Loops(while) and Conditionals

By : Ramkrushna Pradhan

Conditional Statements

- General Format:

```
if (conditions to be met):
tab {do this}
    {and this}
    …
else:
    {do this}
    ...
```

if-else block

Conditional Statements

- Example: check if a number is even or not,

```
num = 53
if (num %2 == 0):
    print(“Number is even.”)
else :
    print(“Number is odd.”)
```

Conditional Statements

- Else if in python: written as ***elif***

```
if (condition_1):
    {do this}
elif (condition_2):
    {do this}
elif ....
else:
    {do this}
```

Conditional Statements

- Example: check if a number is positive, negative or zero

```
num = 62
if (num<0):
  print(“Number is negative”)
elif (num==0):
  print(“Number is zero”)
else:
  print(“Number is positive”)
```

Complex Conditional Statements

- **Nesting** if-else : if block inside another if block

```
if (condition_1):
        if (condition_2):
tab  tab  {do this}
        else:
                {do this instead}
else:
        {do this}
```

nested
if block

if block

Complex Conditional Statements

- **Compound** if statements : **and**

```
if (condition_1 and condition_2):
    {do this}
else:
    {do this instead}
```

Complex Conditional Statements

- **Compound** if statements : **or**

```
if (condition_1 or condition_2):
    {do this}
else:
    {do this instead}
```

While Loop

- General format

```
while (condition):
{ tab }{do this}
       {and this}
       {...}
```

While Loop

- Example : sum of first 10 numbers

```
num=1
sum=0
while (num <= 10):
    sum = sum + num
    num = num + 1
print(sum)
```

While Loop

- while + else ???
- else will execute when condition becomes false

```
while (condition):
    {continue executing this}
else :
    {execute this}
```

While Loop

- Example : sum of first 10 numbers

```
num=1
sum=0
while (num <= 10):
    sum = sum + num
    num = num + 1
else :
   print(sum)
```

Complex Loops

- **Nesting** while : while loop inside another while loop

```
while (condition_1):
        while (condition_2):
 tab   tab  {do this}
        else:
              {do this instead}
else:
        {do this}
```

nested while loop

while loop

Control Statements

- **break** : halt execution of a loop abruptly

```
while (condition_1):
        {keep doing this}
        if (condition_2 is met):
                {do this}
                break
```

Control Statements

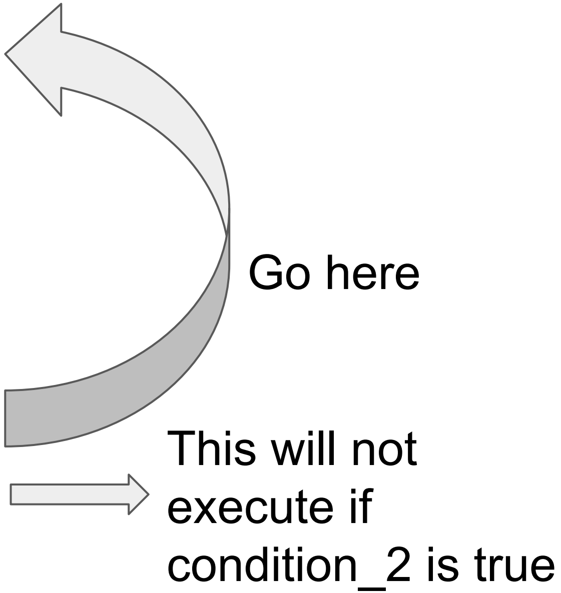
- Example : Read input until user enters a negative integer

```
num = int(raw_input(“Enter a number: ”))
while (num>0):
    print (“You entered ” +str(num))
    num = int(raw_input(“Enter a number: ”))
    if (num<0):
        print (“You entered negative number”)
        break
```

Control Statements

- **continue** : halt execution of the code after continue and repeat the loop

```
while (condition_1):
        {keep doing this}
        if (condition_2 is met):
                {do this}
                continue
        {some code over here}
```



Go here

This will not execute if condition_2 is true

Control Statements

- Example : Read only positive values, discard negative values

```
num = int(raw_input(“Enter a number : ”))
while (num>0):
    print (“You entered ” +str(num))
    num = int(raw_input(“Enter a number: ”))
    if (num<0):
        print (“You entered negative number.”)
        continue
```

Assignments - page 1 of 2

- Input an integer(<1000), and print the number of digits it has.
  (Hint : use if-elif-else, double digit means <100 etc..**)**
- Input an integer, and print the multiplication table for the number.
- Input 10 characters using while loop and count how many vowels are there.

Assignments - page 2 of 2

- Find sum of numbers from 1 to 100 which are divisible by **either 2 or 3.**
  (Hint : use while loop + compound if statement - “or”**)**

- Input a float number which will be the side of a square, prompt user to enter a positive value if user inputs a negative value. Finally calculate the area of the square.
  (Hint : use “continue” **)**