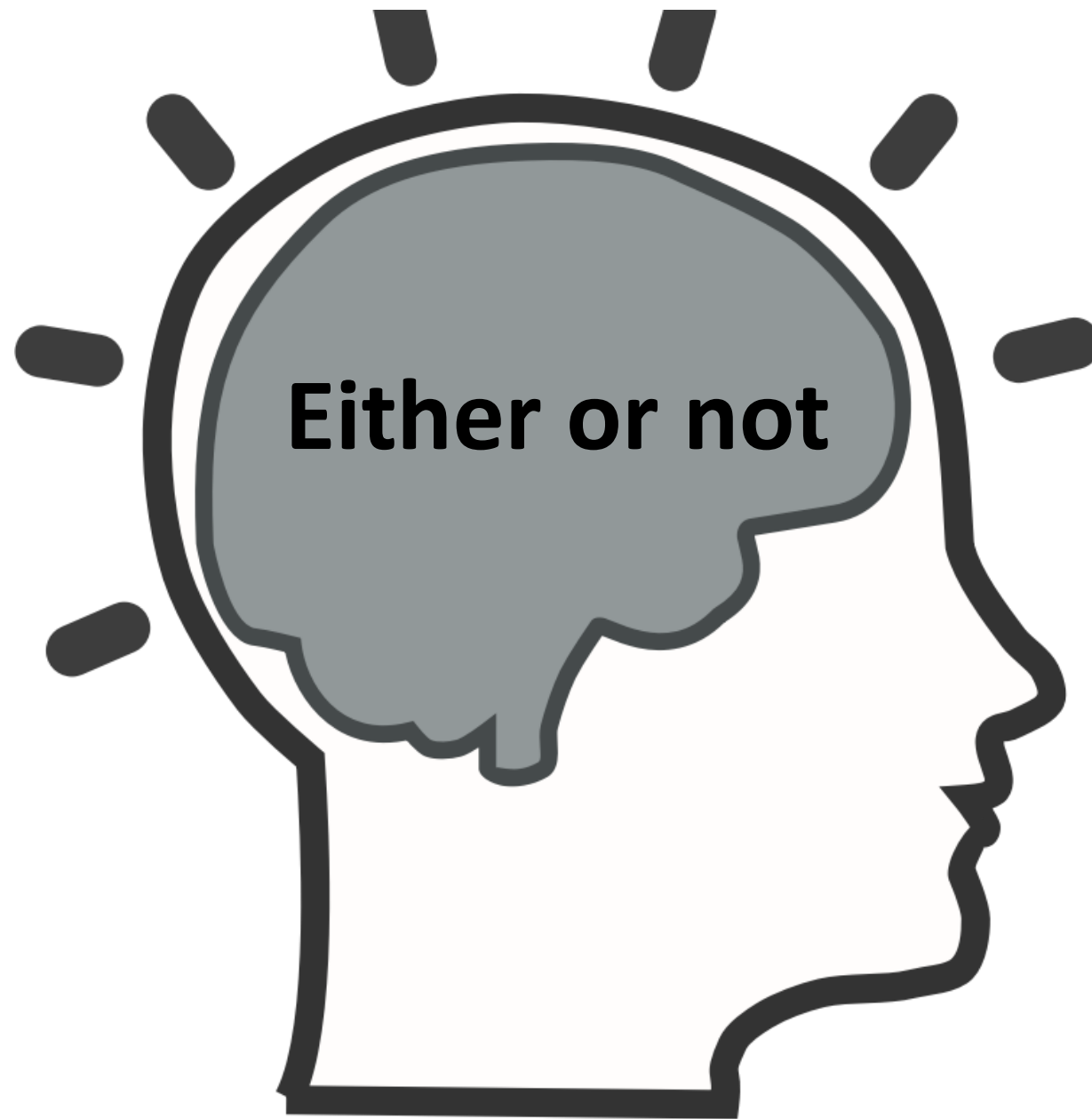
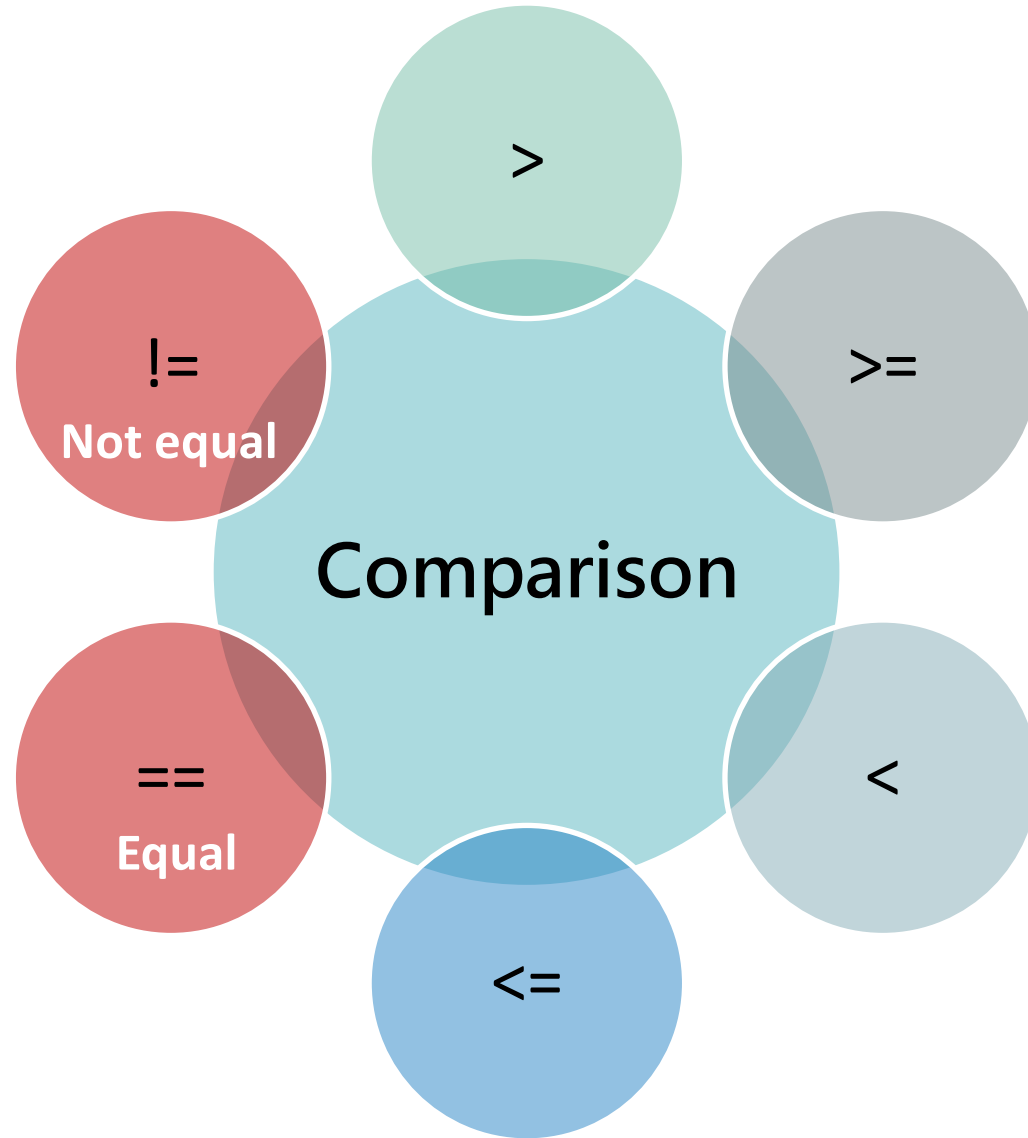




Python

Magic of “else”

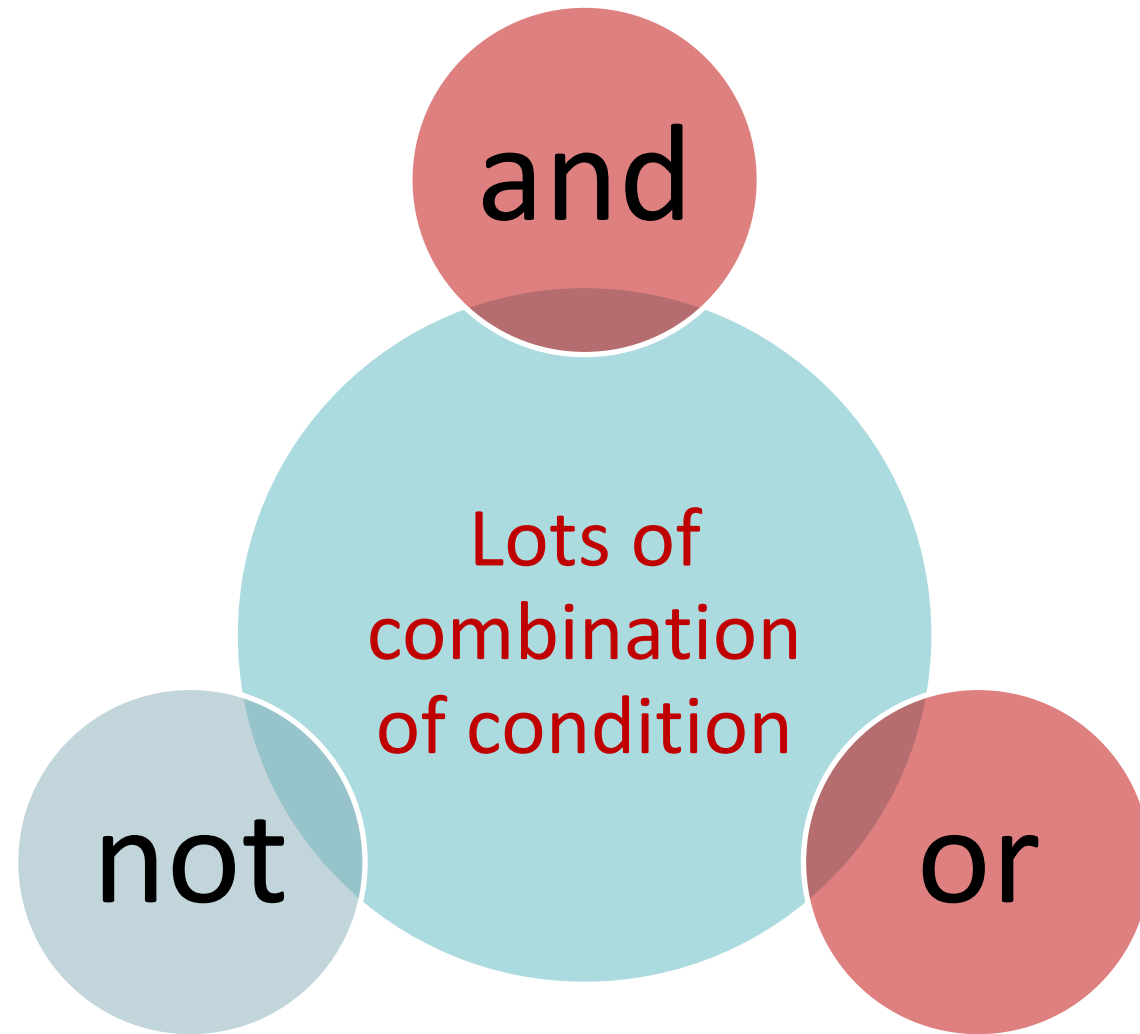




Comparison operators

Operator	Example	Meaning
==	$x==y$	Equal
!=	$x!=y$	Not equal
>	$x>y$	Greater than
<	$x<y$	Less than
>=	$x>=y$	Greater than or equal to
<=	$x<=y$	Less than or equal to





Logical Operator

Operator	Sample	Description
And	Condition1 and condition2 and condition3	All conditions must be established.
Or	Condition1 or condition2 or condition3	One of the conditions is established.
Not	! condition1	The condition1 does not true.



Write a program,
determine a number is
odd or **even**

Exercise





Python

Solve the problem with Python

Please log in to DICE



Code: odd or even

```
num = int(input())  
  
if num % 2 == 0:  
    print( "{0} is even".format(num))  
else:  
    print( "{0} is odd".format(num))
```





Python

Extended concepts



Writing style : block

```
1 num = int(input())
2
3 if num % 2 == 0:
4     print("偶數:", num)
5     print("偶數:", num)
6     print("偶數:", num)
7 else:
8     print("奇數:", num)
9     print("奇數:", num)
10    print("奇數:", num)
```

Block1

Block2



- There is a block below if and else respectively
- The number of consecutive lines with the same indentation is regarded as the same block



Writing style : Semicolon, narrative

■ narrative

```
1 num = int(input())
2
3 if num % 2 == 0:
4     print("偶數:", num)
5     print("偶數:", num)
6     print("偶數:", num)
7 else:
8     print("奇數:", num)
9     print("奇數:", num)
10    print("奇數:", num)
```

- One narrative per line, with or without a semicolon (;) at the end of the narrative
- readable
- Multiple narratives in one line, with a semicolon at the end of each narrative
- Low readability, not recommended

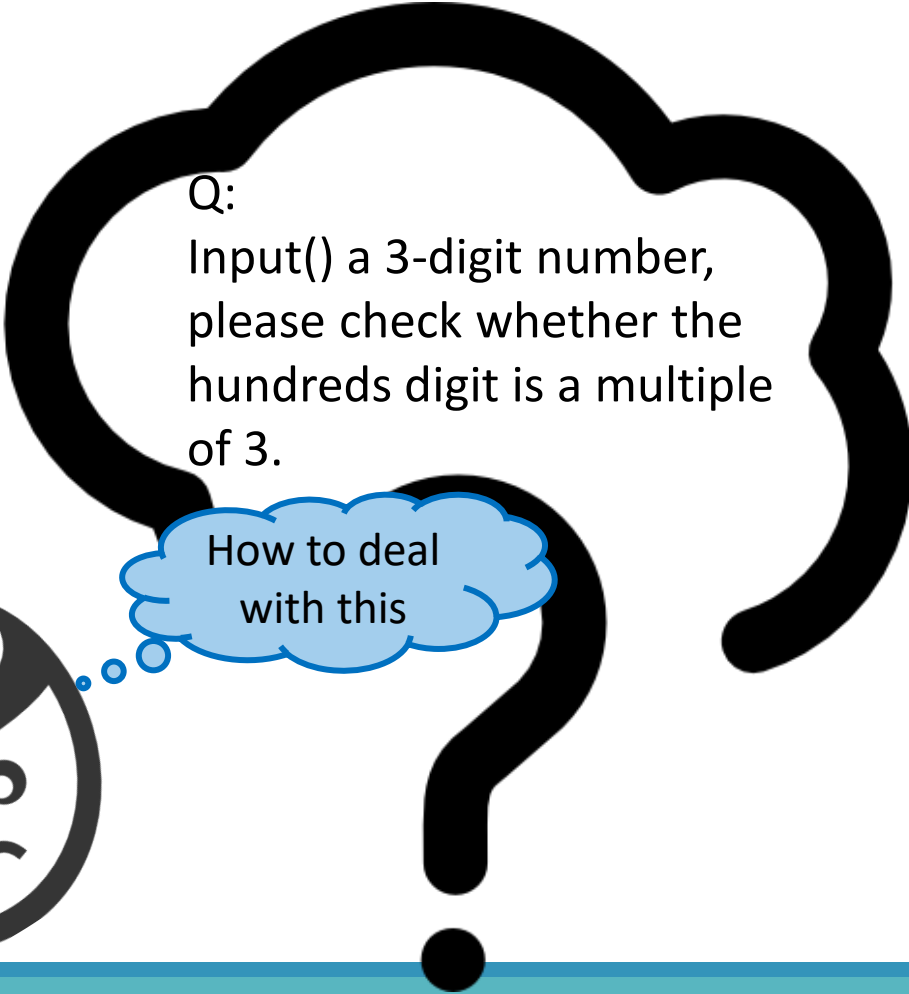
■ semicolon

- In some programming languages (for example: C), a semicolon (;) must be added to the end of the narrative. Python does not require it. However, if multiple narratives are placed on the same line, a semicolon must be added to the end of the narrative. But for the readability of the program, it is recommended to keep a narrative per line and maintain a style that does not use semicolons


```
1 num = int(input())
2
3 if num % 2 == 0:
4     print("偶數:", num); print("偶數:", num); print("偶數:", num);
5 else:
6     print("奇數:", num); print("奇數:", num); print("奇數:", num);
```



Learn to narrow down problems and find out the error step by step



Q:
Input() a 3-digit number,
please check whether the
hundreds digit is a multiple
of 3.



How to deal
with this

■ Tips :

- **Problem Solving** : Decompose a main problem into N items. Once there is a problem with the program, you can confirm whether there are errors in the execution results for each item one by one, and you can quickly find the error!
- Use **comments** to view the code in sections: add **#** in front of each line of the program that has not been executed, and the code after the line **#** will not be executed.



Try it : solve the problem, split the problem and find out the problem!

```
1 num = int(input()) 1. Are the numbers correct?
2 hundred = num // 100 2. Is the hundred digits
3 correct?
4 if hundred % 3 == 0:
5     print("百位數{0}是3的倍數".format(hundred))
6 else:
7     print("百位數{0}不是3的倍數".format(hundred))
8
9 #if : 3. Is it correct for the value of a
10 # printf("Match test case 1") hundred digits is not a multiple of 3?
11 #else :
12 # printf("No match test case 1")
13
14 #if :
15 # printf("Match test case 2")
16 #else :
17 # printf("No match test case 2")
```

Question :

- Input() a 3-digit number, please check whether the hundreds digit is a multiple of 3

Problem solving :

1. Input() number
2. Get the value of the hundreds digit
3. Check whether the hundreds digit is a multiple of 3

