Department of Social Statistics

Faculty of Humanities and Social Sciences

University of Sri Jayewardenepura

MIT/ITE 1213 – Fundamentals of Programming

Tutorial – 03

Python Conditional Statements

There are main types of python conditional statements.

1) If
2) Elif
3) Match

Logical conditions mainly need for this,

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

1) If

Try out the following examples using Google Colabs.

An "if statement" is written by using the if keyword.

```
a = 33

b = 200

if b > a:

        print("b is greater than a")
```

If statement, without indentation (will raise an error):

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

2) Elif

The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition".

See the following examples and identify their use in each case.

(a)

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

(b)

```
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")
```

(c)

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

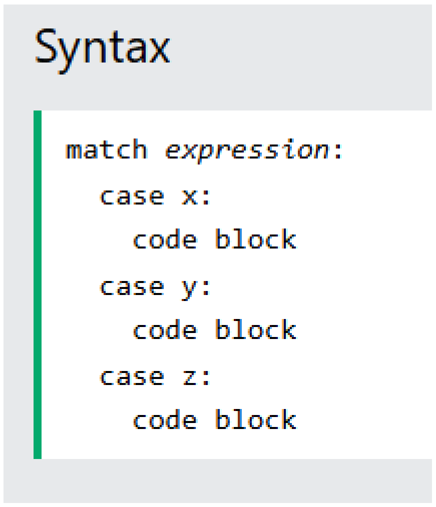
(d) if a > b: print("a is greater than b")

(e)

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

3) Match

This statement is used to perform different actions based on different conditions.



Syntax

```
match expression:
  case x:
    code block
  case y:
    code block
  case z:
    code block
```

This is how it works:

- The match expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

Try out the following examples.

(a)

```
day = 4
match day:
case 1:
print("Monday")
case 2:
print("Tuesday")
case 3:
print("Wednesday")
case 4:
print("Thursday")
case 5:
print("Friday")
case 6:
print("Saturday")
case 7:
print("Sunday")
```

(b)

Use the underscore character _ as the last case value if you want a code block to execute when there are not other matches:

```
day = 4
match day:
    case 6:
        print("Today is Saturday")
    case 7:
        print("Today is Sunday")
    case _:
         print("Looking forward to the Weekend")
```

(c)

Use the pipe character | as an or operator in the case evaluation to check for more than one value match in one case:

```
day = 4
match day:
        case 1 | 2 | 3 | 4 | 5:
                print("Today is a weekday")
        case 6 | 7:
                print("I love weekends!")
```

EXERCISE

According to the above practiced examples write the short programs using python for the given scenarios.

(1) Write a program that asks the user for their age. If the age is less than 13, print "Child". If it is between 13 and 19, print "Teenager". Otherwise, print "Adult".

(2) Ask the user to enter the temperature in Celsius. If it is less than 15, print "Cold". If it is between 15 and 25, print "Warm". Otherwise, print "Hot".

(3) A college uses letter grades for student performance:

Write a program that asks the user for their numerical score (0–100) and then uses Python's match statement to determine and print the letter grade:

- 90–100 ⟶ "A"
- 80–89 ⟶ "B"
- 70–79 ⟶ "C"
- 60–69 ⟶ "D"
- Below 60 ⟶ "F"