



Introducing Python

Lecture# 3

by



Umair bin Mansoor
Network Programming Planner

OBJECTIVES

After this session, students will be able to:

- **To program with Boolean expressions (AdditionQuiz)**
- **To implement and program selection control using one-way if statements**
- **To implement and program selection control using one-way if-else statements**
- **To implement selection control nested if and multi-way if-elif-else statements**
- **To use selection statements with combined conditions (LeapYear, Lottery)**
- **To write expressions that use the conditional expressions**
- **To understand the rules governing operator precedence and associativity**

BOOLEAN VARIABLE AND EXPRESSION

- Boolean variables can have two literals as its value: **True** or **False**
- Relational operators return Boolean values

<i>Python Operator</i>	<i>Mathematics Symbol</i>	<i>Name</i>	<i>Example (radius is 5)</i>	<i>Result</i>
<	<	less than	<code>radius < 0</code>	False
<=	≤	less than or equal to	<code>radius <= 0</code>	False
>	>	greater than	<code>radius > 0</code>	True
>=	≥	greater than or equal to	<code>radius >= 0</code>	True
==	=	equal to	<code>radius == 0</code>	False
!=	≠	not equal to	<code>radius != 0</code>	True

BOOLEAN **EXPRESSION**

[AdditionQuiz\(3_1\).py](#)

1. **import random**
2. *# Generate random numbers*
3. **number1 = random.randint(0, 9)** *#Generates random numbers from 0-9*
4. **number2 = random.randrange(0, 10)** *#Generates random numbers from 0-9*
5. *# Prompt the user to enter an answer*
6. **answer = eval(input("What is " + str(number1) + " + " + str(number2) + "? "))**
7. *# Display result*
8. **print(number1, "+", number2, "=", answer, "is", number1 + number2 == answer)**

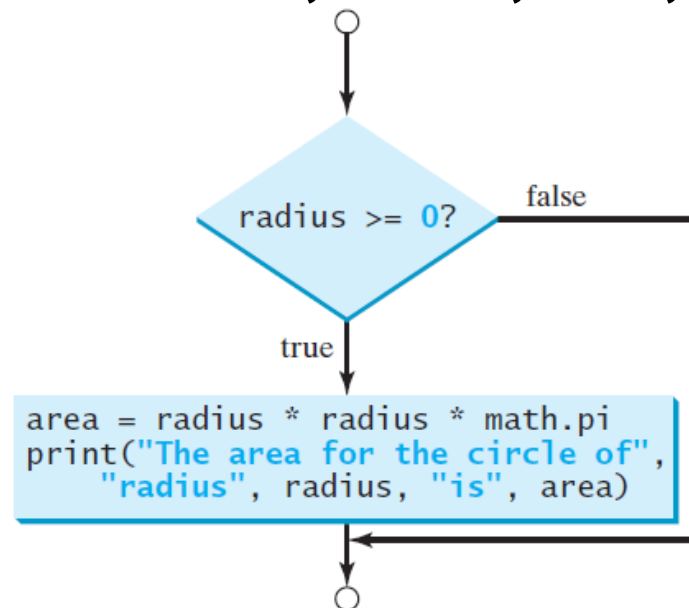
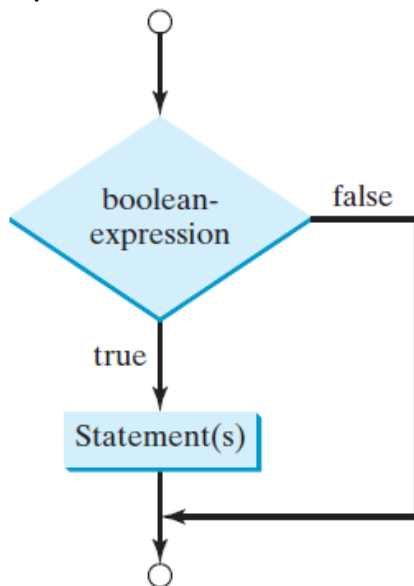
if STATEMENTS

- A one-way if statement executes the statements if the condition is true.

if radius >= 0:

→ **area = radius * radius * math.pi**

→ **print("The area for the circle of radius", radius, "is", area)**



SIMPLE if PROGRAM

[SimpleIfDemo\(3_2\).py](#)

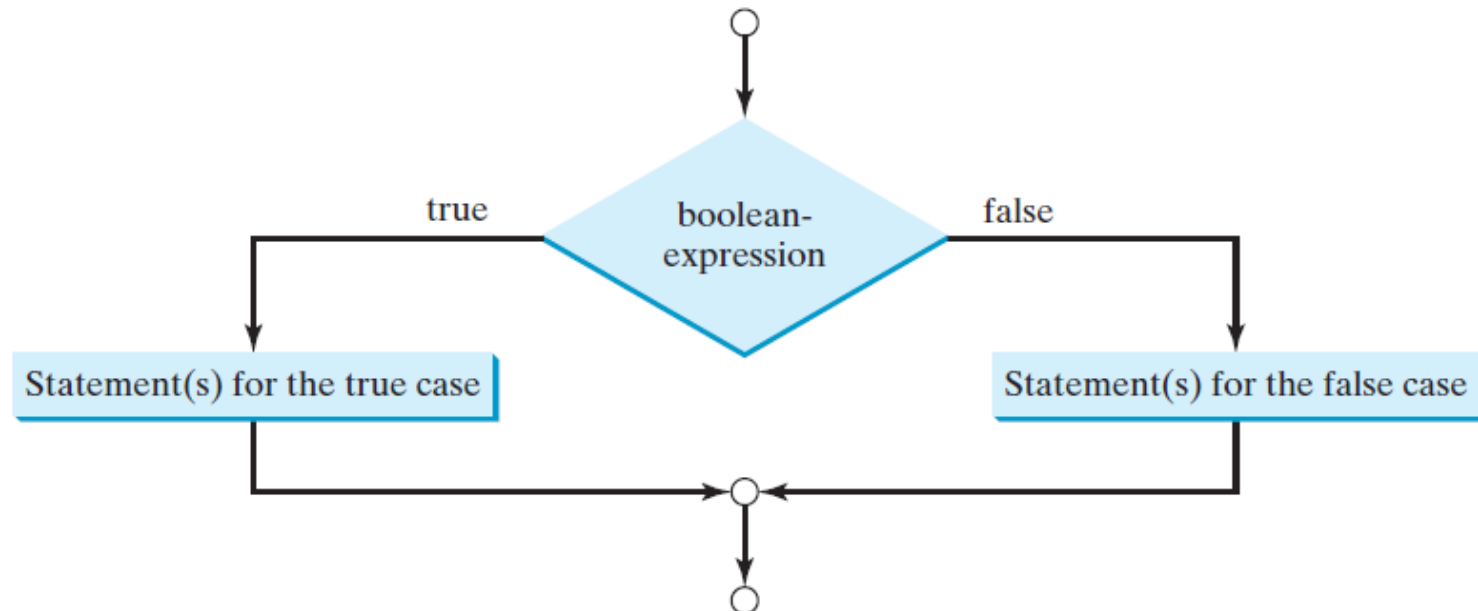
```
1.  # This program checks the number if it is multiple of 5 or 2
2.  number = eval(input("Enter an integer: "))
3.  if number % 5 == 0:
4.      print("HiFive")
5.  if number % 2 == 0:
6.      print("HiEven")
```

[SubtractionQuiz\(3_3\).py](#)

```
1.  import random # This program subtracts and then compares the answer of to numbers
2.  number1 = random.randint(0, 9)
3.  number2 = random.randint(0, 9)
4.  if number1 < number2:
5.      number1, number2 = number2, number1 # Simultaneous assignment
6.  answer = eval(input("What is "+ str(number1) + " - " + str(number2) + "? "))
7.  if number1 - number2 == answer:
8.      print("You are correct!")
9.  else:
10. print("Your answer is wrong.\n", number1, '-', number2, "is", number1 - number2, '.')
```

TWO-WAY if PROGRAM

- A two-way **if-else** statement decides which statements to execute based on whether the condition is **true** or **false**.



NESTED if AND MULTI-WAY if-elif-else STATEMENTS

- *One if statement can be placed inside another if statement to form a nested if statement.*

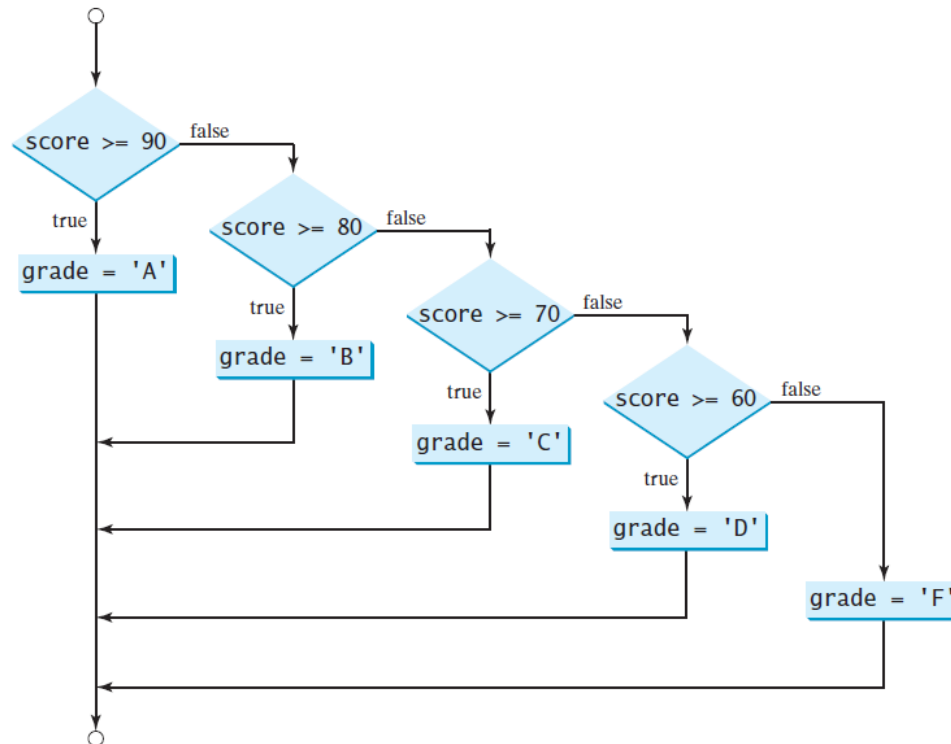
```
if score >= 90.0:  
    grade = 'A'  
else:  
    if score >= 80.0:  
        grade = 'B'  
    else:  
        if score >= 70.0:  
            grade = 'C'  
        else:  
            if score >= 60.0:  
                grade = 'D'  
            else:  
                grade = 'F'
```

Equivalent

This is better

```
if score >= 90.0:  
    grade = 'A'  
elif score >= 80.0:  
    grade = 'B'  
elif score >= 70.0:  
    grade = 'C'  
elif score >= 60.0:  
    grade = 'D'  
else:  
    grade = 'F'
```


MULTI-WAY if-elif-else BLOCK DIAGRAM



MULTI-WAY if-elif-else EXAMPLE

[ChineseZodiac\(3_4\).py](#)

```
1. year = eval(input("Enter a year: "))
2. zodiacYear = year % 12
3. if zodiacYear == 0:
4.     print("monkey")
5. elif zodiacYear == 1:
6.     print("rooster")
7. elif zodiacYear == 2:
8.     print("dog")
9. elif zodiacYear == 3:
10.    print("pig")
11. elif zodiacYear == 4:
12.    print("rat")
13. elif zodiacYear == 5:
14.    print("ox")
15. elif zodiacYear == 6:
16.    print("tiger")
17. elif zodiacYear == 7:
18.    print("rabbit")
19. elif zodiacYear == 8:
20.    print("dragon")
21. elif zodiacYear == 9:
22.    print("snake")
23. elif zodiacYear == 10:
24.    print("horse")
25. else:
26.    print("sheep")
```

LOGICAL OPERATORS

- The logical operators not, and, and or can be used to create a composite condition.
- Logical operators, also known as Boolean operators, operate on Boolean values to create a new Boolean value.
 - The not operator negates True to False and False to True.
 - The and of two Boolean operands is True if and only if both operands are True.
 - The or of two Boolean operands is True if at least one of the operands is True.

<i>Operator</i>	<i>Description</i>
not	logical negation
and	logical conjunction
or	logical disjunction

LOGICAL OPERATORS

EXAMPLE

[LeapYear\(3_5\).py](#)

1. `year = eval(input("Enter a year: "))`
2. `isLeapYear = (year % 4 == 0 and year % 100 != 0) or \`
`(year % 400 == 0) # Check if the year is a leap year`
3. `print(year, "is a leap year?", isLeapYear) # Display the result`

Enter a year: 2008
2008 is a leap year? True

Enter a year: 1900
1900 is a leap year? False

OPERATOR PRECEDENCE AND ASSOCIATIVITY

- Operator precedence and associativity determine the order in which operators are evaluated.
3 + 4 * 4 > 5 * (4 + 3) - 1
- What is its value? What is the execution order of the operators?
 - Arithmetically, the expression in the parentheses is evaluated first. (Parentheses can be nested, in which case the expression in the inner parentheses is executed first.)

<i>Precedence</i>	<i>Operator</i>
	+, - (Unary plus and minus)
	** (Exponentiation)
	not
	*, /, //, % (Multiplication, division, integer division, and remainder)
	+, - (Binary addition and subtraction)
	<, <=, >, >= (Comparison)
	==, != (Equality)
	and
	or
	=, +=, -=, *=, /=, //=, %= (Assignment operators)

OPERATOR PRECEDENCE AND ASSOCIATIVITY

- If operators with the same precedence are next to each other, their associativity determines the order of evaluation.
- All binary operators are left-associative. For example, since + and – are of the same precedence and are left-associative, the expression

$$a - b + c - d \quad \text{is equivalent to} \quad \underline{\underline{((a - b) + c) - d}}$$

EXERCISES

1. Assuming that x is 1, show the result of the following Boolean expressions.

- True and (3 > 4)
- not (x > 0) and (x > 0)
- (x > 0) or (x < 0)
- (x != 0) or (x == 0)
- (x >= 0) or (x < 0)
- (x != 1) == not (x == 1)

2. Rewrite the following if statements using a conditional expression:

```
if ages >= 16:  
    ticketPrice = 20  
else:  
    ticketPrice = 10
```

```
if count % 10 == 0:  
    print(count)  
else:  
    print(count, end = " ")
```

EXERCISES

3. Rewrite the following conditional expressions using `if/else` statements:

(a) `score = 3 * scale if x > 10 else 4 * scale`

(b) `tax = income * 0.2 if income > 10000 else income * 0.17 + 1000`

(c) `print(i if number % 3 == 0 else j)`

4. Evaluate the following expressions keeping operator precedence in mind:

`2 * 2 - 3 > 2 and 4 - 2 > 5`

`2 * 2 - 3 > 2 or 4 - 2 > 5`

Questions & Answers

