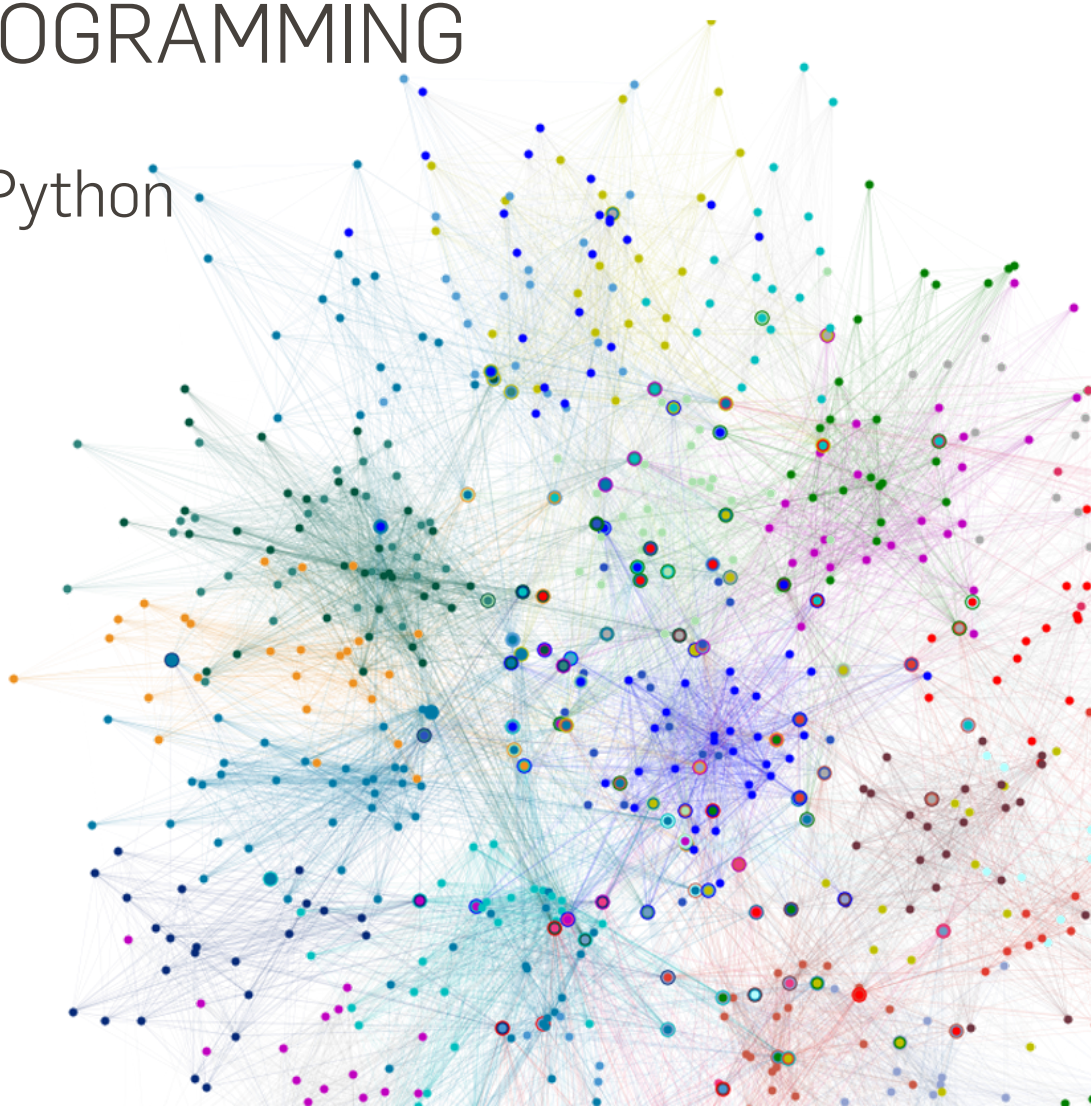
ThangLong University

CS100: INTRO TO PROGRAMMING

Making decisions with code
Conditional statements in Python

October 17, 2017



Lecturer: Thang Duc Nguyen

Email: thangdn.tlu@gmail.com
Phone: (+84)968-486-632

Mathematics and Informatics
ThangLong University

Overview

.

Using conditional statements, you can write Python code that makes decisions in your problems. In this lecture, We're going to learn "Human computer interaction" or "How to use the term flow of control" to refer to the sequence of statements that are executed in a program.

Overview

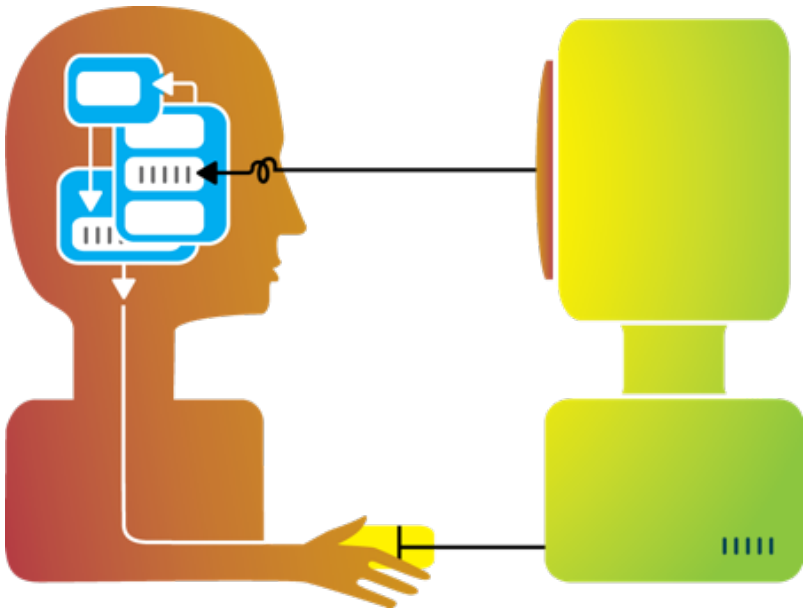
1. Human computer interaction
2. Decision making in python
3. Your challenge
4. Conclusion

HUMAN COMPUTER INTERACTION

Human computer interaction

Two way conversations allow you to do more with computers

- → Websites need your address and payment information so they can ship you products
- → Insurance companies need information to calculate how much you would pay for car insurance
- → Cortana will tell you a joke if you ask her



How can we ask a user for information?

$gedit myprogram.py &

```
1    import stdio
2
3    stdio.write("What's your name? ")
4    name = stdio.readString()
5    stdio.writeln('Hi, '+name+'How are you?')
```

$python3 myprogram.py
> What's your name? Thang
> Hi, Thang. How are you?

How can we ask a user for information?

- → stdio.readString(): To read a string value, and return it
- → stdio.readInt() : To read an integer number, and return it
- → stdio.readFloat() : To read a floating-point number, and return
- → stdio.readBool(): To read a true-false value, and return it

Your challenge

$gedit add.py &

```
1     import stdio
2
3     stdio.write("Enter a number of A = ")
4     a = stdio.readFloat()
5
6     stdio.write("Enter a number of B = ")
7     b = stdio.readFloat()
8
9     stdio.write('min('+str(a)+','+str(b)+')= ')
10    stdio.writeln(min(a,b))
```

$python3 add.py

> Enter a number of A = 5.6

> Enter a number of B = 4.4

> min(5.6, 4.4) = 4.4

DECISION MAKING IN PYTHON

Why make decisions in programming?

Every day we are faced with decisions

- → Should I drive or take the bus?
- → Should I cook at home or go out for dinner?
- → Which laptop should I buy?

Why make decisions in programming?

The choice we make depends on different conditions

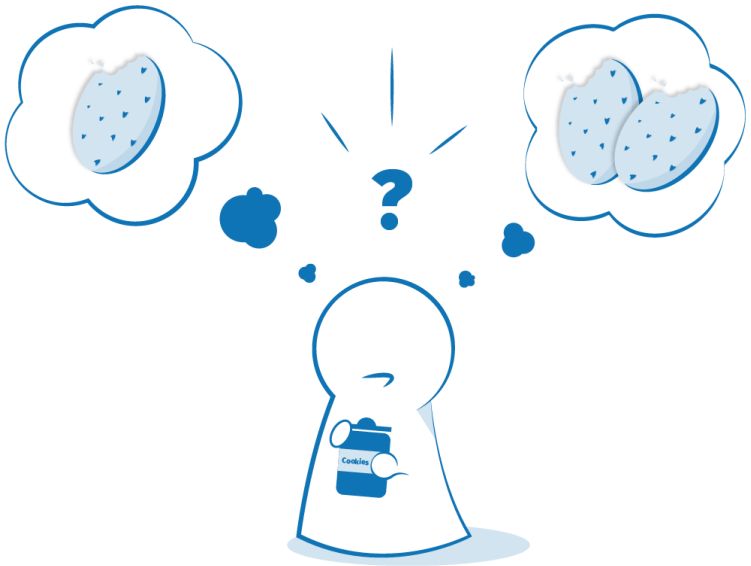
.

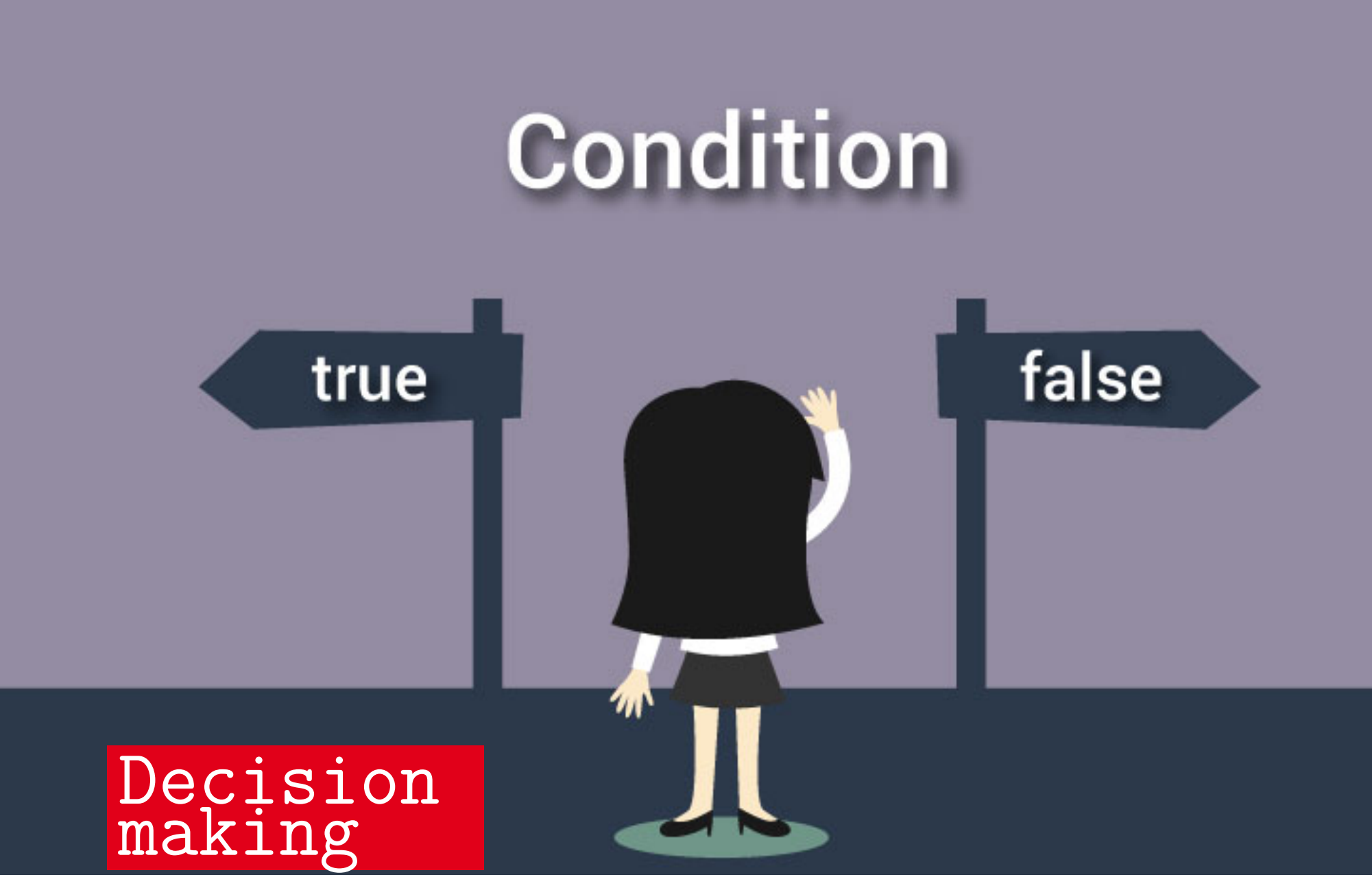
- → Should I drive or take the bus?
  Am I late? What's the price of gas?
- → Should I cook at home or go out for dinner?
  Do I have any food at home? Do I have enough money to go out?
- → Which laptop should I buy?
  How much RAM do I need? How much money do I have?

Conditionals?

In any programming language, code needs to make decisions and carry out actions accordingly depending on different inputs.
For example:

- → In a game, if the player's number of lives is 0, then it's game over.
- → In a weather app, if it is being looked at in the morning, show a sunrise graphic; show stars and a moon if it is nighttime.





Decision making

is required when we want to execute a code only if a certain condition is satisfied.

.

Booleans

The bool data type has just two values: True and False.

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Source: wikibooks.org

Comparisons

The comparison operators ==, !=, <, <=, >, and >= are defined for both integers and floats, and evaluate to a boolean result.

op	*meaning*	True	False
==	*equal*	`2 == 2`	`2 == 3`
!=	*not equal*	`3 != 2`	`2 != 2`
<	*less than*	`2 < 13`	`2 < 2`
<=	*less than or equal*	`2 <= 2`	`3 <= 2`
>	*greater than*	`13 > 2`	`2 > 13`
>=	*greater than or equal*	`3 >= 2`	`2 >= 3`

Comparisons with `int` operands and a `bool` result

Source: intro to programming in python

#Task01

Suppose that a and b are booleans. Show that this expression evaluates to True:

1 (not (a and b) and (a or b)) or ((a and b) or not (a or b))

2 (not (a < b) and not (a > b))

3 a = True

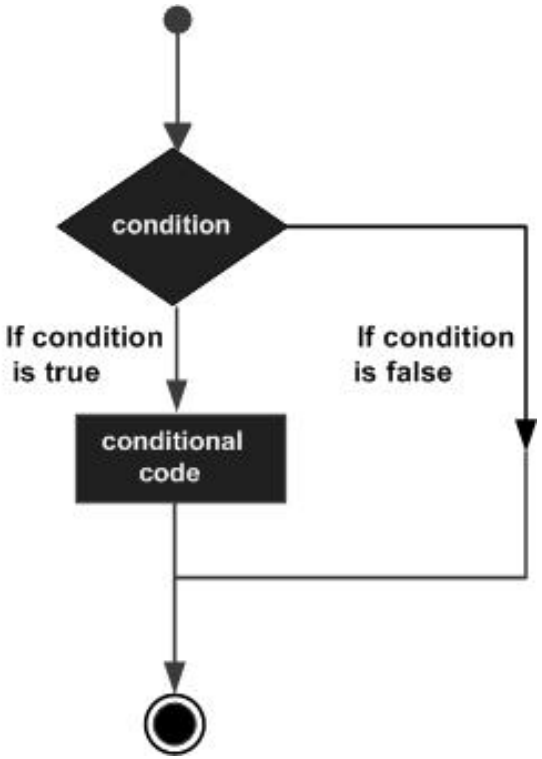
a = not a

a = not a

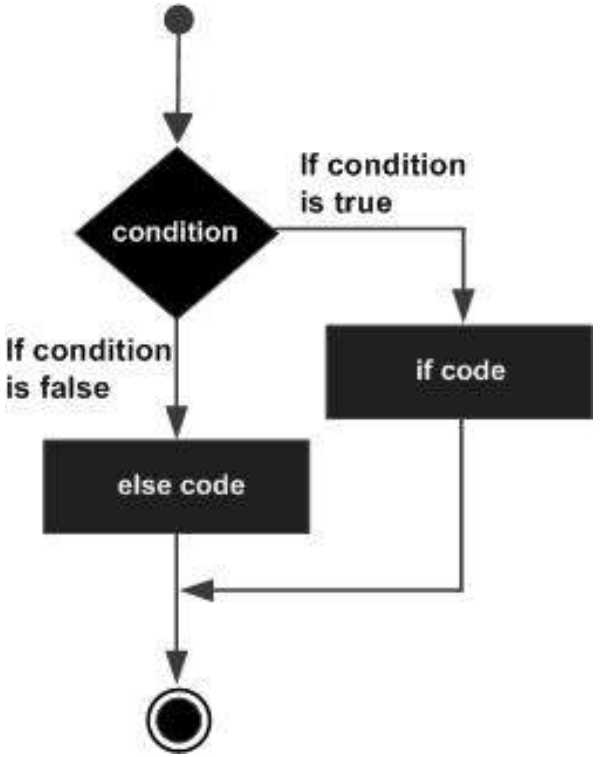
a = not a

a = ?

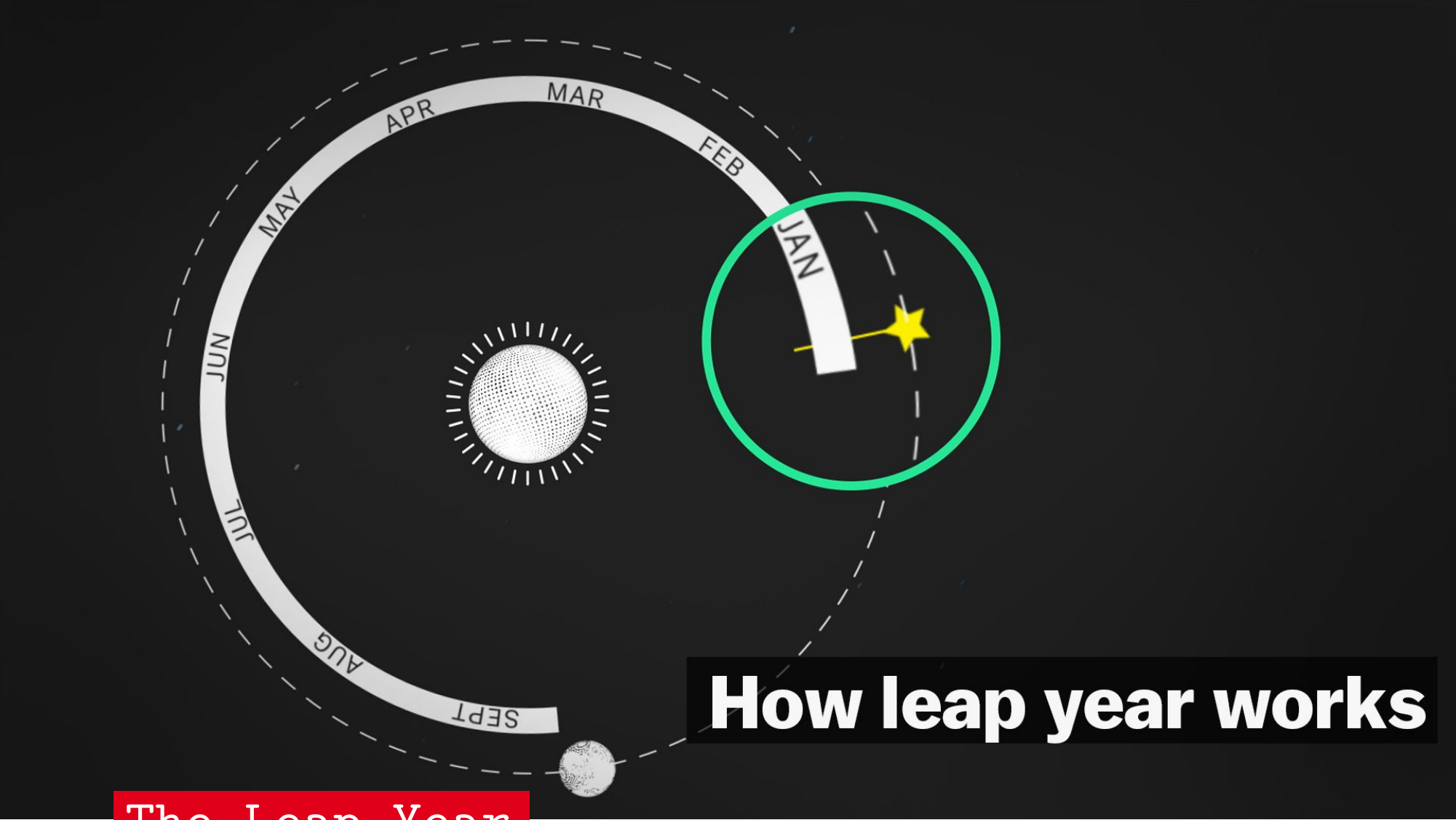
IF and IF ELSE statement



IF statement



IF statement
ELSE statement



The Leap Year Explained

https://www.youtube.com/watch?v=YTOr8_lLqGw

.

.

Leap year program

$gedit leapyear.py &

```
1   import stdio
2   import sys
3
4   year = int(sys.argv[1])
5
6   isLeapYear = (year % 4 == 0)
7   isLeapYear = isLeapYear and (year % 100 != 0)
8   isLeapYear = isLeapYear or  (year % 400 == 0)
9
10  stdio.writeln(isLeapYear)
```

$python3 leapyear.py 2016
> True
$python3 leapyear.py 1900
> False

[Update] Leap year program

$gedit leapyear.py &

```
1       import stdio
2       import sys
3
4       year = int(sys.argv[1])
5
6       if ((year % 4 == 0) and (year % 100 != 0)) or
              (year % 400 == 0):
7           stdio.writeln('It is a leap year')
8       else:
9           stdio.writeln('It is a common year')
```

$python3 leapyear.py 2016
> It is a leap year
$python3 leapyear.py 1900
> It is a common year

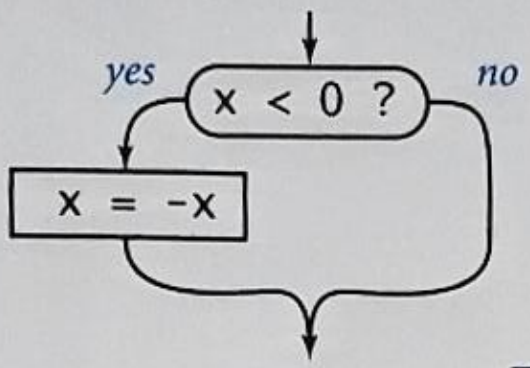
Odd-even number program

$gedit oddeven.py &

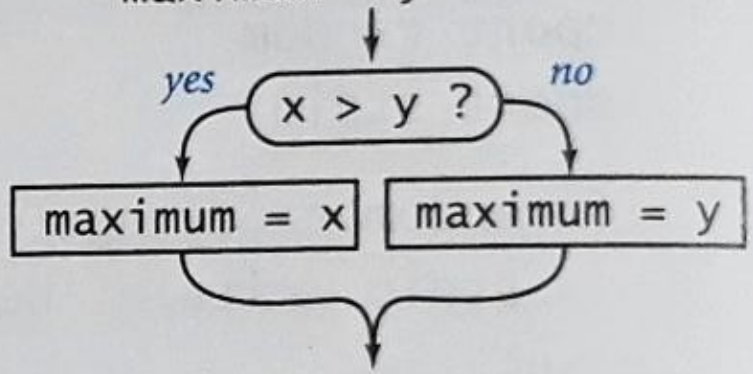
```
1       import stdio
2       import sys
3
4       number = int(sys.argv[1])
5
6       if number % 2 == 0:
7           stdio.writeln('It is a even number')
8       else:
9           stdio.writeln('It is a odd number')
```

$python3 oddeven.py 26
> It is a even number
$python3 oddeven.py 15
> It is a odd number

```
if x < 0:
    x = -x
```




```
if x > y:
    maximum = x
else:
    maximum = y
```



Flowchart examples (if statements)

Common example Programs

absolute value	`if x < 0:` `    x = -x`

put x and y into sorted order	`if x > y:`
	`    temp = x`
	`    x = y`
	`    y = temp`

maximum of x and y	`if x > y: maximum = x` `else:     maximum = y`
error check for remainder operation	`if den == 0: stdio.writeln('Division by zero')` `else:        stdio.writeln('Remainder = ' + num % den)`

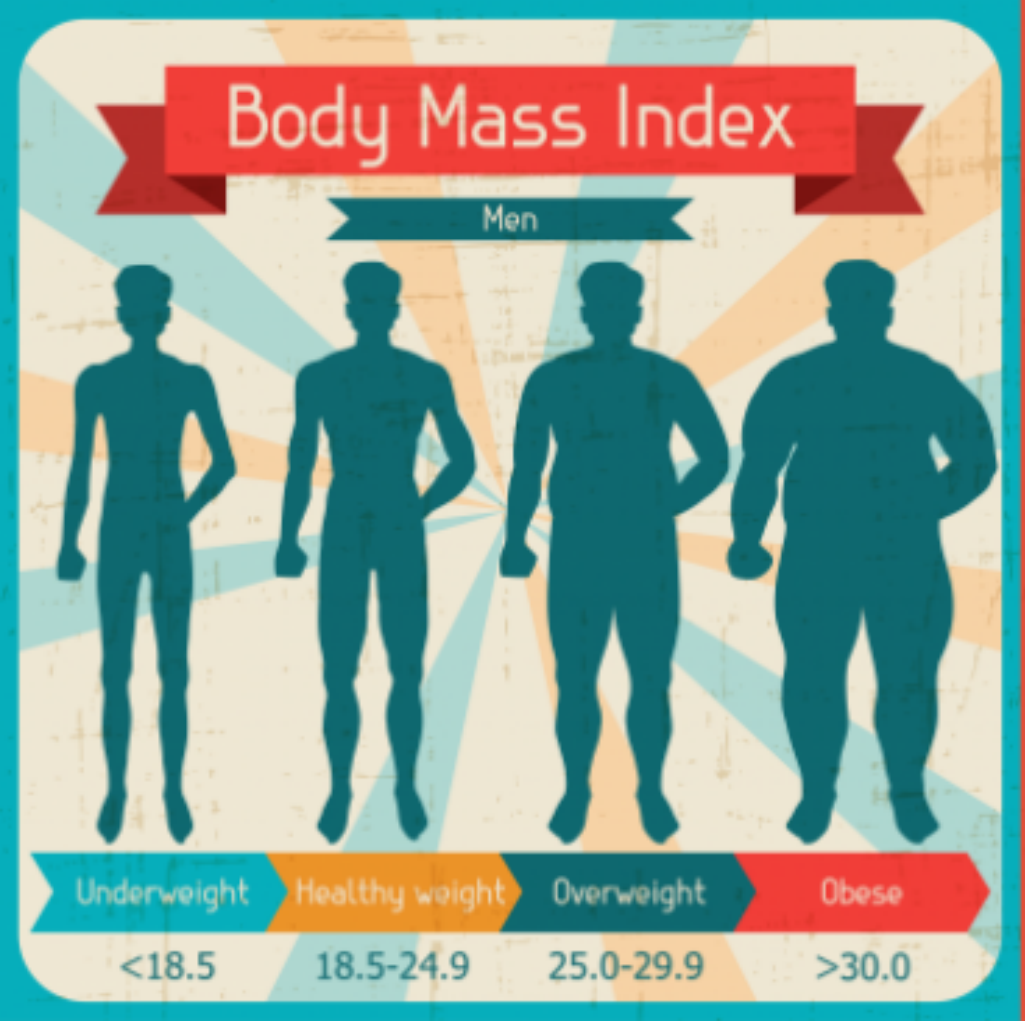
error check for quadratic formula	`discriminant = b*b - 4.0*a*c`
	`if discriminant < 0.0:`
	`    stdio.writeln('No real roots')`
	`else:`
	`    d = math.sqrt(discriminant)`
	`    stdio.writeln((-b + d)/2.0)`
	`    stdio.writeln((-b - d)/2.0)`

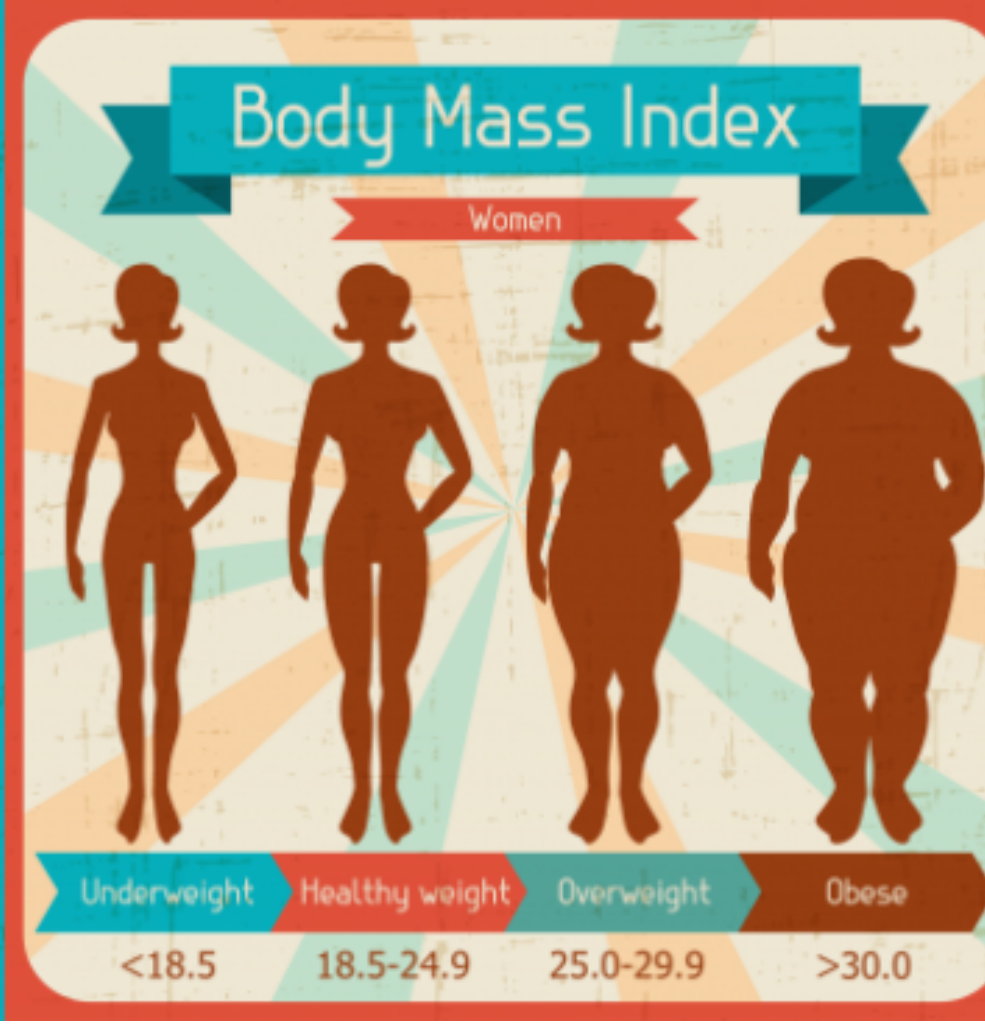
Common example Programs

.

.

.





Body Mass Index

$$BMI = \frac{mass}{height^2}$$

.

.

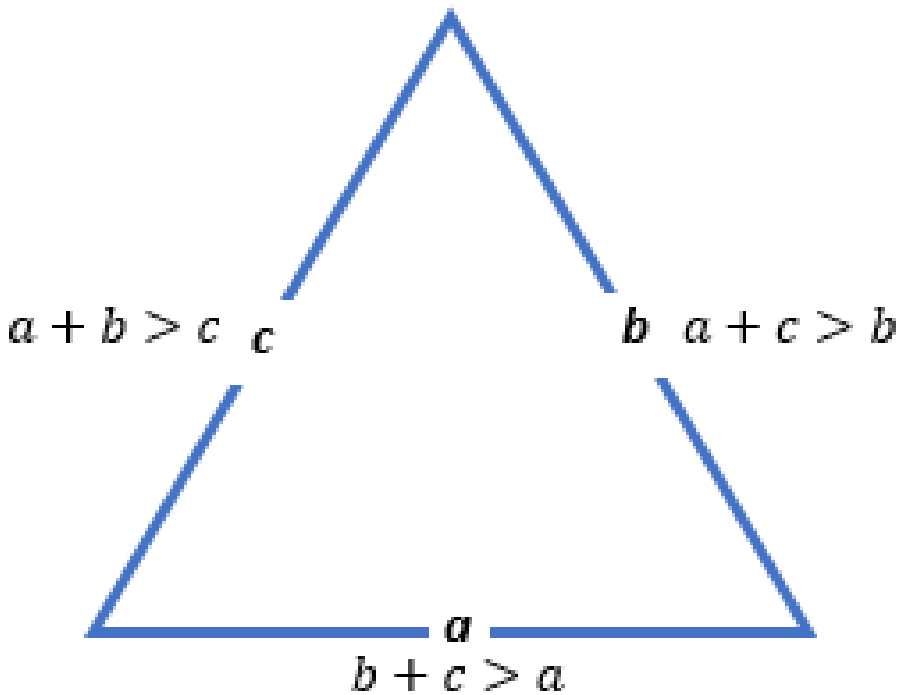
BMI Calculator

```
1  import stdio
2  import sys
3  height = float(sys.argv[1])
4  mass = float(sys.argv[2])
5  BMI = mass / (height**2)
6  if BMI < 18.5:
7      stdio.writeln("Underweight")
8  else:
9      if BMI < 24.9:
10         stdio.writeln("Healthy weight")
11     else:
12         if BMI < 29.9:
13             stdio.writeln("Overweight")
14         else:
15             stdio.writeln("Obese")
```

YOUR CHALLENGE

Your challenge

Given three sides a, b, c. Write a program to check whether the triangle is valid or not. And what type of this triangle? Isosceles, Equilateral, Right or normal.



CONCLUSION

Have a good nice!

Try your best!