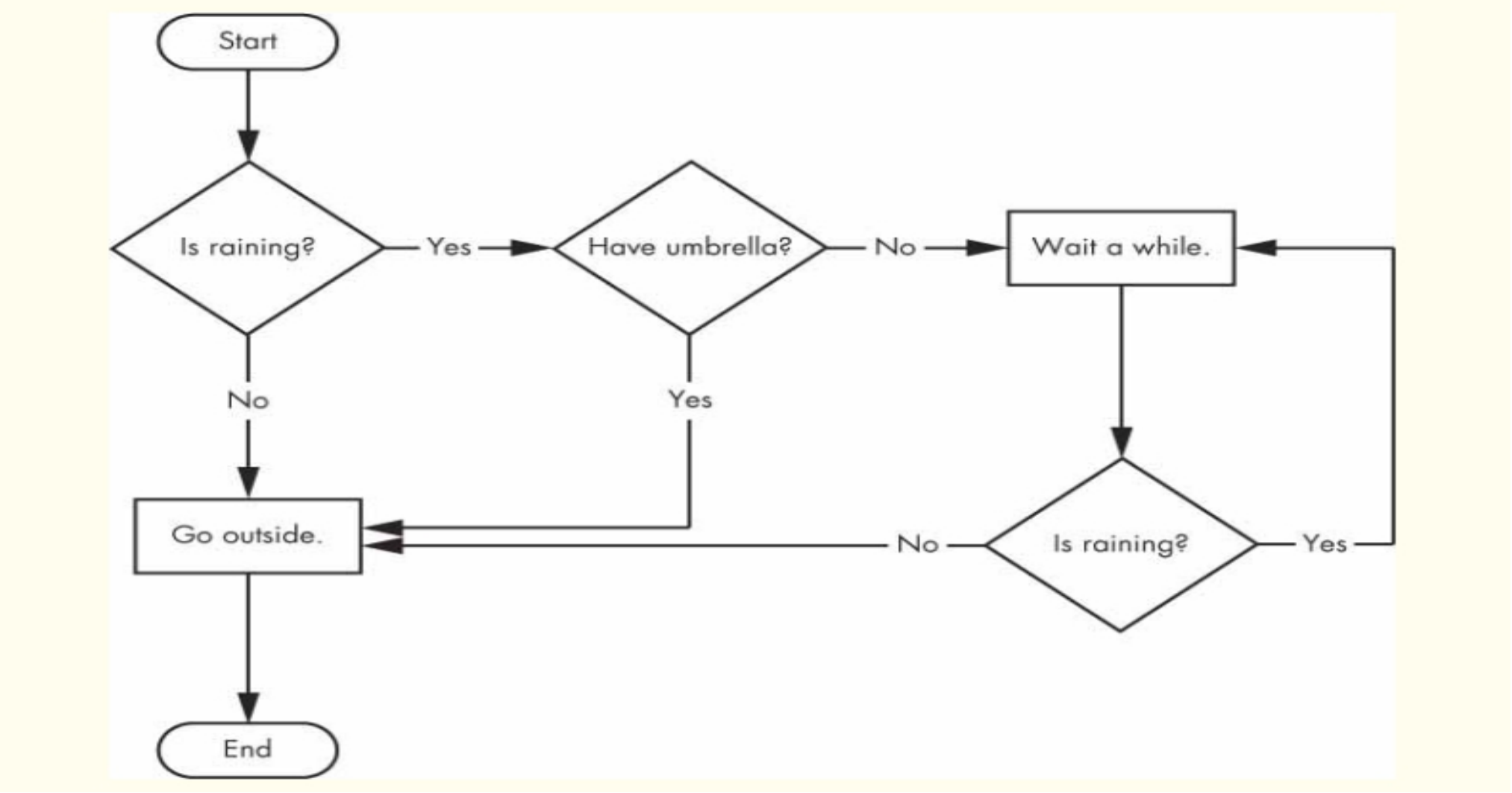
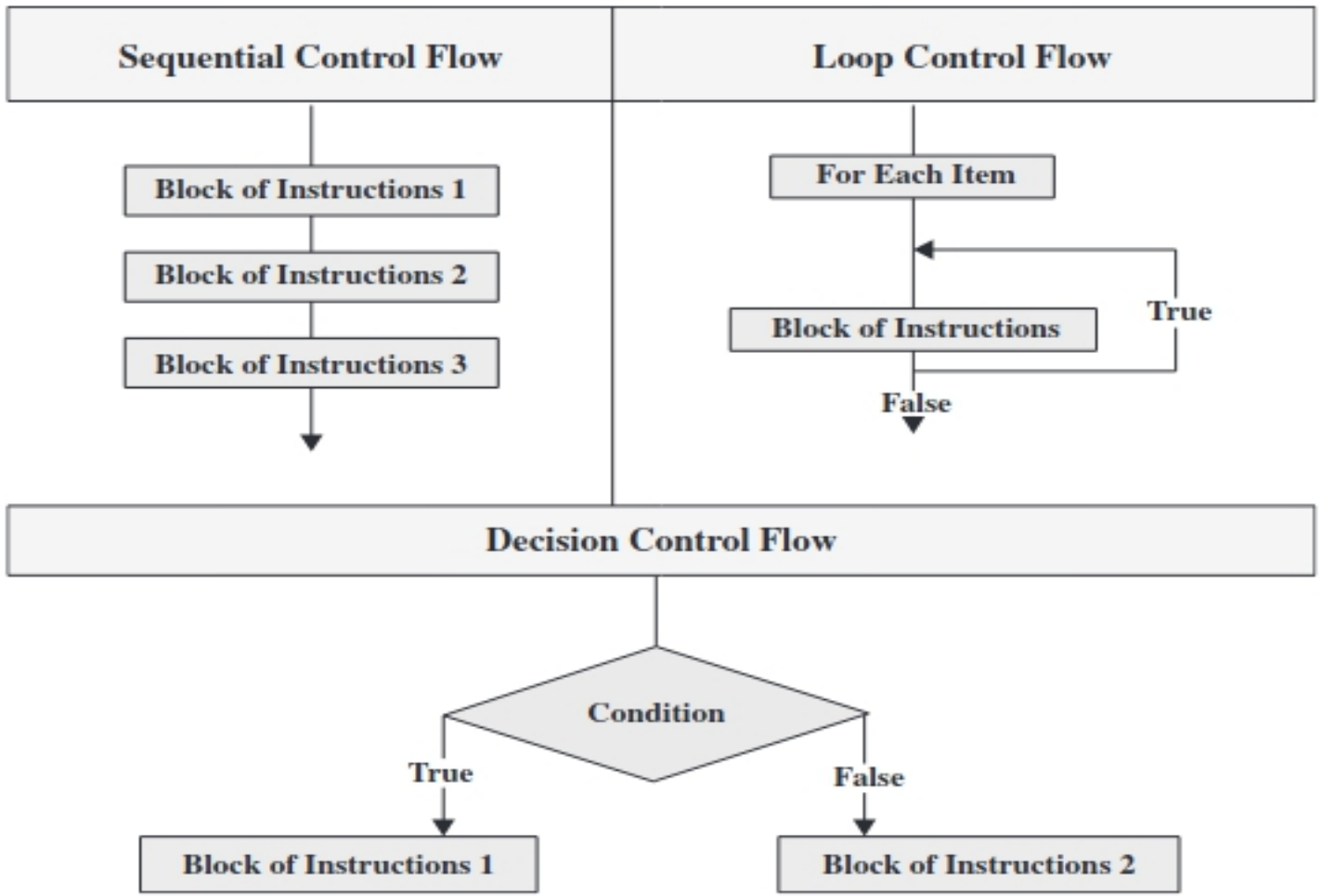
Lecture 2

Decision Flow Control

A flowchart to tell you what to do if it is raining



Forms of control flow statements



Condition Statement

- Condition Statements:

Statement that check the correctness of a certain condition.

- Condition may be: comparision , logical or compound condition

Comparision Condition

```
In [1]:
1 x=2
2 y=0
3 x>y
```

```
Out[1]: True
```

Logical Condition

```
In [5]:
1 x= False
2 y=True
3 x or y
```

```
Out[5]: True
```

Complex Condition

```
In [3]:
1 x=2
2 y=0
3 x>y or y==x
```

```
Out[3]: True
```

Comparison Operators

- Comparison operators, also called relational operators, compare two values and evaluate down to a single Boolean value.

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

Comparison Operators... Example

```
>>> 42 == 42
True
>>> 42 == 99
False
>>> 2 != 3
True
>>> 2 != 2
False
```

```
>>> 'hello' == 'hello'
True
>>> 'hello' == 'Hello'
False
>>> 'dog' != 'cat'
True
>>> True == True
True
>>> True != False
True
>>> 42 == 42.0
True
>>> 42 == '42'
False
```

```
>>> eggCount = 42
>>> eggCount <= 42
True
>>> myAge = 29
>>> myAge >= 10
True
```

THE DIFFERENCE BETWEEN THE == AND = OPERATORS

You might have noticed that the == operator (equal to) has two equal signs, while the = operator (assignment) has just one equal sign. It's easy to confuse these two operators with each other. Just remember these points:

- The == operator (equal to) asks whether two values are the same as each other.
- The = operator (assignment) puts the value on the right into the variable on the left.

To help remember which is which, notice that the == operator (equal to) consists of two characters, just like the != operator (not equal to) consists of two characters.

Boolean (Logical) Operators

List of Logical Operators

Operator	Operator Name	Description	Example
and	Logical AND	Performs AND operation and the result is True when both operands are True	p and q results in False
or	Logical OR	Performs OR operation and the result is True when any one of both operand is True	p or q results in True
not	Logical NOT	Reverses the operand state	not p results in False

Note: The Boolean value of p is True and q is False.

and operator truth table

Expression	Evaluates to . . .
`True and True`	`True`
`True and False`	`False`
`False and True`	`False`
`False and False`	`False`

```
>>> True and True
True
>>> True and False
False
```

or operator truth table

Expression	Evaluates to . . .
True or True	True
True or False	True
False or True	True
False or False	False

```
>>> False or True
True
>>> False or False
False
```

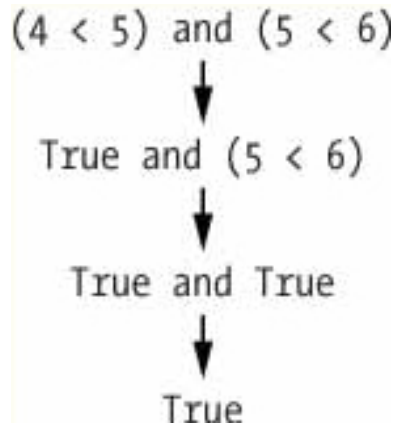
not operator truth table

Expression	Evaluates to . . .
`not True`	`False`
`not False`	`True`

```
>>> not True
False
>>> not not not not True
True
```

Mixing Boolean and Comparison Operators

```
>>> (4 < 5) and (5 < 6)
True
>>> (4 < 5) and (9 < 6)
False
>>> (1 == 2) or (2 == 2)
True
>>> 2 + 2 == 4 and not 2 + 2 == 5 and 2 * 2 == 2 + 2
True
```




```
>>> not(True) and False
False
>>> not(True and False)
True
>>> (10 < 0) and (10 > 2)
False
>>> (10 < 0) or (10 > 2)
True
>>> not(10 < 0) or (10 > 2)
True
>>> not(10 < 0 or 10 > 2)
False
```

Flow Control

- *Decision* statement

```
if condition:
    indented block of code
```

- Loops

```
while condition:
    indented block of code
```

- Blocks are Python statements.
- Blocks begin when the indentation increases.
- Blocks can contain other blocks.
- Blocks end when the indentation decreases to zero or to a containing block's indentation.

```
name = 'Mary'
password = 'swordfish'
if name == 'Mary':
    ❶ print('Hello, Mary')
    if password == 'swordfish':
        ❷ print('Access granted.')
    else:
        ❸ print('Wrong password.')
```

The Decision Control Flow Statement (*if*)

The Decision Control Flow Statement (*if*)

- if-statement

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

```
b is greater than a
```

- if-else statement

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

```
b is not greater than a
```

- if-elif statement

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

```
a and b are equal
```

- if-elif-else statement

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

```
a is greater than b
```

if Statement

- The syntax for if statement is,

```
if condition:
    indented block of code
```

```
1  x=10
2  if x>0:
3      print('positive number')
```

```
positive number
```

- The if decision control flow statement starts with **if** keyword and ends with a colon.
- The condition in an if statement should be a Boolean expression.
- The if statement decides whether to run some particular statement or not depending upon the value of the Boolean expression.
- If the Condition evaluates to True then statements in the indented block will be executed; otherwise the result is False then none of the statements are executed.
- In Python, the **if** block statements are determined through indentation and the first unindented statement marks the end.

if Statement

Syntax

if condition:
    body

if condition: body

If you have only one statement to execute, you can put it on the same line as the if statement.

Example

```
[160]: a = 33
       b = 200
       if b > a:
       print("b is greater than a") # you will get an error
```

```
  File "<ipython-input-160-4276c1871af7>", line 4
    print("b is greater than a") # you will get an error
        ^
IndentationError: expected an indented block
```

```
[161]: a = 33
       b = 200
       if b > a:
         print("b is greater than a")
```

```
b is greater than a
```

```
[156]: if a > b: print("a is greater than b")
```

Example 1

Program Reads a Number and Prints a Message If It Is Positive

```
1 number = int(input("Enter a number"))
2 if number >= 0:
3     print("The number entered by the user is a positive number")
4
```

```
Enter a number-10
```

```
Enter a number10
The number entered by the user is a positive number
```

Example 2

- Program to Read Luggage Weight and Charge the Tax Accordingly

```
In [9]:
1 weight = int(input("How many pounds does your suitcase weigh?"))
2 if weight > 50:
3     print(f"There is a $25 surcharge for heavy luggage")
4     print(f"Thank you")
```

```
How many pounds does your suitcase weigh?120
There is a $25 surcharge for heavy luggage
Thank you
```

```
How many pounds does your suitcase weigh?50
```

Example 3

- Write a python program that takes an integer number from the user. Then it prints "this number is greater than 10" if the user enters a number >10.

In [31]:

```
1 n=int(input("enter an integer number: "))
2 if n>10:
3     print("This number is:")
4     print("greater than 10")
5 print("Finish")
```

```
enter an integer number: 11
This number is:
greater than 10
Finish
```

Example 4

- What is the output of the following:

```
In [47]:
1 myInt=12
2 num1 = 12 if myInt==10 else 13
3 print(num1)

13
```

```
In [48]:
1 myInt=5
2 print ("This is task A" if myInt == 10 else 'This is task B')

This is task B
```

Example 5: Correct the following Code

Error

```
In [13]:
1 number = int(input("Enter a number"))
2 if number >= 0:
3 print("The number entered by the user is a positive number")
```

```
  File "<ipython-input-13-b71150c7508f>", line 3
    print("The number entered by the user is a positive number")
        ^
IndentationError: expected an indented block
```

Correction

```
In [7]:
1 number = int(input("Enter a number"))
2 if number >= 0:
3     print("The number entered by the user is a positive number")
4
```

```
Enter a number-10
```

Example 6 : Correct the folowing code

Error

In [12]:
```
1 weight = int(input("How many pounds does your suitcase weigh?"))
2 if weight > 50:
3       print(f"There is a $25 surcharge for heavy luggage")
4     print(f"Thank you")
```

```
  File "<tokenize>", line 4
    print(f"Thank you")
    ^
IndentationError: unindent does not match any outer indentation level
```

Correction

In [9]:
```
1 weight = int(input("How many pounds does your suitcase weigh?"))
2 if weight > 50:
3     print(f"There is a $25 surcharge for heavy luggage")
4     print(f"Thank you")
```

```
How many pounds does your suitcase weigh?120
There is a $25 surcharge for heavy luggage
Thank you
```

if-else statement

The *if...else* Statement

- An if statement can also be followed by an else statement which is optional.
- An else statement does not have any condition.
- indented block of code1 is executed if the Condition is True.

```
if condition:
    indented block of code1
else:
    indented block of code2
```

- Use the optional **else** block to execute indented block of code2 if the Condition is False.
- The ***if...else*** statement allows for a two-way decision.
- Note, if and else keywords should be aligned at the same column position.

if-else Statements...

Syntax

```
if condition:
    indented block of code1
else:
    indented block of code2
```

Example

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

```
b is not greater than a
```

If you have only one statement to execute, you can put it on the same line as the if/else statement.

```
if condition:    indented block of code1
else:    indented block of code2
```

```
a = 2
b = 330
print("A") if a > b else print("B")
```

```
B
```

Example 1

- Program to Find If a Given Number Is Odd or Even

```
In [14]:
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4 else:
5     print(f"{number} is Odd number")
```

```
Enter a number2
2 is Even number
```

```
In [14]:
1 number = int(input("Enter a number"))
2 if number % 2 == 0: print(f"{number} is Even number")
3 else: print(f"{number} is Odd number")
```

```
Enter a number2
2 is Even number
```

```
Enter a number5
5 is Odd number
```

```
Enter a number5
5 is Odd number
```

Eample 2

- Program to Find the Greater of Two Numbers

In [24]:
```
number_1 = int(input("Enter the first number"))
number_2 = int(input("Enter the second number"))
if number_1 > number_2:
    print(f"{number_1} is greater than {number_2}")
else:
    print(f"{number_2} is greater than {number_1}")
```

```
Enter the first number2
Enter the second number10
10 is greater than 2
```

In [24]:
```
number_1 = int(input("Enter the first number"))
number_2 = int(input("Enter the second number"))
if number_1 > number_2: print(f"{number_1} is greater than {number_2}")
else: print(f"{number_2} is greater than {number_1}")
```

```
Enter the first number2
Enter the second number10
10 is greater than 2
```

Example 3

- Write a python program that takes two interger numbers from the user.
- Then it divides the first number by the second number if the second number is not equals to zero, otherwise, it prints "cannot divide by zero".

In [32]:

```
num1=int(input("enter the first number: "))
num2 = int(input("enter the second number: "))
if num2==0:
    print("cannot divide by zero")
else:
    print(num1/num2)
print('finish')
```

```
enter the first number: 2
enter the second number: 0
cannot divide by zero
finish
```

```
enter the first number: 4
enter the second number: 8
0.5
finish
```

Example 4: Correct the following code

Error

In [16]:

```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4 print("the condition is true")
5 else:
6     print(f"{number} is Odd number")
```

```
  File "<ipython-input-16-fb762e1be263>", line 5
    else:
       ^
SyntaxError: invalid syntax
```

Correction

In [17]:

```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4     print("the condition is true")
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number10
10 is Even number
the condition is true
```

Error

```
In [18]:
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4         print("the condition is true")
5 else:
6     print(f"{number} is Odd number")
```

```
  File "<ipython-input-18-372f49645137>", line 4
    print("the condition is true")
    ^
IndentationError: unexpected indent
```

Correction

```
In [17]:
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4     print("the condition is true")
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number10
10 is Even number
the condition is true
```

Error

In [19]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4     print("the condition is true")
5 else:
6 print(f"{number} is Odd number")
```

```
  File "<ipython-input-19-b6b20e550a77>", line 6
    print(f"{number} is Odd number")
        ^
IndentationError: expected an indented block
```

Correction

In [17]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4     print("the condition is true")
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number10
10 is Even number
the condition is true
```

Error

In [21]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3 print(f"{number} is Even number")
4
5 else:
6       print(f"{number} is Odd number")
```

```
  File "<ipython-input-21-b18a241b25fa>", line 3
    print(f"{number} is Even number")
        ^
IndentationError: expected an indented block
```

Correction

In [22]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

In [23]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4
5 else:
6       print(f"{number} is Odd number")
```

```
Enter a number13
13 is Odd number
```

Error

In [29]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4
5 else number %2 !=0:
6     print(f"{number} is Odd number")
```

```
  File "<ipython-input-29-13cec48b0fa5>", line 5
    else number %2 !=0:
               ^
SyntaxError: invalid syntax
```

Correction

In [22]:
```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

```
1 number = int(input("Enter a number"))
2 if number % 2 == 0: print(f"{number} is Even number")
3 else: print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

Error

```
In [25]:
1 number = int(input("Enter a number"))
2
3 else:
4       print(f"{number} is Odd number")
```

```
  File "<ipython-input-25-b10bf9f5c29c>", line 3
    else:
       ^
SyntaxError: invalid syntax
```

Correction

```
In [22]:
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

Error

In [27]:
```
number = int(input("Enter a number"))
else:
    print(f"{number} is Odd number")
if number % 2 == 0:
    print(f"{number} is Even number")
```

```
  File "<ipython-input-27-56064c5a3028>", line 2
    else:
       ^
SyntaxError: invalid syntax
```

Correction

In [22]:
```
number = int(input("Enter a number"))
if number % 2 == 0:
    print(f"{number} is Even number")

else:
    print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

Error

In [28]:

```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4 
5 else:
6     print(f"{number} is Odd number")
7 else:
8     print("the second else!!")
```

```
  File "<ipython-input-28-0049872b0d86>", line 7
    else:
    ^
SyntaxError: invalid syntax
```

Correction

In [22]:

```
1 number = int(input("Enter a number"))
2 if number % 2 == 0:
3     print(f"{number} is Even number")
4 
5 else:
6     print(f"{number} is Odd number")
```

```
Enter a number16
16 is Even number
```

if-elif statement

```
if condition1:
    indented block of code1
elif condition2:
    indented block of code2
elif condition3:
    indented block of code3
                .
                .
```

```
if condition1:
    indented block of code1
else:
  if condition2:
      indented block of code2
  else:
  if condition3:
      indented block of code3
```

- The keyword '**elif**' is short for 'else if' and is useful to avoid excessive indentation.
- When you need to choose from several possible alternatives, then an elif statement is used along with an if statement.
- indented block of code1 is executed if the Condition1 is True
- indented block of code2 is executed if Condition1 is False and Condition2 is True
- indented block of code3 is executed if Condition1 and Condition2 are False while Condition3 is True

if-elif statement Example

```
name = 'Carol'
age = 3000
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
elif age > 2000:
    print('Unlike you, Alice is not an undead, immortal vampire.')
elif age > 100:
    print('You are not Alice, grannie.')
```

if...elif...else Statement

- The ***if...elif...else*** is also called as **multi-way decision control** statement.
- The else statement must always come last, and will again act as the default action.
- indented block of code1 is executed if the Condition1 is True
- indented block of code2 is executed if Condition1 is False and Condition2 is True
- indented block of code3 is executed if Condition1 and Condition2 are False while Condition3 is True
- indented block of codeN is executed if Condition1, Condition2, and Condition3 are False
- Note: There can be only one **else** block.

```
if condition1:
    indented block of code1
elif condition2:
    indented block of code2
    .
elif condition3:
    indented block of code3
    .
    .
else :
    indented block of codeN
```

if-elif-else Statement ...

Syntax

```
if condition1:
    indented block of code1
elif condition2:
    indented block of code2
                .
elif condition3:
    indented block of code3
                .
                .
else :
    indented block of codeN
```

```
if condition1:      indented block of code1
elif condition2:    indented block of code2
                                .
elif condition3:    indented block of code3
                                .
                                .
else :      indented block of codeN
```

Example

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

```
a is greater than b
```

Example 1

Program 3.5: Write a Program to Prompt for a Score between 0.0 and 1.0. If the Score Is Out of Range, Print an Error. If the Score Is between 0.0 and 1.0, Print a Grade Using the Following Table

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

Example solution

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

In [51]:

```
1  score = float(input("Enter your score"))
2  if score < 0 or score > 1:
3      print('Wrong Input')
4  elif score >= 0.9:
5      print('Your Grade is "A" ')
6  elif score >= 0.8:
7      print('Your Grade is "B" ')
8  elif score >= 0.7:
9      print('Your Grade is "C" ')
10 elif score >= 0.6:
11     print('Your Grade is "D" ')
12 else:
13      print('Your Grade is "F" ')
```

```
Enter your score0.3
Your Grade is "F"
```

Example 2

Program 3.6: Program to Display the Cost of Each Type of Fruit

```
1. fruit_type = input("Enter the Fruit Type:")
2. if fruit_type == "Oranges":
3.     print('Oranges are $0.59 a pound')
4. elif fruit_type == "Apples":
5.     print('Apples are $0.32 a pound')
6. elif fruit_type == "Bananas":
7.     print('Bananas are $0.48 a pound')
8. elif fruit_type == "Cherries":
9.     print('Cherries are $3.00 a pound')
10. else:
11.     print(f'Sorry, we are out of {fruit_type}')
```

OUTPUT

```
Enter the Fruit Type: Cherries
Cherries are $3.00 a pound
```

Nested *if* Statement

- An if statement that contains another if statement either in its if block or else block is called a **Nested if statement**.

- If the condition1 is evaluated to True, then the control shifts to condition2:
  - if condition2 is evaluated to True, then ident**e**d_block1 is executed,
  - if condition2 is evaluated to False then the ident**e**d_block2 is executed.
- If condition1 is evaluated to False, then ident**e**d_block3 is executed.

```
if condition1:
   if condition2:
        indented_block1
   else:
        indented_block2
else:
    idented_block3
```

- If the condition1 is evaluated to True, then ident**e**d_block1 is executed.

- If condition1 is evaluated to False, then the control shifts to condition2:
  - if condition2 is evaluated to True, then ident**e**d_block2 is executed,
  - if condition2 is evaluated to False then the ident**e**d_block3 is executed.

```
if condition1:
    indented_block1
else:
   if condition2:
        indented_block2
   else:
        indented_block3
```

Nested *if* Statements...

General Syntax

```
if condition1:
    body1 {if or if-else or if-elif-end statment}
elif condition2:
    body2 {if or if-else or if-elif-end statment}
else:
    body3 {if or if-else or if-elif-end statment}
```

Example

In [58]:

```
num=int(input("enter a number: "))
if num%2==0:
    if num >0:
        print("{:d} is positive even number".format(num))
else:
    print("odd")
print('finish')
```

```
enter a number: 32
32 is positive even number
finish
```

In [60]:

```
num=int(input("enter a number: "))
if num%2==0:
    print("Even")
else:
    if num >0:
        print("{:d} is positive odd number".format(num))
print('finish')
```

```
enter a number: 33
33 is positive odd number
finish
```

Example1: Program to Check If a Given Year Is a Leap Year

All years which are perfectly divisible by 4 are leap years except for century years (years ending with 00) which is a leap year only it is perfectly divisible by 400. For example, years like 2012, 2004, 1968 are leap years but 1971, 2006 are not leap years. Similarly, 1200, 1600, 2000, 2400 are leap years but 1700, 1800, 1900 are not.

```
1  year = int(input('Enter a year'))
2  if year % 4 == 0:
3      if year % 100 == 0:
4          if year % 400 == 0:
5              print(f'{year} is a Leap Year')
6          else:
7              print(f'{year} is not a Leap Year')
8      else:
9          print(f'{year} is a Leap Year')
10 else:
11     print(f'{year} is not a Leap Year')
```

```
Enter a year2022
2022 is not a Leap Year
```

inline *if*

If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

body if condition else body

```
a = 2
b = 330
print("A") if a > b else print("B")
```

```
B
```

One line if else statement, with 3 conditions:

body if condition else body if condition else body

```
a = 330
b = 330
print("A") if a > b else print("=") if a == b else print("B")
```

```
=
```

The pass Statement

- The **pass** statement does nothing.
- It can be used when a statement is required syntactically but the program requires no action.
- if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error.
- Example

[164]:
```
a = 33
b = 200

if b > a:
  pass
```