



The Islamic University of Gaza
Faculty of Engineering
Computer Engineering Department



Introduction to Computers

LAB #4

Selection

Eng. Hashem M. Hejazy

Eng. Mohammed O. AlHabbash

Fall 2026

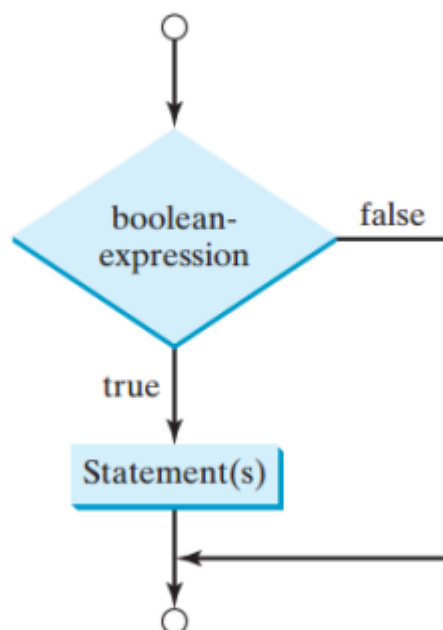
Introduction

We have already learned how to calculate the area of a circle and learned how to declare a variable and constant. But, what if the user enters a negative number for the radius? Is it possible?

Actually, This is something that cannot happen. So we need to check the radius before completing the calculation.

In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. Conditional statements give us this ability. The simplest form is the **if statement**:

```
if boolean-expression:  
    statement(s) # Note that the statement(s) must be indented
```



Notes

- The boolean expression after if is called the condition. If it is true, the indented statement runs. If not, nothing happens.
- There is no limit on the number of statements that can appear in the body, but there has to be at least one.

Boolean Types, Values, and Expressions

Booleans are either true or false. Python has two constants, cleverly named True and False, which can be used to assign boolean values directly. Expressions can also evaluate to a boolean value. As follow:

```
isTrue=True  
isFalse=False
```

How do you compare two values, such as whether a radius is greater than 0, equal to 0, or less than 0? Python provides six comparison operators (also known as relational operators).

Operator	Name	Example (radius =5)	Result
<	Less Than	radius < 0	False
<=	Less than or equal to	radius <= 0	False
>	greater than	radius > 0	True
>=	greater than or equal to	radius >= 0	True
==	equal to	radius == 0	False
!=	not equal to	radius != 0	True

Notes:

- Python uses 1 to represent True and 0 for False. You can use the int function to convert a Boolean value to an integer.

```
print(int(True))  
print(int(False))
```

Output:

```
1  
0
```

- You can also use the bool function to convert a numeric value to a Boolean value. The function returns False if the value is 0; otherwise, it always returns True.

```
print(bool(0))  
print(bool(4))
```

Output:

```
False  
True
```

if Statements

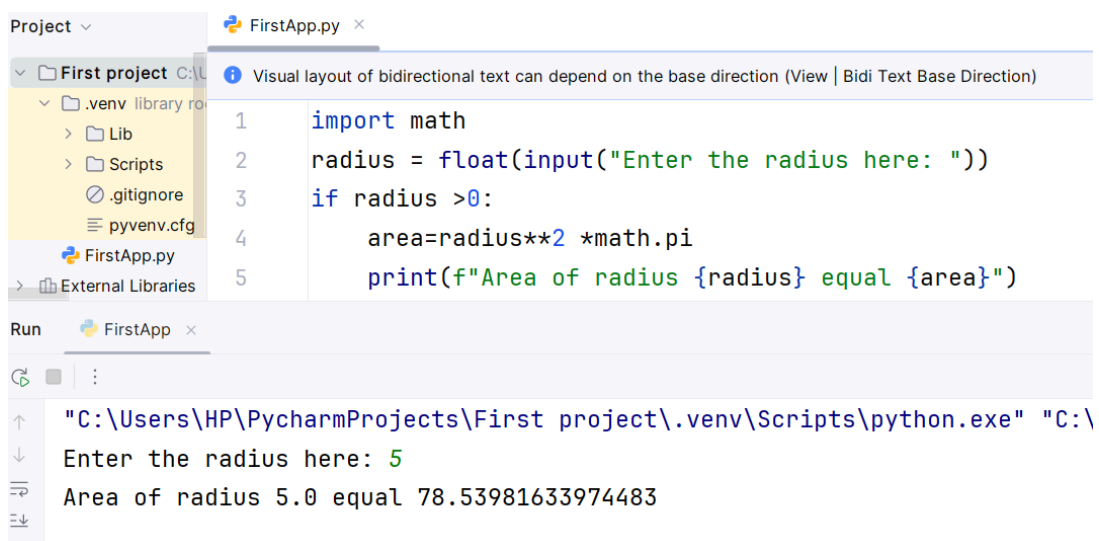
Python has several types of selection statements: one-way if statements, two-way if-else statements, nested if statements, multi-way if-elif-else statements, and conditional expressions.

A one-way if statement executes an action if and only if the condition is true. The syntax for a one-way if statement is:

```
if boolean-expression:  
    statement(s) # Note that the statement(s) must be indented
```

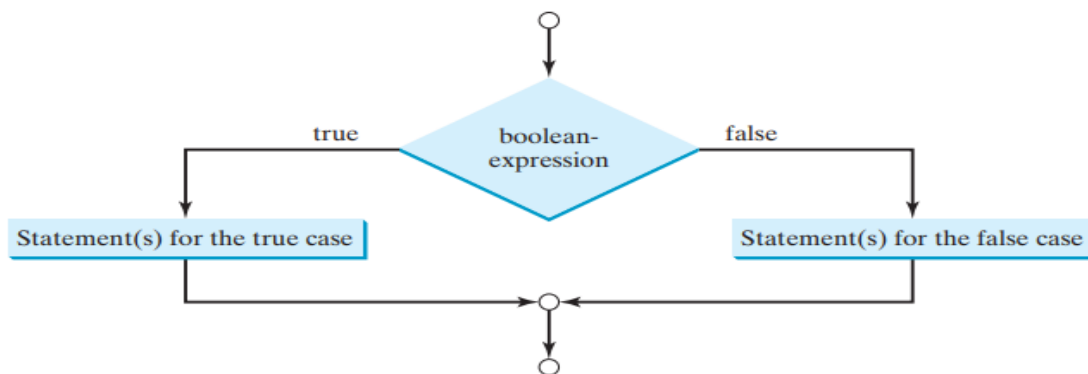
The statement(s) must be indented at least one space to the right of the if keyword and each statement must be indented using the same number of spaces. For consistency.

EX #1: Calculating the area of a circle



```
Project ▾ FirstApp.py ×  
Visual layout of bidirectional text can depend on the base direction (View | Bidi Text Base Direction)  
First project C:\...  
  .venv library root  
    > Lib  
    > Scripts  
    .gitignore  
    pyvenv.cfg  
  FirstApp.py  
  External Libraries  
Run FirstApp ×  
C:\Users\HP\PycharmProjects\First project\.venv\Scripts\python.exe "C:\...  
Enter the radius here: 5  
Area of radius 5.0 equal 78.53981633974483
```

Two-Way if-else Statements



A one-way if statement takes an action if the specified condition is True. If the condition is False, nothing is done. But what if you want to take one or more alternative actions when the condition is False? You can use a two-way if-else statement. The actions that a two-way if-else statement specifies differ based on whether the condition is True or False. Here is the syntax for a two-way if-else statement:

```
if boolean-expression:
    statement(s)-for-the-true-case
else:
    statement(s)-for-the-false-case
```

EX #2

```
1 import math
2 radius = float(input("Enter the radius here: "))
3 if radius > 0:
4     area = radius**2 * math.pi
5     print(f"Area of radius {radius} equal {area}")
6 else:
7     print(f"{radius} is invalid input")
```

Run FirstApp

```
"C:\Users\HP\PycharmProjects\First project\.venv\Scripts\python.exe" "C:\U
Enter the radius here: -3
-3.0 is invalid input"
```

EX #3: Guessing Birthday

```

day = 0 # birth day to be determined

# Prompt the user to answer the first question
question1 = "Is your birthday in Set1?\n" + \
    " 1  3  5  7\n" + \
    " 9 11 13 15\n" + \
    "17 19 21 23\n" + \
    "25 27 29 31" + \
    "\nEnter 0 for No and 1 for Yes: "
answer = eval(input(question1))

if answer == 1:
    day += 1

# Prompt the user to answer the second question
question2 = "Is your birthday in Set2?\n" + \
    " 2  3  6  7\n" + \
    "10 11 14 15\n" + \
    "18 19 22 23\n" + \
    "26 27 30 31" + \
    "\nEnter 0 for No and 1 for Yes: "
answer = eval(input(question2))

if answer == 1:
    day += 2

# Prompt the user to answer the third question
question3 = "Is your birthday in Set3?\n" + \
    " 4  5  6  7\n" + \
    "12 13 14 15\n" + \
    "20 21 22 23\n" + \
    "28 29 30 31" + \
    "\nEnter 0 for No and 1 for Yes: "

answer = eval(input(question3))

if answer == 1:
    day += 4

# Prompt the user to answer the fourth question
question4 = "Is your birthday in Set4?\n" + \
    " 8  9 10 11\n" + \
    "12 13 14 15\n" + \
    "24 25 26 27\n" + \
    "28 29 30 31" + \
    "\nEnter 0 for No and 1 for Yes: "
answer = eval(input(question4))

if answer == 1:
    day += 8

# Prompt the user to answer the fifth question
question5 = "Is your birthday in Set5?\n" + \
    "16 17 18 19\n" + \
    "20 21 22 23\n" + \
    "24 25 26 27\n" + \
    "28 29 30 31" + \
    "\nEnter 0 for No and 1 for Yes: "
answer = eval(input(question5))

if answer == 1:
    day += 16

print("\nYour birthday is " + str(day) + "!")

```

EX #4: SubtractionQuiz

First project C:\

.venv library root

Lib

Scripts

.gitignore

pyenv.cfg

FirstApp.py

External Libraries

Scratches and Console

1

import random

2

1. Generate two random single-digit integers

3

number1 = random.randint(a: 0, b: 9)

4

number2 = random.randint(a: 0, b: 9)

5

2. If number1 < number2, swap number1 with number2

6

if number1 < number2:

7

number1, number2 = number2, number1 # Simultaneous assignment

8

4. Prompt the student to answer "what is number1 - number2?"

9

answer = eval(input("What is " + str(number1) + " - " +

10

str(number2) + "? "))

11

4. Grade the answer and display the result

12

if number1 - number2 == answer:

13

print("You are correct!")

14

else:

15

print("Your answer is wrong.\n", number1, "-",

16

number2, "is", number1 - number2)

Run

SubtractionQuiz x

↑

↓

≡

>

"C:\Users\HP\PycharmProjects\First project\.venv\Scripts\python.exe" C:\Users\HP\AppData

What is 4 - 1? 2

Your answer is wrong.

4 - 1 is 3

Nested if and Multi-Way if-elif-else Statements

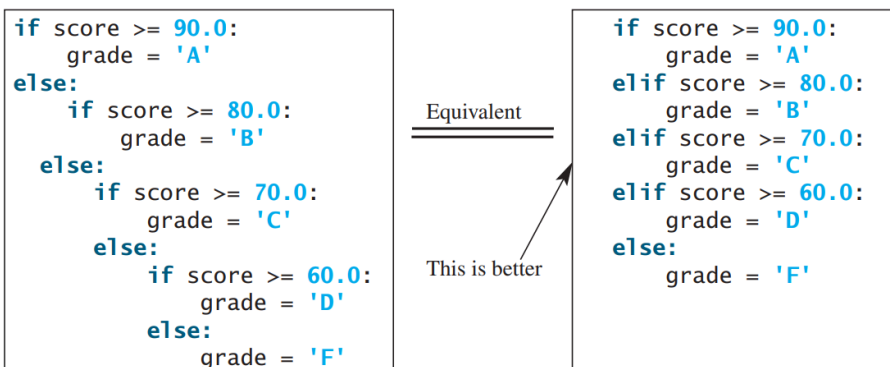
The statement in an if or if-else statement can be any legal Python statement, including another if or if-else statement. The inner if statement is said to be nested inside the outer if statement. The inner if statement can contain another if statement; in fact, there is no limit to the depth of the nesting. For example, the following is a nested if statement:

```
if i > k:
    if j > k:
        print("i and j are greater than k")
else:
    print("i is less than or equal to k")
```

The if-elif-else statement allows you to check multiple expressions for True and execute a block of code as soon as one of the conditions evaluates to True. The syntax for multi-way if-elif-else statement:

```
if boolean-expression1:
    statement(1)
elif boolean-expression2:
    statement(2)
elif boolean-expression3:
    statement(3)
else:
    statement(4)
```

Nested if and Multi-Way if-elif-else Statements is similar, So you can use any of them to implement your code. For example:



Logical Operators

Sometimes, a combination of several conditions determines whether a statement is executed. You can use logical operators to combine these conditions to form a compound expression.

Operator	Name	Description
not	logical negation	It just reverse the state
and	logical conjunction	when both operands are true then condition becomes true
or	logical disjunction	when one of operands are true then condition becomes true

Example:

```
if i > k:
    if j > k:
        print("i and j are greater than k")
else:
    print("i is less than or equal to k")
```

Can write like this:

```
if i > k and j > k:
    print("i and j are greater than k")
else:
    print("i is less than or equal to k")
```

Conditional Expressions

You might want to assign a value to a variable that is restricted by certain conditions. For example, the following statement assigns 1 to y if x is greater than 0, and -1 to y if x is less than or equal to 0.

```
if x > 0:
    y = 1
else:
    y = -1
```

You can use a conditional expression to achieve the same result.

```
y = 1 if x > 0 else -1
```

Conditional expressions are in a completely different style. The syntax is:

```
expression1 if boolean-expression else expression2
```

The result of this conditional expression is **expression1** if **boolean-expression** is true; otherwise, the result is **expression2**.

EX #4: Compute BMI

```
# Prompt the user to enter weight in pounds
weight = eval(input("Enter weight in pounds: "))

# Prompt the user to enter height in inches
height = eval(input("Enter height in inches: "))

KILOGRAMS_PER_POUND = 0.45359237 # Constant
METERS_PER_INCH = 0.0254 # Constant

# Compute BMI
weightInKilograms = weight * KILOGRAMS_PER_POUND
heightInMeters = height * METERS_PER_INCH
bmi = weightInKilograms / (heightInMeters *
heightInMeters)

# Display result
print("Weight In Kilograms is", format(weightInKilograms,
".3f"))
print("Height In Meters is", format(heightInMeters,
".3f"))
print("BMI is", format(bmi, ".2f"))
if bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal")
elif bmi < 30:
    print("Overweight")
else:
    print("Obese")
```

Homework

1. Write a program that prompts the user to enter three integers and displays them in increasing order.(without using any functions).

2. Write a program that asks the user to enter:

1. The student's grade (0–100)

2. Number of absence days

Then:

- If $\text{grade} < 50 \rightarrow \text{Fail}$
- If $\text{grade} \geq 50$ and $\text{absences} > 7 \rightarrow \text{Fail due to absence}$
- If $\text{grade} \geq 85$ and $\text{absences} \leq 3 \rightarrow \text{Excellent}$
- Otherwise $\rightarrow \text{Pass}$

3. Write a program that simulates a login system:

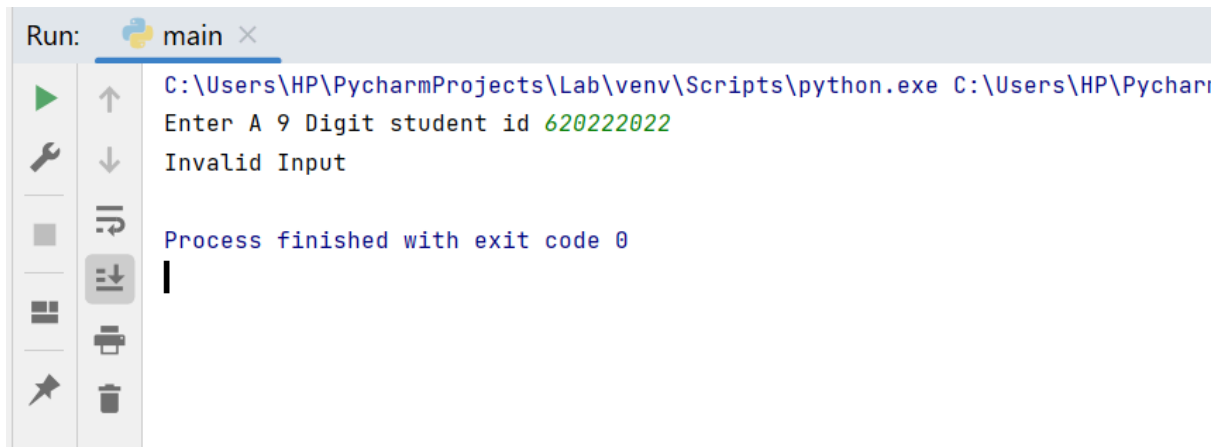
- Correct username: "admin"
- Correct password: "1234"

Conditions:

- Both correct $\rightarrow \text{Access Granted}$
- One incorrect $\rightarrow \text{Invalid Credentials}$
- Both empty $\rightarrow \text{No Input Provided}$

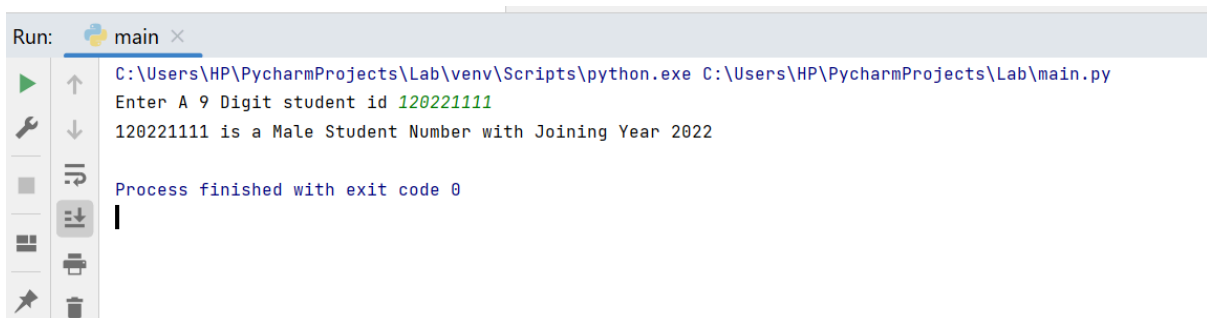
3. Write a program that allows the user to enter a nine-digit valid student ID and then find the following
 - Find if the ID for male or female (male if starting with 1, female if starting with 2)
 - Year of joining

Sample Output



```
Run: main x
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
Enter A 9 Digit student id 620222022
Invalid Input

Process finished with exit code 0
```



```
Run: main x
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
Enter A 9 Digit student id 120221111
120221111 is a Male Student Number with Joining Year 2022

Process finished with exit code 0
```