

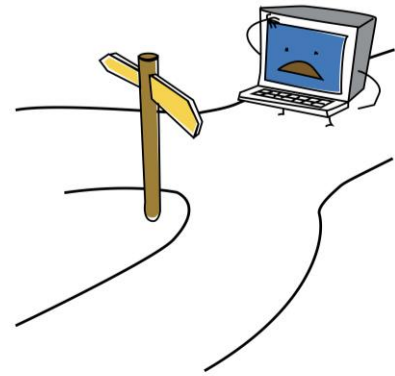
# CSX3001/ITX3001

## FUNDAMENTALS OF COMPUTER PROGRAMMING

### **CLASS 03 CONDITIONAL CONTROL STRUCTURE**

COMPARISON OPERATORS, BOOLEAN VARIABLE AND OPERATORS,  
USING IF-ELSE/IF-ELIF STATEMENTS, NESTED IF STATEMENTS

PYTHON



## BOOLEAN RECAP

**Boolean value** can be either True or False. They are used to represent truth values. For instance, isOpen and isPossible are Boolean variable.

```
isOpen = True  
isPossible = False
```

The common type of Boolean expression is one that compares two values using a *comparison operator*. There are six comparison operators.

Symbol	Definition
==	Is equal to
!=	Is not equal to
>	Is greater than
<	Is less than
>=	Is greater than or equal to
<=	Is less than or equal to

Considering the following code and each line's value on the right column.

```
5 + 7 == 12  
5 != -1
```

```
"Ada" == "Steve"
```

```
23 + 2 >= 7
```

**True**

**True**

**False**

**True**

In addition, Boolean expressions can be joined together using compound Boolean expressions. It's a lot like making compound sentences in English with the words *and* and *or*. In programming, there is a third case: *not*, and we call these words *logical operators*.

Symbol	Definition
and	Logical AND
or	Logical OR
not	Logical NOT

The result of using compound Boolean expressions will be based on the truth logic. The following code shows all possible joined expressions of the truth logic of compound Boolean expressions: *and*, *or*, and *not*.

<code>True and True</code>	<code>True</code>
<code>True and False</code>	<code>False</code>
<code>False and True</code>	<code>False</code>
<code>False and False</code>	<code>False</code>
<code>True or True</code>	<code>True</code>
<code>True or False</code>	<code>True</code>
<code>False or True</code>	<code>True</code>
<code>False or False</code>	<code>False</code>
<code>not True</code>	<code>False</code>
<code>not False</code>	<code>True</code>

Considering the following code and each line's value on the right column.

<code>b1 = True</code>	<code>True</code>
<code>b2 = False</code>	<code>False</code>
<code>b3 = True</code>	<code>True</code>
<code>b4 = False</code>	<code>False</code>
<code>b1 or b2</code>	<code>True</code>
<code>b3 and not (b1 or b3)</code>	<code>False</code>

## CONDITIONAL STATEMENT

Conditional statement in python called the *if* statement. This statement presents the computer with a condition that the computer makes a choice based on.

### IF STATEMENTS

```
if Boolean expression:  
    code block
```

An *if* statement starts with the keyword *if* followed by a *condition*, which is always a **Boolean expression**. The computer checks the condition and executes the code inside the *if* statement if the condition is *true* or skips over that code if the condition is *false*. Let's write some code below that tests whether a kid is tall enough to ride a roller coaster.

```
1 heightToRideAlone = 150.0  
2 userHeight = 152.5  
3  
4 if userHeight >= heightToRideAlone:  
5     print("You are tall enough to ride this roller coaster.")
```

Here, we set 150 cm as the minimum height at which a kid can ride our roller coaster alone, and we set our rider's height to 152.5 cm. At line number 3, we test whether the rider's height is greater than or equal to *heightToRideAlone*. If it is, the program says that they are tall enough to ride the roller coaster.

Again! To write our *if* statement, we put the keyword *if* in front of the condition. We then **indent the code that we want to execute** when that condition is true next line after the colon (:).

1. What happens if we change our rider's height to a number less than 150?

.....

Nothing shows up?...

## ELSE STATEMENTS

Previously, you want to tell the computer to print something if a statement is true. However, what if we want to print something else when the statement is false? To do this,

```
if Boolean expression:
    code block
else:
    code block
```

... after the if statement and block of code, just type the keyword **else** followed by another code block that you want to execute when the **if** condition isn't true.

If the rider isn't tall enough to meet the condition, let's have the computer tell them they can't ride the roller coaster:

```
1 heightToRideAlone = 150.0
2 height = 152.5
3
4 if height >= heightToRideAlone:
5     print("You are tall enough to ride this roller coaster.")
6 else:
7     print("Sorry. You cannot ride this roller coaster.")
```

We could also test different conditions for the rider's height to create more rules. We can do this by adding **elif** conditions. Let's see the example below.

```
1 heightToRideAlone = 150.0
2 heightToRideWithAdult = 140.0
3 height = 152.5
4
5 if height >= heightToRideAlone:
6     print("You are tall enough to ride this roller coaster.")
7 elif height >= heightToRideWithAdult:
8     print("You can ride this roller coaster with an adult.")
9 else:
10    print("Sorry. You cannot ride this roller coaster.")
```

Note that: once any part of an **if** or **elif** statements is found to be **true**, the rest of the conditions will not be checked.

### ♦ IF-ELSE EXERCISES

Complete the following exercises.

2. Write a Python code that takes two input integers and assign them to two variables, namely *numX* and *numY*. Define a conditional if-else statement so that the code prints out only the lowest integer. The example of expected input/output is as follows.

```
Enter first number: 32
Enter second number: 50
The lowest value is 32
```

3. Write a Python code that takes three input integer and store them in three variables, namely *numX*, *numY* and *numZ*. Define a conditional if-elif-else statement so the code prints out the middle value. The expected input/output is as follows.

```
Enter first number: 32
Enter second number: 50
Enter third number: 94
The mid value is 50
```

4. Write a Python code that takes an age as an input integer. If the age is 20 or higher, the code prints "You are eligible to buy cannabis." Otherwise, the code prints "You are too young to consume cannabis." However, if an age is above 70, the code prints "You are too old for cannabis.". The expected input/output is as follows.

```
Enter your age: 32
You are eligible buy cannabis
```

```
Enter your age: 14
You are too young to consume cannabis.
```

```
Enter your age: 70
You are too old for cannabis.
```

5. Write a Python code that takes an integer value as an input. If the value is an odd number, the code prints “Awesome Odd Number”. Otherwise, the code prints “Marvelous Even Number”. The expected input/output as follows.

```
Enter the number: 11
Awesome Odd Number
```

```
Enter the number: 22
Marvelous Even Number
```

6. Write a Python code that takes 3-digits number as an input. If the left-most digit is an odd number, the code prints “I know that the number you entered starts with either 1, 3, 5, 7 or 9.” Otherwise, the code prints “I am sure either 2, 4, 6 or 8 must be a left-most digit”.

```
Enter the 3-digits number: 324
I know that the number you entered starts with
either 1, 3, 5, 7, or 9.
```

```
Enter the 3-digits number: 271
I am sure either 2, 4, 6, or 8 must be a left-most
digit.
```

7. Write a Python code that takes an integer value as an input, if the value is greater than 100 and is odd number, the code prints “WoW”. Otherwise, the code prints “Wonderful”. The expected input/output as follow.

```
Enter the number: 101
Wow
```

8. Create a variable called heroName and assign a string value of "Superman". Then the code takes your name as an input. If a total number of characters of your name is higher than “Superman”, print "You are stronger than Superman." Otherwise, print "You are weaker than an ant." Hint: use *len()* function to determine a string’s length.

Let’s see an example of *len()* function below;

```
name = "Jimmy"
lengthOfString = len(name)
print(lengthOfString) # this will print 5
```

The expected input/output as follow.

```
Enter your name: Jimmy
You are weaker than an ant.
```



9. A regular size of basketball has a volume of 455.9 cubic inches. Write a code to take a radius as an input. Based on the entered value, if the calculated volume is larger than a regular basketball's volume, the code prints "The entered radius is too big.". Otherwise, the code prints "The entered radius is OK." The basketball's volume can be calculated from Sphere formula. The expected input/output is as follows.

```
Enter the radius: 6
The entered radius is too big.
```

```
Enter the radius: 4.5
The entered radius is OK.
```

10. Write a Python code that takes 3-digit integer value as an input. If the sum of each digit is an even number, then print "Sum of all digits is an even number." Otherwise, print "Sum of all digits is not an even number." The expected input/output as follow.

```
Enter the 3-digit number: 325
Sum of all digits is an even number.
```

11. Write a Python code to take a height as an input. Print a message on the screen according to the information shown in the table below. Note that you must use f-strings for your output format.

Lower than 80	Your height is {height} and you are too small for this ride.
80 – 180	You are ok for this ride.
Higher than 180	Your height is {height} are too tall for this ride.

12. Imagine an alien was just shot down in a game. Create a variable called `alienColor` and assign a value of either 'green', 'yellow', or 'red'. Then the code prints the output message based on the information given in the following table.

Color	Score Message
green	You have earned 5 points.
yellow	You have earned 10 points.
red	You have earned 100 points.

Hint: Let's see an example of String comparison below

```
name = "Kasper"
if name == "Kasper":
    print("Yes")
```

13. Stages of life: write a Python code that determines a person's stage of life. Take a value of age as an input, and print the message according the following ranges:

Age (years old)	Message
Less than 2 years old	The person is a baby.
2 – 4 years old	The person is a toddler.
4 – 13 years old	The person is a kid.
13 – 20 years old	The person is a teenager.
20 – 64 years old	The person is an adult.
65 or older	The person is an elder.

## ASSIGNMENTS

Complete the following exercises in Python IDLE or IDE of your choice. You must name the python file as,

`yourid_name_section_class03_as1.py`

for example, for assignment 1 will be named,

`6310001_Harry_541_class03_as1.py`

1. Write a Python code that asks a user for three integers namely, x1, x2, and x3. The program then determines whether x3 is a factor of x1, x2, both x1 and x2, or neither. Examples of inputs/outputs are as follows.

Sample 1

```
Enter x1: 100
Enter x2: 20
Enter x3: 2
x3 is a factor of both x1 and x2.
```

Sample 2

```
Enter x1: 100
Enter x2: 13
Enter x3: 2
x3 is a factor of x1.
```

Sample 3

```
Enter x1: 100
Enter x2: 14
Enter x3: 7
x3 is a factor of x2.
```

Sample 4

```
Enter x1: 23
Enter x2: 15
Enter x3: 7
x3 is neither a factor of x1 nor x2.
```

2. Write a Python code that asks a user for 4 integer numbers as inputs. The code then determines whether the sum of 4 integer numbers is greater than, less than, or equal to 0.
  - 2.1 If the sum value is greater than 0, check whether the sum is odd or even number.
    - 2.1.1 If it is an even number, prints "{number} is Positive Even".
    - 2.1.2 If it is an odd number, prints "{number} is Positive Odd".
  - 2.2 If the value is less than 0, check the number is odd or even.
    - 2.2.1 If it is an even number, prints "{number} is Negative Even".
    - 2.2.2 If it is an odd number, prints "{number} is Negative Odd".
  - 2.3 If the sum is equal to 0, print "0 is Zero".

#### Sample 1

```
Enter the first number: 6
Enter the second number: 9
Enter the third number: 3
Enter the fourth number: 4
#Output
The sum is: 22
22 is Positive Even.
```

#### Sample 2

```
Enter the first number: 6
Enter the second number: -9
Enter the third number: -6
Enter the fourth number: 9
#Output
The sum is: 0
0 is Zero
```

#### Sample 3

```
Enter the first number: 6
Enter the second number: -9
Enter the third number: -6
Enter the fourth number: 0
#Output
The sum is: -9
Negative Odd
```

3. Write a Python code that takes three integer numbers namely, shopping, hour, and minute. The program then calculates a parking fee according to the conditions below;
- 3.1 If an amount of shopping is greater than or equal to 500 baht, the parking is free for the first 2 hours. After that a rate of 30 baht will be charged every 30 minutes.
  - 3.2 If an amount of shopping is less than 500 baht, the parking is free for the first hour. The later hours will be charged 30 baht every 30 minutes.
  - 3.3 Fraction of 30 minutes will be rounded as 30 minutes.
  - 3.4 If the shopping amount is greater than or equal 1500 baht, the customer pays nothing if the parking is less than or equal to 2 hours. Otherwise, the customer pays just 30 baht for the parking.

#### Sample 1

```
Enter shopping amount (in Baht): 700
Enter hours: 2
Enter minutes: 24
The parking fee for the first 2 hours is free. You have
to pay 30 Baht for the parking fee.
```

#### Sample 2

```
Enter shopping: 700
Enter hours: 2
Enter minutes: 54
The parking fee for the first 2 hours is free. You have
to pay 60 Baht for the parking fee.
```

#### Sample 3

```
Enter shopping: 400
Enter hours: 2
Enter minutes: 24
The parking fee for the first hour is free. You have
to pay 90 baht for the parking fee.
```

#### Sample 4

```
Enter shopping: 2000
Enter hours: 6
Enter minutes: 24
The parking fee is 30 baht.
```

4. Write a Python code that takes 5-digits number as an input. If the digit in the hundreds place is greater than the sum of other four digits, the two digits in ones and ten thousands place are swapped. Otherwise, the digits in tens and thousands place are swapped. In the case that the sum of other four digits is equal to the digit in the hundreds place, the code print that 5-digit as an output. Note that if the input is not 5-digit number, the code prints “Wrong Input. Try Again”.

#### Sample 1

```
Enter 5-digits number: 2000
Wrong Input. Try Again.
```

#### Sample 2

```
Enter 5-digits number: 21049
The output is 24019
```

#### Sample 3

```
Enter 5-digits number: 11923
The output is 31921
```

#### Sample 3

```
Enter 5-digits number: 40923
The output is 40923
```