Chapter 3.
Decision Structures and Boolean Logics

Starting out with Python

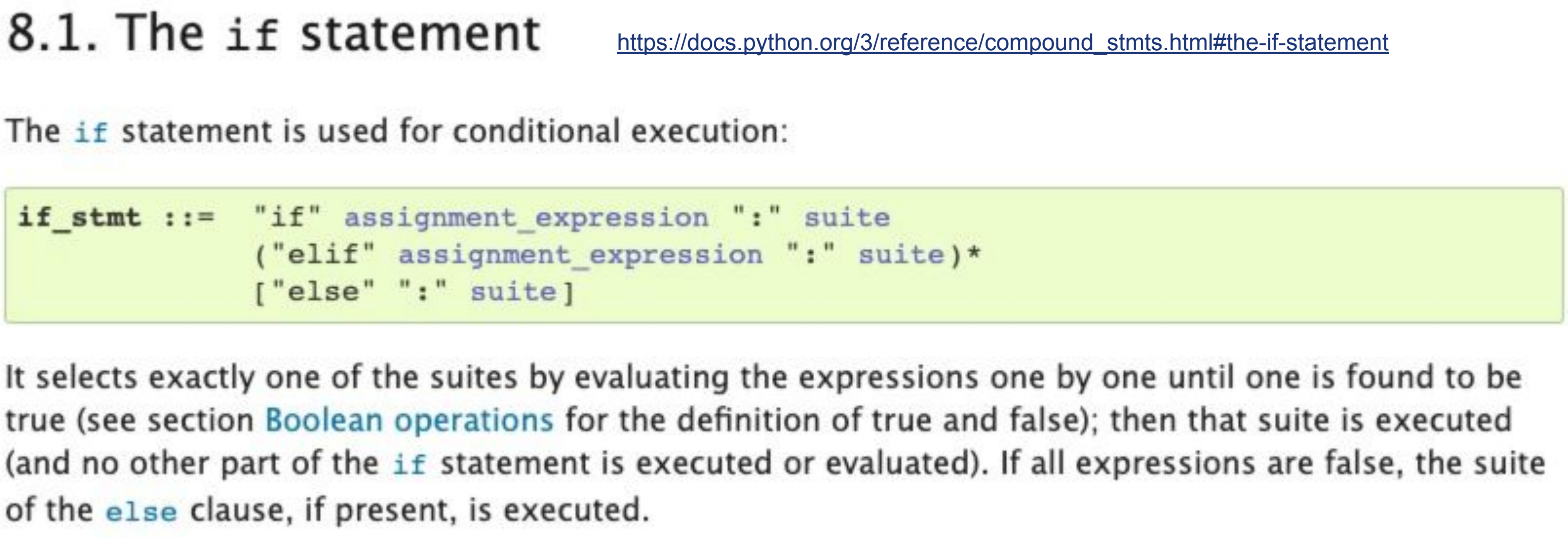
Code Examples with Jupyter Lab

Kyu Lee. Ph. D.

Computer Science

If statement

The if statement



8.1. The if statement

https://docs.python.org/3/reference/compound_stmts.html#the-if-statement

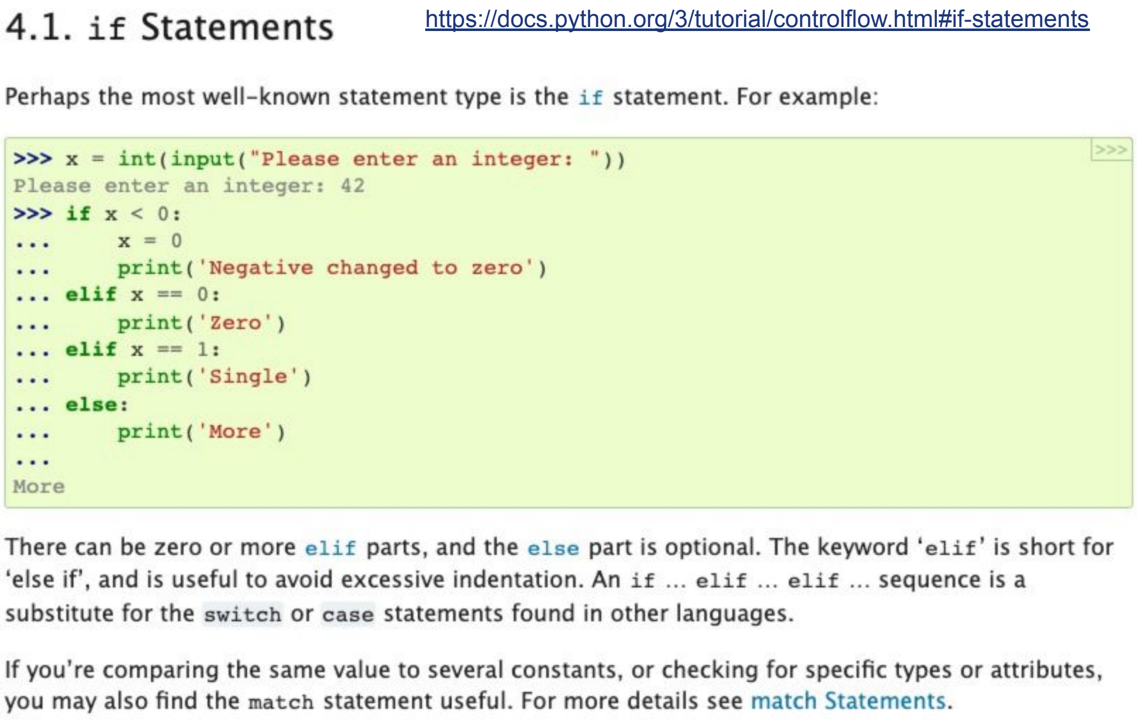
The if statement is used for conditional execution:

```
if_stmt ::=  "if" assignment_expression ":" suite
             ("elif" assignment_expression ":" suite)*
             ["else" ":" suite]
```

It selects exactly one of the suites by evaluating the expressions one by one until one is found to be true (see section Boolean operations for the definition of true and false); then that suite is executed (and no other part of the if statement is executed or evaluated). If all expressions are false, the suite of the else clause, if present, is executed.

The if statement

https://docs.python.org/3/tutorial/controlflow.html#if-statements



4.1. if Statements

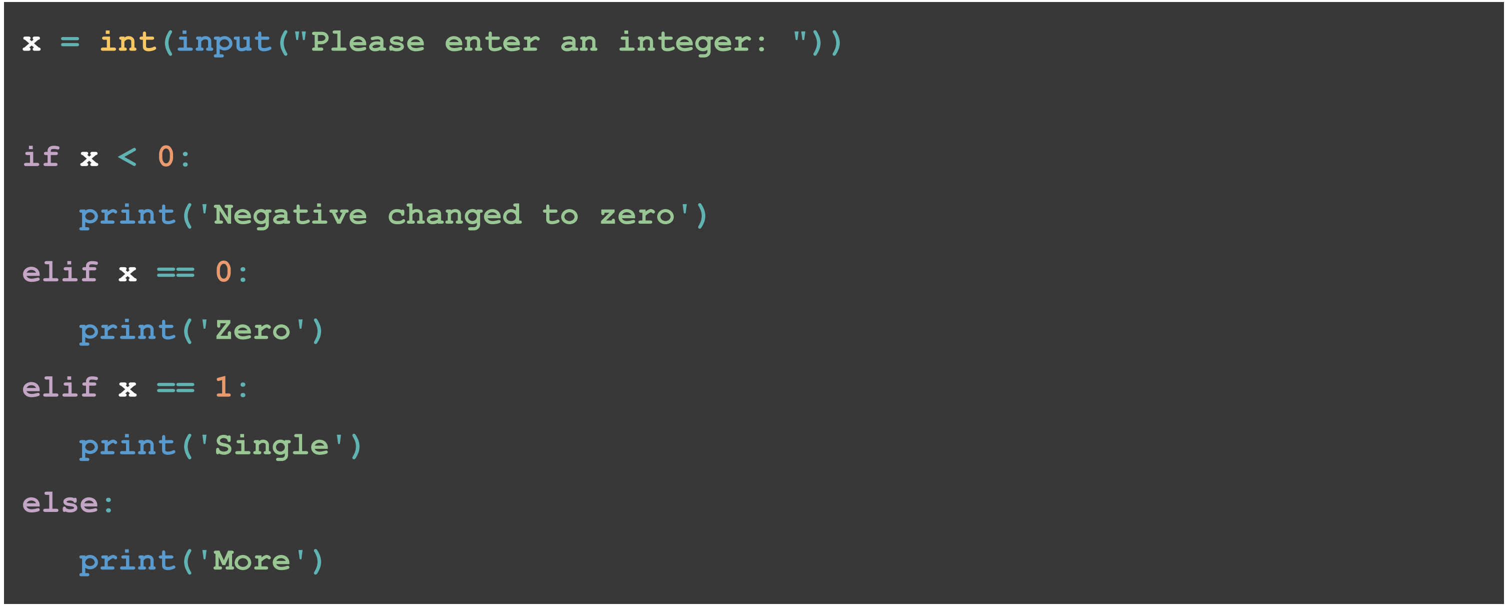
Perhaps the most well-known statement type is the `if` statement. For example:

```
>>> x = int(input("Please enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...     x = 0
...     print('Negative changed to zero')
... elif x == 0:
...     print('Zero')
... elif x == 1:
...     print('Single')
... else:
...     print('More')
...
More
```

There can be zero or more `elif` parts, and the `else` part is optional. The keyword ‘`elif`’ is short for ‘else if’, and is useful to avoid excessive indentation. An `if` … `elif` … `elif` … sequence is a substitute for the `switch` or `case` statements found in other languages.

If you’re comparing the same value to several constants, or checking for specific types or attributes, you may also find the `match` statement useful. For more details see match Statements.

The if statement



```python
x = int(input("Please enter an integer: "))

if x < 0:
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

The if statement

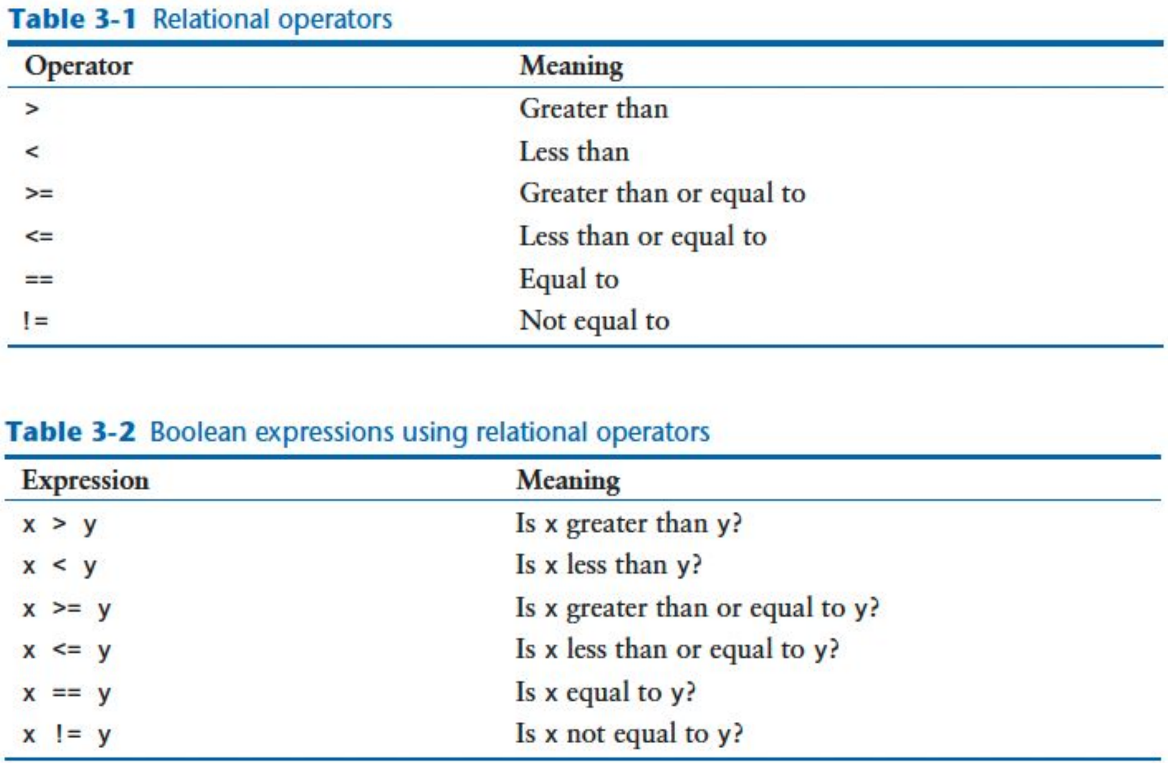


Table 3-1 Relational operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Table 3-2 Boolean expressions using relational operators

Expression	Meaning
x > y	Is x greater than y?
x < y	Is x less than y?
x >= y	Is x greater than or equal to y?
x <= y	Is x less than or equal to y?
x == y	Is x equal to y?
x != y	Is x not equal to y?

Exercises

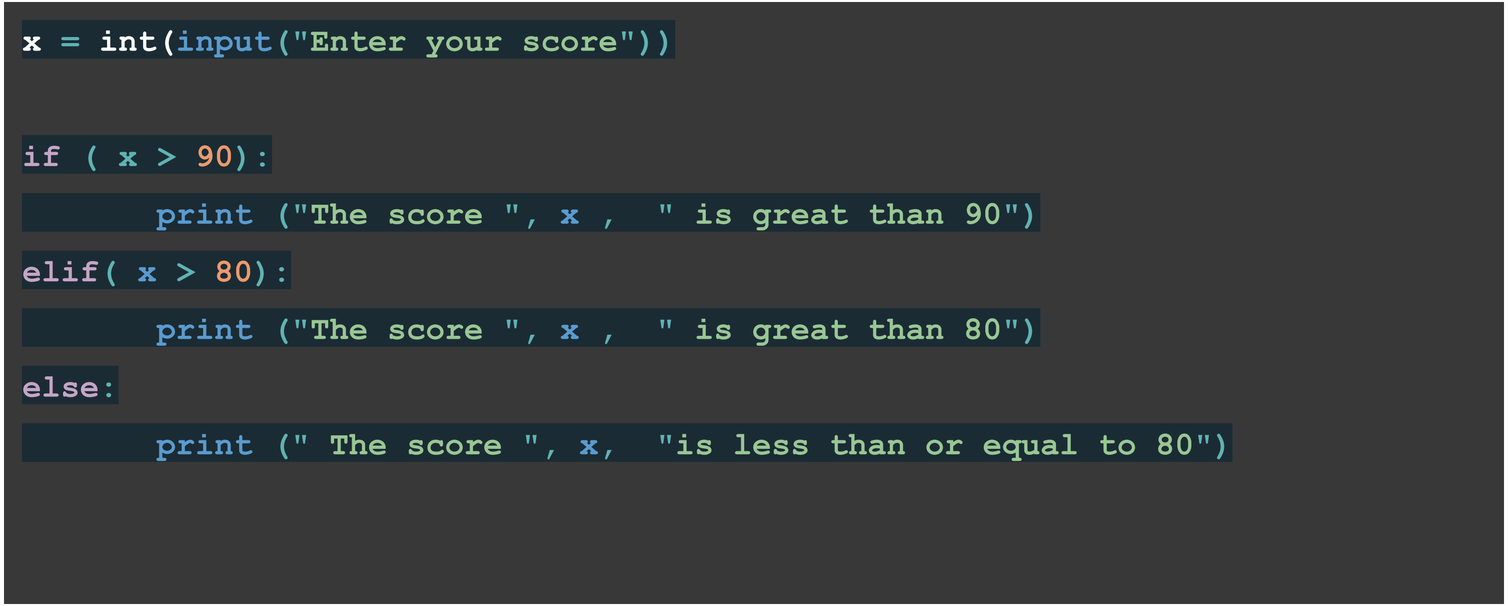
- Exercise 0
    - Ask to user for one number(integer)
    - if the number is odd number
        - Set the variable **`result`** to 1
    - otherwise
        - Set the variable **`result`** to 0

Use the same variable name

Use the modulus operator `%` `to check the even number`

-
    - Run Example
        - Input = 10
            - The result is 0
        - Input = 5
            - The result is 1

The if statement



```
x = int(input("Enter your score"))

if ( x > 90):
       print ("The score ", x ,  " is great than 90")
elif( x > 80):
       print ("The score ", x ,  " is great than 80")
else:
       print (" The score ", x,  "is less than or equal to 80")
```

Compound Condition

Logical Operators

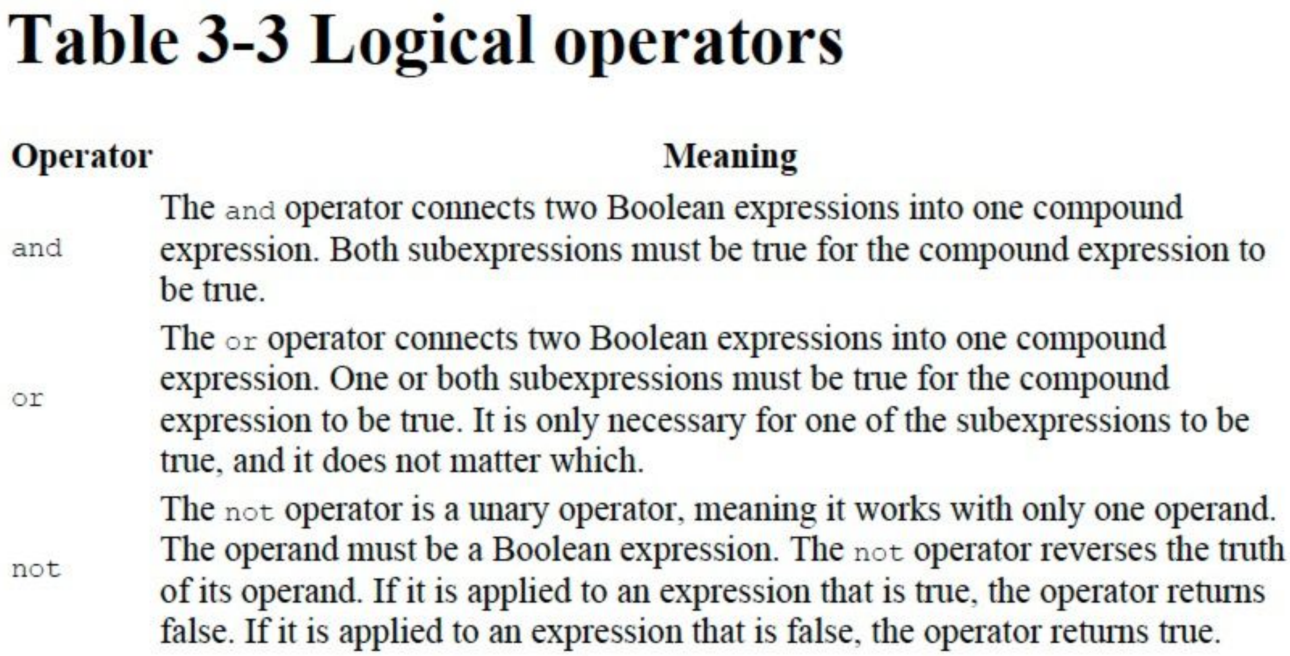
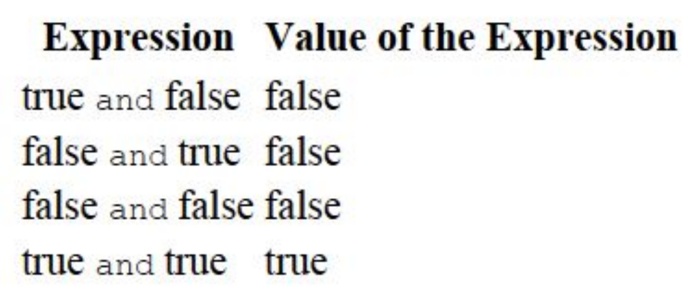


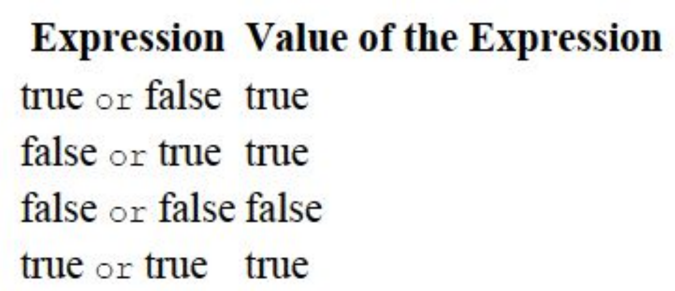
Table 3-3 Logical operators

Operator	Meaning
`and`	The `and` operator connects two Boolean expressions into one compound expression. Both subexpressions must be true for the compound expression to be true.
`or`	The `or` operator connects two Boolean expressions into one compound expression. One or both subexpressions must be true for the compound expression to be true. It is only necessary for one of the subexpressions to be true, and it does not matter which.
`not`	The `not` operator is a unary operator, meaning it works with only one operand. The operand must be a Boolean expression. The `not` operator reverses the truth of its operand. If it is applied to an expression that is true, the operator returns false. If it is applied to an expression that is false, the operator returns true.

Logical Operators



Expression	Value of the Expression
true and false	false
false and true	false
false and false	false
true and true	true



Expression	Value of the Expression
true or false	true
false or true	true
false or false	false
true or true	true

Logical Operators

```
score = int(input("Enter your score"))


if ( score < 0 or score > 100):
       print ("Score is out of range")
```

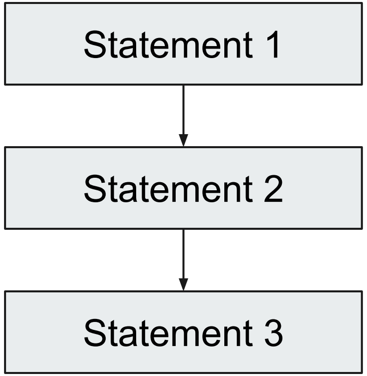
Practical Exercise:

- Take a user input for the score
- Check the score whether it is between 0 and 100
- Print the decision result
    - between 0 and 100 or not

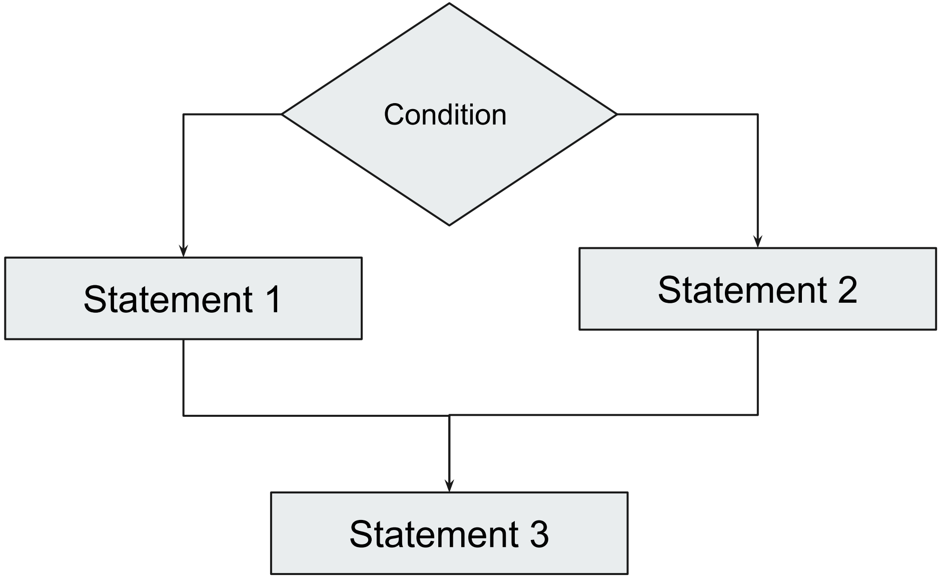
Nested If Structure

Flow charts of if statement

Sequential Flow

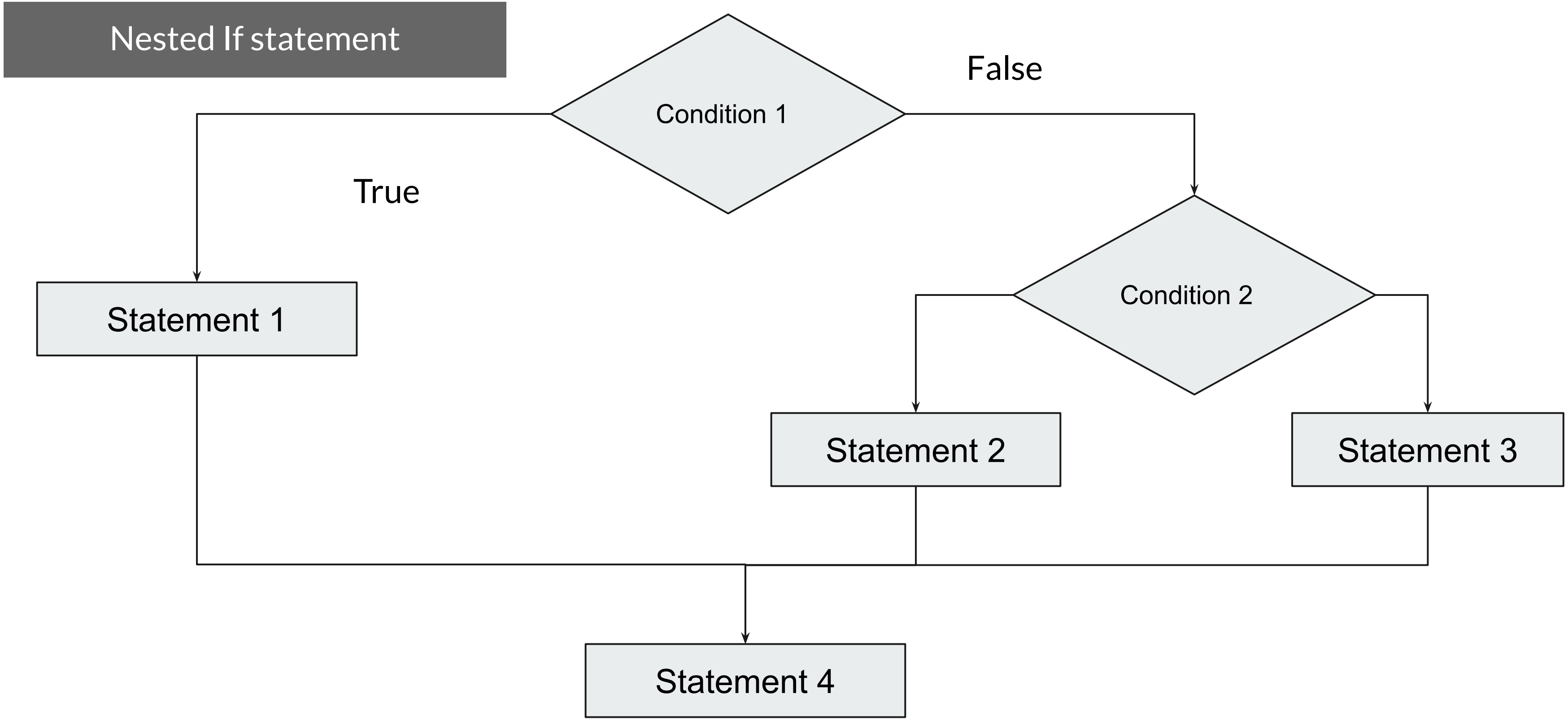


Decision Structure(Branch)

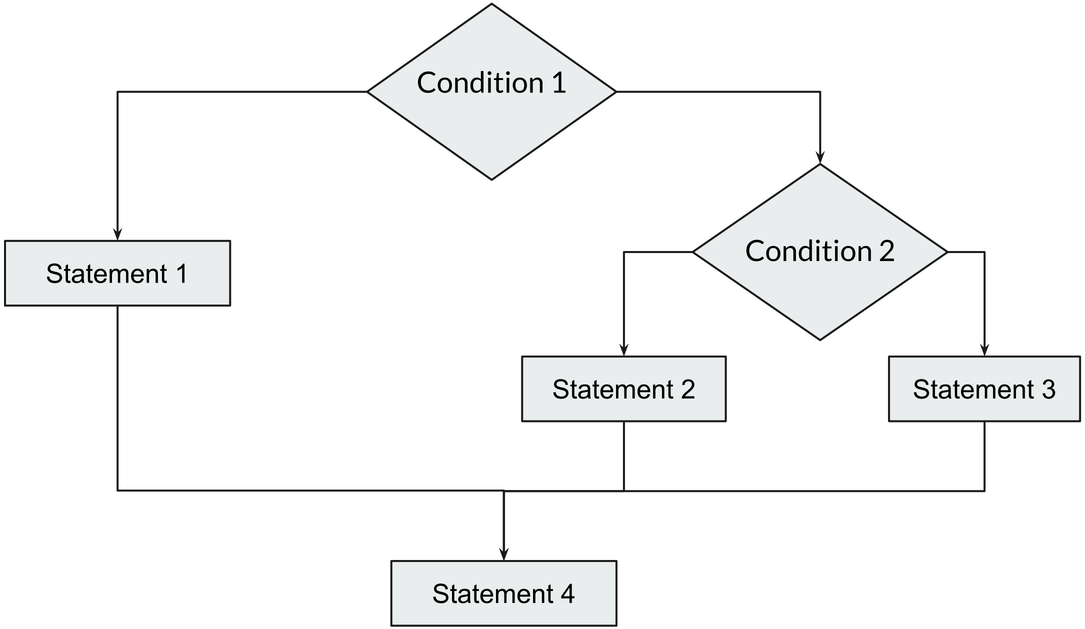


Flow charts of nested-if statement

Nested If statement



Nested-if statement



```
if (condition1):
       statement1
else:
       if ( condition2):
               statement2
       else:
               statement3
statement4
```

Try to convert this structure to if-elif-else

Truth Table for Logical operator (and, or)

And	T	F
T	T	F
F	F	F

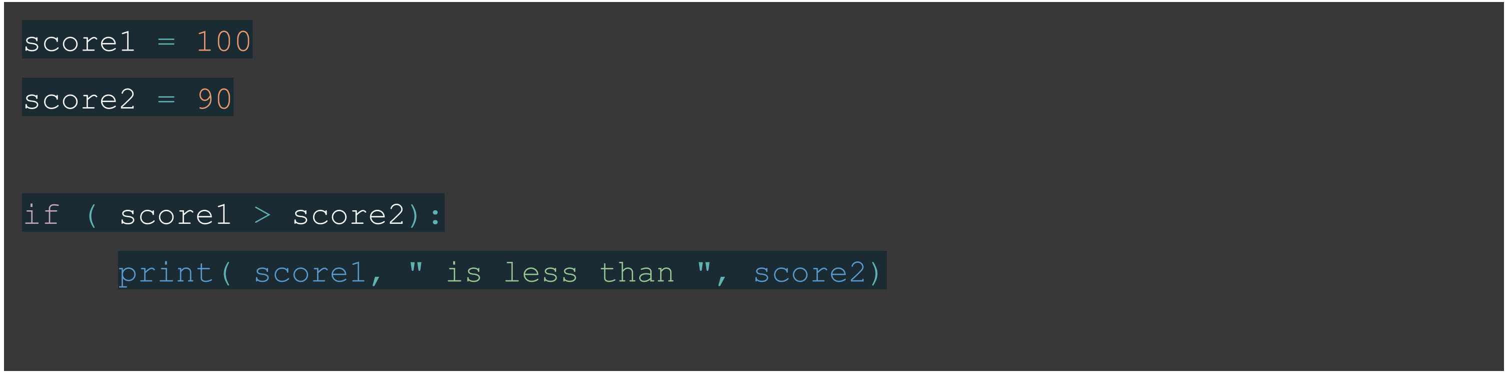
Or	T	F
T	T	T
F	T	F

cond1 and cond2

X = 50
X > 0 and X < 25

Short Hand if

Short Hand if



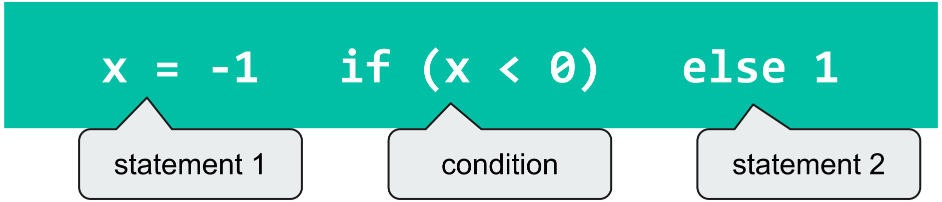
```
score1 = 100
score2 = 90

if ( score1 > score2):
    print( score1, " is less than ", score2)
```

Short Hand if - else

statement 1 **if** condition **else** statement 2

```
if ( x < 0):
      x = -1;
else:
      x = 1
```



```
x = -1   if (x < 0)   else 1
```

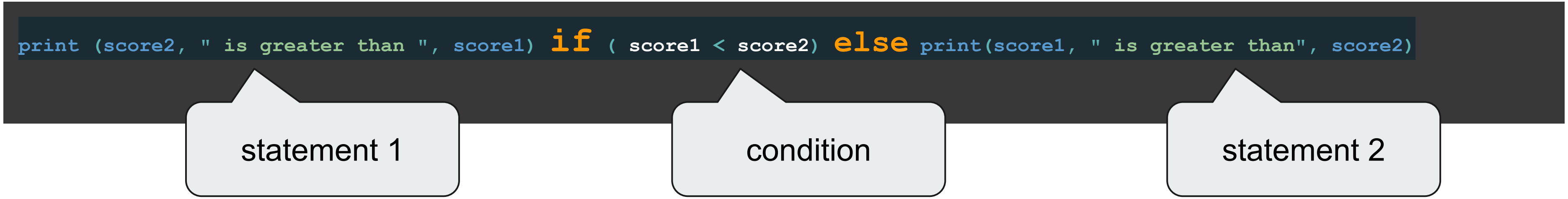
statement 1

condition

statement 2

Short-Hand if - else

statement 1 if condition else statement 2



```
print (score2, " is greater than ", score1) if ( score1 < score2) else print(score1, " is greater than", score2)
```

statement 1

condition

statement 2

Exercises

- Exercise 1
  - Write a program that prompts the user to enter a number. Based on the entered **number**, assign a value to the variable "**range**" as follows:
    - If the number is less than 50, set the variable "**range**" to 1.
    - If the number is greater than or equal to 50 and less than 100, set the variable "**range**" to 2.
    - If the number is greater than or equal to 100, set the variable "**range**" to 3.
    - Ensure that the program uses only two comparisons with the use of the "else" keyword.
  - 3 cases
    - input < 50
    - 50 <= input < 100
    - input >= 100
  - Run Example
    - Input = 40
      - range = 1
    - Input = 70
      - range = 2
    - Input = 120
      - range = 3

Use the variable name "number"

Use the least number of if-conditions.
(2 comparisons with **`else`**)

Exercises

- **Exercise 2: Score Grading**
  - Ask for an input(score) to User
  - Decide the ‘letter’ grade based on the following grading policy
    - `grade` F: score < 60
    - grade D: 60 <= score < 70
    - grade C: 70 <= score < 80
    - grade B: 80 <= score < 90
    - grade A: 90 <= score < 100
  - Print the decided the grade and score
    - Example
      - Input 85
        - Grade B
      - Input 95
        - Grade A
  - Make the flow chart first
    - https://draw.io

Use the variable name `grade` `for the answer`

Use the least number of if-conditions.
(4 `<` operators with `else` are enough)

Exercises

- **Exercise 3: Find the greatest number among three user inputs**
  - **Do not use max() function. Make your own algorithm to find max value.**

  - Take the three integer values and save them as **num1, num2 and num3**
  - Find the **greatest number** among three inputs
    - `maxnum`
  - Run Example
    - [Input]
      - 10
      - 97
      - 45
    - [Output]
      - The greatest number is 97
  - Do 'commit' frequently whenever you test your program

Use the variable name `maxnum`

Please separate each input with a new line.

Do not use the standard library like max( )
Make if-statement to practice

Exercises

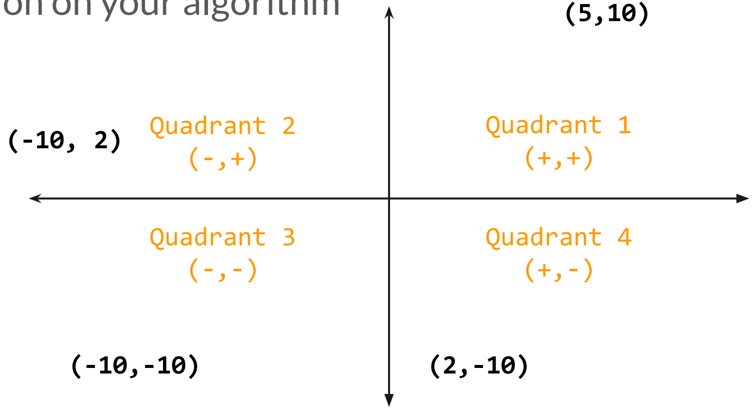
- **Exercise 4: Decide the quadrant number in the coordinate plane.**
  - Take two integers (x, y) for the coordinate values
  - Determine the points (x,y) in which **`quadrant`**
    - **`quadrant = 1,2,3 or 4`**
  - Submit your flow chart and elaboration on your algorithm
  - Make the flow chart
    - https://draw.io

Use the variable name **`quadrant`**

```
[Input]
        -10
        2
[Output]
        Quadrant 2
```

```
[Input]
        2
        -10
[Output]
        Quadrant 4
```

Please separate each input with a new line.



Exercises

- **Exercise 5: Check the same values**
  - Ask users three integer values
  - Check all three values are same or not
    - if all values are the same,
      - `duplication` = 3
    - if two values are the same,
      - `duplication` = 2
    - if all values are distinct
      - `duplication` = 0
  - Try to make a program code with the **least** number of if-condition
    - Submit your flow chart and elaboration on your algorithm
    - Make the flow chart first
      - https://draw.io
    - To practice the if-else statement, **<u>do not use</u>** any standard library functions

Use the same variable name `duplication`

Please separate each input with a new line.

```
[Input]
        10
        2
        2
[Output]
        duplication 2
```

```
[Input]
        2
        2
        2
[Output]
        duplication 3
```

```
[Input]
        1
        2
        3
[Output]
        duplication 0
```

Kyuwoong Lee, Ph. D

Algorithm Development

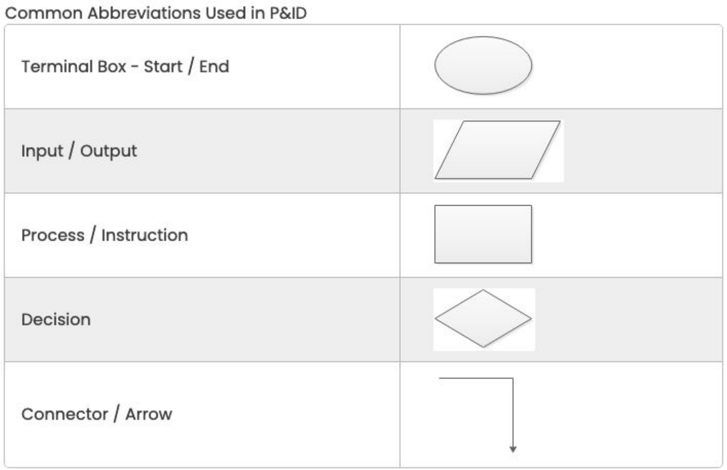
Introduction to Python Programming

1. drawio

- https://draw.io
  - Sign Up
  - Try to draw any shape and connect it with lines and connector

2. What is flowchart ?

- Flowchart
    - A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows to demonstrate a process or a program. With algorithms, we can easily understand a program. The main purpose of using a flowchart is to analyze different methods. Several standard symbols are applied in a flowchart (https://www.edrawsoft.com/explain-algorithm-flowchart.html)
- Basic Symbols



Common Abbreviations Used in P&ID

Terminal Box - Start / End	⬭
Input / Output	▱
Process / Instruction	▭
Decision	◇
Connector / Arrow	↴

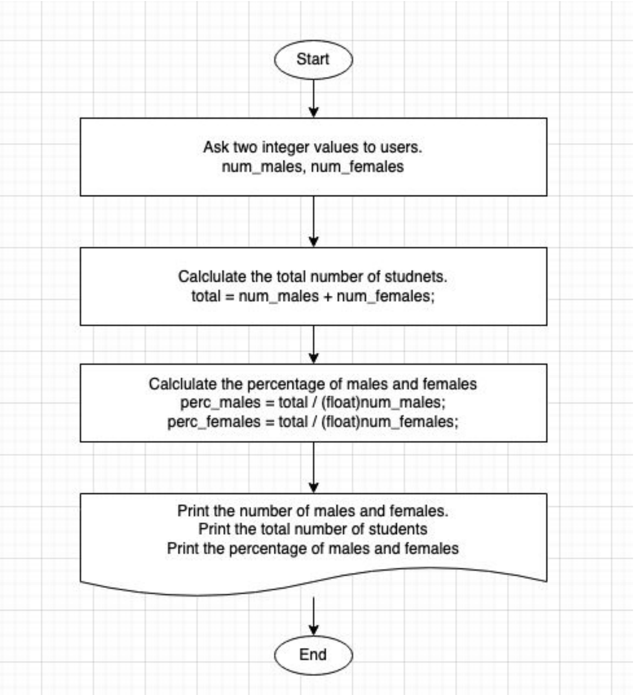
How to draw flowchart

- **Rules for Creating Flowchart**

  (https://www.geeksforgeeks.org/an-introduction-to-flowcharts/)

- A flowchart is a graphical representation of an algorithm.it should follow some rules while creating a flowchart

  Rule 1: Flowchart opening statement must be 'start' keyword.
  Rule 2: Flowchart ending statement must be 'end' keyword.
  Rule 3: All symbols in the flowchart must be connected with an arrow line.
  Rule 4: The decision symbol in the flowchart is associated with the arrow line.



Start

Ask two integer values to users.
num_males, num_females

Calclulate the total number of studnets.
total = num_males + num_females;

Calclulate the percentage of males and females
perc_males = total / (float)num_males;
perc_females = total / (float)num_females;

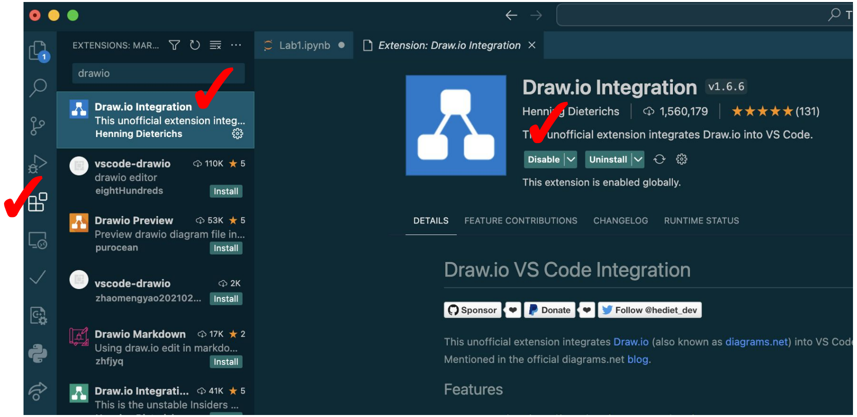
Print the number of males and females.
Print the total number of students
Print the percentage of males and females

End

Drawio in VS Code

- Open VS Code

1. Go to Package
2. Find the package “drawio”
3. Install the package “drawio”
4. Create a file “lab1.drawio”
5. Draw the flowchart
6. Export it as a PNG
7. Submit it



drawio

Draw.io Integration
This unofficial extension integ...
Henning Dieterichs

vscode-drawio 110K ★ 5
drawio editor
eightHundreds
Install

Drawio Preview 53K ★ 5
Preview drawio diagram file in...
purocean
Install

vscode-drawio 2K
zhaomengyao202102...
Install

Drawio Markdown 17K ★ 2
Using draw.io edit in markdo...
zhfjyq
Install

Draw.io Integrati... 41K ★ 5
This is the unstable Insiders ...

Draw.io Integration v1.6.6
Henning Dieterichs | 1,560,179 | ★★★★★ (131)
This unofficial extension integrates Draw.io into VS Code.
Disable | Uninstall
This extension is enabled globally.

Draw.io VS Code Integration

This unofficial extension integrates Draw.io (also known as diagrams.net) into VS Cod
Mentioned in the official diagrams.net blog.

Features

Algorithm Development 1 : Find min value

- There is an integer number in the variables num1, num2, and num3 respectively. You can compare only two numbers at a time. Show your algorithm to find the **least** number.
- No need to use programming syntax

- Show the all the detail steps to develop your algorithm
  - Pseudo-code can be used to explain the algorithm
  - Draw the flowchart (**draw.io**) to show your algorithm

Algorithm Development 2 : Print numbers in order

- There is an integer number in the variables num1, num2, and num3 respectively. You can **compare only two numbers** at a time.
  - e.g, num1 < num2
- Show your algorithm to the least number, greatest number and other
  - print the least, other and greatest number

- Show the all the detail steps to develop your algorithm
  - Pseudo-code can be used to explain the algorithm
  - Draw the flowchart (**draw.io**) to show your algorithm

Assignments

Introduction to Python Programming

Assignment 3-1

- **Find the smallest number among three integer values**
  - Generate three random integer values between 0 and 100
    - `import random`
    - `number1 = random.randint(0,100)`
    - `number2 = random.randint(0,100)`
    - `number3 = random.randint(0,100)`
  - Print all random numbers
  - Find the smallest number among random values
    - `min_value`
    - Do **not** use any standard module/functions(min ( ))
    - **Develop your algorithm** to find the smallest number
  - Try to make a program code with the **least** number of if-condition
    - Submit your flow chart and elaboration on your algorithm
    - Make the flow chart first
      - https://draw.io
    - To practice the if-else statement, do not use any standard library or data structures

Use the same variable name `min_value`

```
[Input]
[Output]
          5 87 43
          5
```

```
[Input]
[Output]
          95 10 25
          10
```

Assignment 3-2

- **Numbers in ascending order.**
    - Ask for three integer values to user
        - `num1, num2, num3`
    - Find the minimum, maximum and other value and set to
        - **`minval, median, maxval`**
    - Run Example
        - The input numbers:
            - 45
            - 97
            - 10
        - Output:
            - 10 45 97

```
[Input]
        10
        2
        1
[Output]
        1 2 10
```

```
[Input]
        2
        3
        1
[Output]
        1 2 3
```

Please separate each input with a new line.

Use the same variable names **`minval, median, maxval`**

-
    - Do **NOT** use any standard functions and libraries(Sort(), min()). Use only **if statement.**
    - Submit your flow chart and elaboration on your algorithm
        - use the online flowchart tool
            - https://draw.io

Assignment 3-3

- Email validation program
  - Ask for the **email string** to user
  - Determine the email string is valid or not. Set the **boolean variable** **`result`** to True or False
    - **`result`**
  - Validate Condition
    - It must start with the alphabet character (‘a~z’, ‘A~z’)
      - isalpha() ; to check the alphabet or not
    - The email string **length** is greater than 5 and less than 30
      - use the **len**() function from String
    - It must include the letter ‘@’
    - It must include at least one ‘.’ after ‘@’
      - find(), len(), index()
  - Input
    - Email string
  - Output
    - True or False

Use the same variable name **`result`**

```
[Input]
        john@company.com
[Output]
        True

[Input]
        2john@company.com
[Output]
        False

[Input]
        john@company#com
[Output]
        False
```