Conditional statements

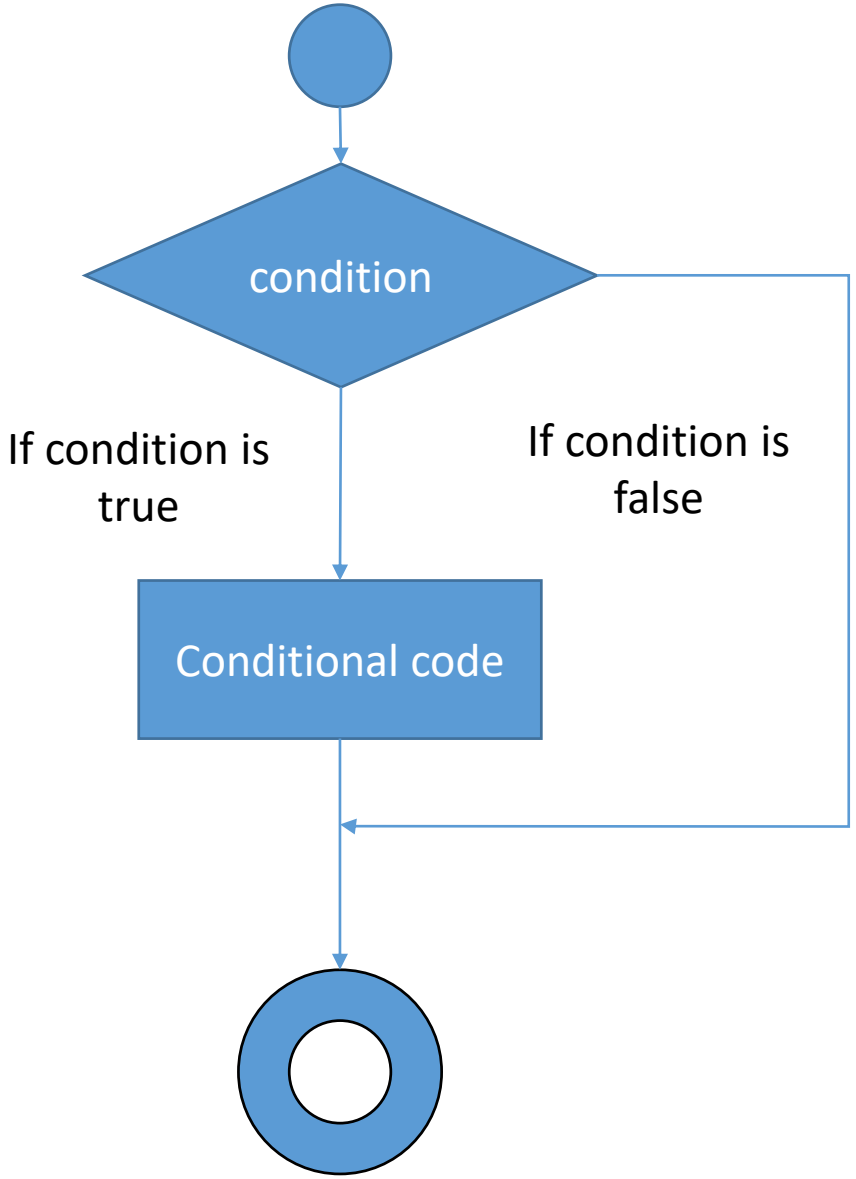
Lecture three

practical

Conditional statements

- Decision making is anticipation of conditions occurring while execution of the program specifying actions taken according to the conditions.
- Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome TRUE or FALSE otherwise.

General form of a typical decision making structure



Conditional statements

- Python programming language assumes any non-zero and non-null values as TRUE, and if its either zero Or null, then its assumed as FALSE value.
- Python programming language provides following types of decision making statements.

1. if statements
2. if … else statements
3. Nested if statements

if Statement

- It is similar to that of other languages. The if statement contains a logical expression using which data is compared and a decision is made based on the result of the comparison.
- Syntax

```
if expression:
    statement(s)
```

if Statement- example

- Example:

```
var1 = 100
if var1:
   print("1 - Got a true expression value”)
   print(var1)
var2 = 0
if var2:
   print("2 - Got a true expression value”)
   print(var2)
print("Good bye!”)
```

- When the above code is executed, it produces the following result:

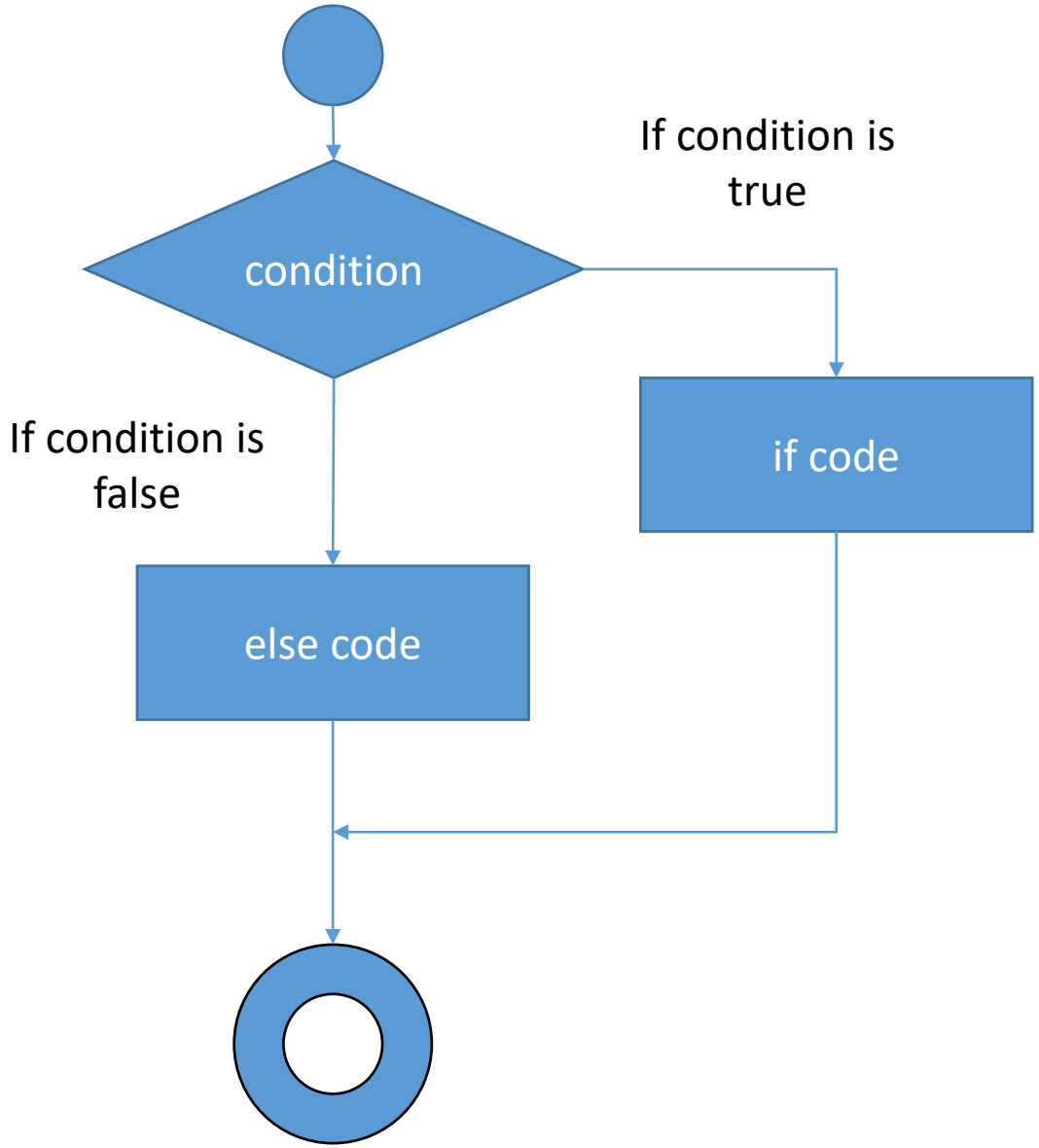
```
1 - Got a true expression value
100
Good bye!
```

if...else Statements

- An else statement can be combined with an if statement.
- An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.
- The else statement is an optional statement and there could be at most only one else statement following if.
- The syntax of the if...else statement is:

```
if expression:
        statement(s)
else:
        statement(s)
```

Flow diagram of if… else statement



if...else Statements- example

```
var1 = 100
if var1:
   print("1 - Got a true expression value“)
   print(var1)
else:
   print("1 - Got a false expression value“)
   print(var1)
var2 = 0
if var2:
   print("2 - Got a true expression value“)
   print(var2)
else:
   print("2 - Got a false expression value“)
   print(var2)
print("Good bye!“)
```

- When the above code is executed, it produces the following result

```
1 - Got a true expression value
 100
2 - Got a false expression value
00
Good bye!
```

elif Statement

- The elif code 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.
- Similar to the else, the elif statement is optional. However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following an if.
- syntax

```
if expression1:
   statement(s)
elif expression2:
  statement(s)
elif expression3:
  statement(s)
else:
  statement(s)
```

- Core Python does not provide switch or case statements as in other languages, but we can if..elif statements to simulate switch case.

elif Statement- example

```
var=100
if var==200:
   print("1 - Got a true expression value")
   print(var)
elif var==150:
   print("2 - Got a true expression value")
   print(var)
elif var==100:
   print("3 - Got a true expression value")
   print(var)
else:
   print("4 - Got a false expression value")
   print(var)
print("Good bye!")
```

- When the above code is executed, it produces the following result:

```
3 - Got a true expression value
100
Good bye!
```

nested IF statements

- There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested **if** construct.
- In a nested if construct, you can have an if...elif...else construct inside another if...elif...else construct.
- The syntax of the nested if...elif...else construct is:

```
if expression1:
   statement(s)
   if expression2:
      statement(s)
   elif expression3:
      statement(s)
   elif expression4:
      statement(s)
else:
   statement(s)
```

nested IF statements- example

```
var=100
if var<200:
   print("Expression value is less than 200")
   if var==150:
      print("Which is 150")
   elif var==100:
      print("Which is 100")
   elif var==50:
      print("Which is 50")
   elif var<50:
      print("Expression value is less than 50")
else:
   print("Could not find true expression")
print("Good bye!")
```

- When the above code is executed, it produces following:

```
Expression value is less than 200
Which is 100
Good bye!
```