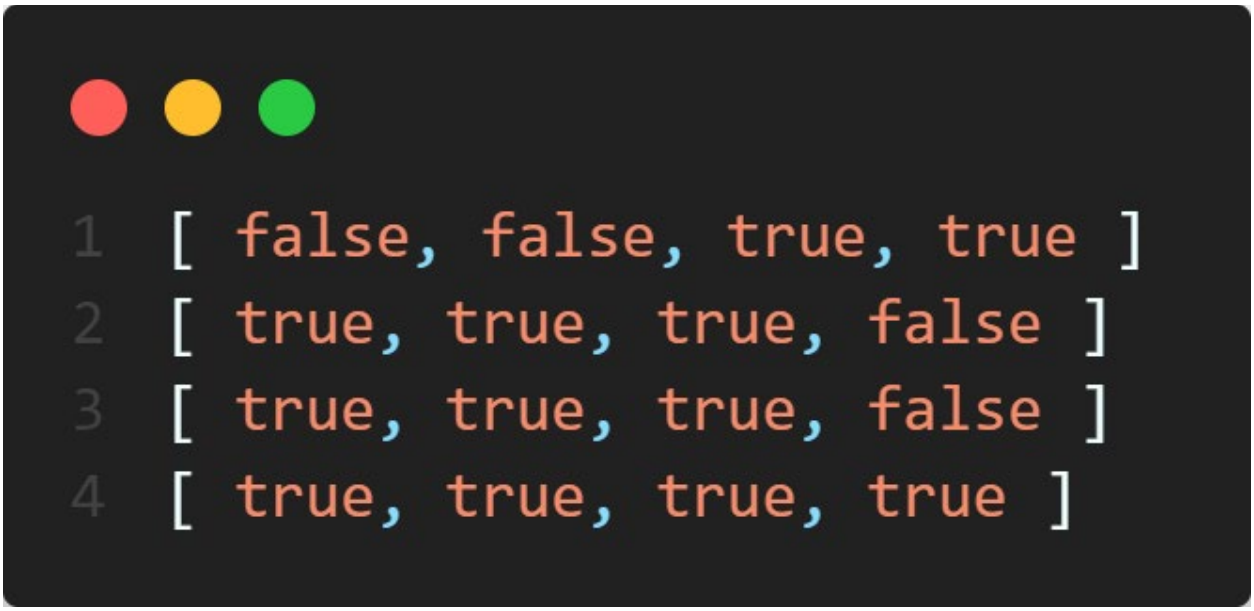

logic.js
logic > logic.js > ...
1 let p= false; //answer
2 let q =false;
3 let result= false;
4 if((p||q)==result)
5 console.log(p+' or '+q);
6
OUTPUT
JUPYTER
SQL CONSOLE
DEBUG CONSOLE
TERMINAL
C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math> node
false or false
C:\Users\KEN\Desktop\Home work\ปี 2 เทอม 1\Math>

```
let p = [true, true, false, false];
let q = [true, false, true, false];
let pRq = [];
let NpRq = [];
let qAp = [];
let result = [];
for (let i = 0; i < p.length; i++) {
    pRq.push(p[i] || q[i]); //p or q }
    console.log(pRq);
    for (let i = 0; i < pRq.length; i++) {
        NpRq.push(!pRq[i]); //not p or q }
        console.log(NpRq);
    }
}
for (let i = 0; i < q.length; i++) {
    qAp.push(q[i] && p[i]); //q and p
}
console.log(qAp);
for (let i = 0; i < NpRq.length; i++) { //(2) -> (3)
    if ((NpRq[i] == true) && (qAp[i] == false))
        result.push(false);
    else {
        result.push(true);
    }
    console.log(result);
}
```

```
[ true ]
[ false ]
[ true, true ]
[ false, false ]
[ false, false, false ]
[ true, true, true ]
[ false, false, false, false ]
[ false, false, false, false, false ]
[ false, false, false, false, false, false ]
[ true, true, true, false ]
[
  false, false,
  false, false,
  false, false,
  false
]
[
  false, false,
  false, false,
  false, false,
  false, false
]
[
  false, false,
  false, false,
  false, false,
  false, false,
  false
]
[
  false, false, false,
  false, false, false,
  false, false, false,
  true
]
[
  true, true, true,
  true, true, true,
  true, true
]
[
  true, true, true,
  true, true, true,
  true, true, true
]
[
  true, true, true,
  true, true, true,
  true, true, true,
  true
]
```

```
let p = [true, true, false, false];
let q = [true, false, true, false];
let Np = [];
let NpTq = [];
let pRq = [];
let result = [];
for (let i = 0; i < p.length; i++) {
    Np.push(!p[i]); // not p
}
console.log(Np);
for (let i = 0; i < Np.length; i++) {
    if ((Np[i] == true) && (q[i] == false))
        NpTq.push(false);
    else
        NpTq.push(true);
}
console.log(NpTq);
for (let i = 0; i < p.length; i++) {
    pRq.push(p[i] || q[i]); //q and p
}
console.log(pRq);
for (let i = 0; i < NpTq.length; i++) {
    //(2) <-> (3)
    if ((NpTq[i] == pRq[i]))  // สมมูลหรือไม่
        result.push(true);
    else
        result.push(false);
}
console.log(result);

```


[false, false, true, true]
[true, true, true, false]
[true, true, true, false]
[true, true, true, true]

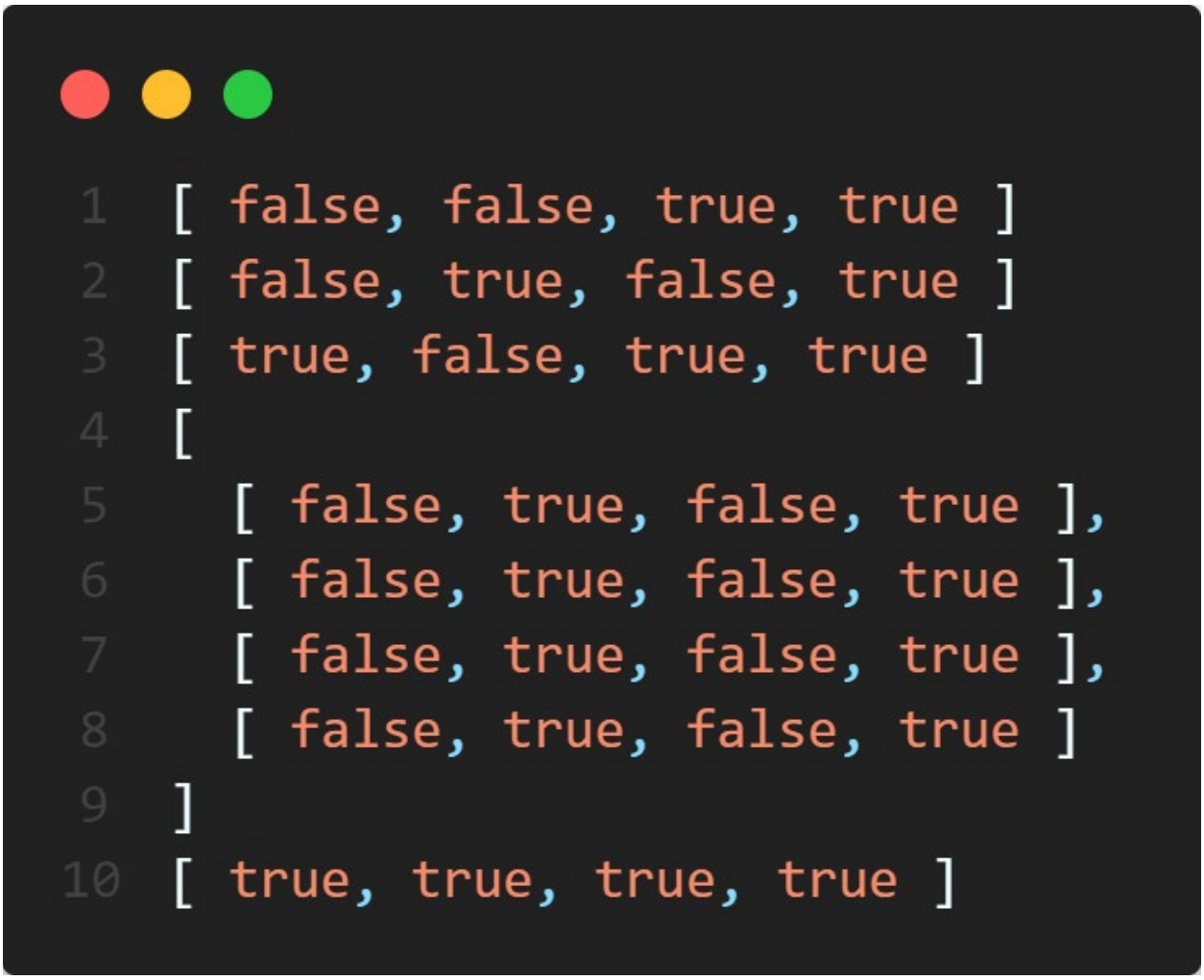
```
let p = [true, true, true, true, false, false, false, false];
let q = [true, true, false, false, true, true, false, false];
let r = [true, false, true, false, true, false, true, false];
let pTq = [];
let qTr = [];
let pTqAqTr = [];
let pTr = [];
let result = [];
for (let i = 0; i < p.length; i++) {
    if ((p[i] == true) && (q[i] == false)) //p ->q
        pTq.push(false);
    else
        pTq.push(true);
}
for (let i = 0; i < q.length; i++) {
    if ((q[i] == true) && (r[i] == false)) //q ->r
        qTr.push(false);
    else
        qTr.push(true);
}
for (let i = 0; i < pTq.length; i++) {
    pTqAqTr.push(pTq[i] && qTr[i]);
}
for (let i = 0; i < p.length; i++) {
    if ((p[i] == true) && (r[i] == false)) //p ->r
        pTr.push(false);
    else
        pTr.push(true);
}
for (let i = 0; i < pTqAqTr.length; i++) {
    if ((pTqAqTr[i] == true) && (pTr[i] == false))
        result.push(false);
    else
        result.push(true);
}
console.log(result);

```


[
true, true, true,
true, true, true,
true, true
]

```
let p = [true, true, false, false];
let q = [true, false, true, false];
let Np = [];
let Nq = [];
let pTq = [];
let pTqANq = [];
let result = [];
for (let i = 0; i < p.length; i++) {
    Np.push(!p[i]); // not p
    Nq.push(!q[i]); // not q
    if ((p[i] == true) && (q[i] == false)) //p -> q
        pTq.push(false);
    else
        pTq.push(true);

    pTqANq.push(pTq && Nq); // and
    if ((pTqANq[i] == true) && (Np[i] == false))
        result.push(false);
    else
        result.push(true);
}
console.log(Np);
console.log(Nq);
console.log(pTq);
console.log(pTqANq);
console.log(result);
```


[false, false, true, true]
[false, true, false, true]
[true, false, true, true]
[
[false, true, false, true],
[false, true, false, true],
[false, true, false, true],
[false, true, false, true]
]
[true, true, true, true]