Statements and IO

Python Open Lab

Special Thanks: Sriharsha Devulapalli & Kunal Baweja

IO

- We have a system, which records working hours of each worker in this week and their hourly wage, and compute their total salary
- Workers input their working hours and hourly rate
- How to implement that?

IO

- Read from console

```
input = raw_input(‘Hours: ’)
x = int(input)
input = raw_input(‘Rate: ’)
y = float (input)
print x*y
```

IO

raw_input: read input from console

The program will pause for the user to input something

The input will be get by the type of string

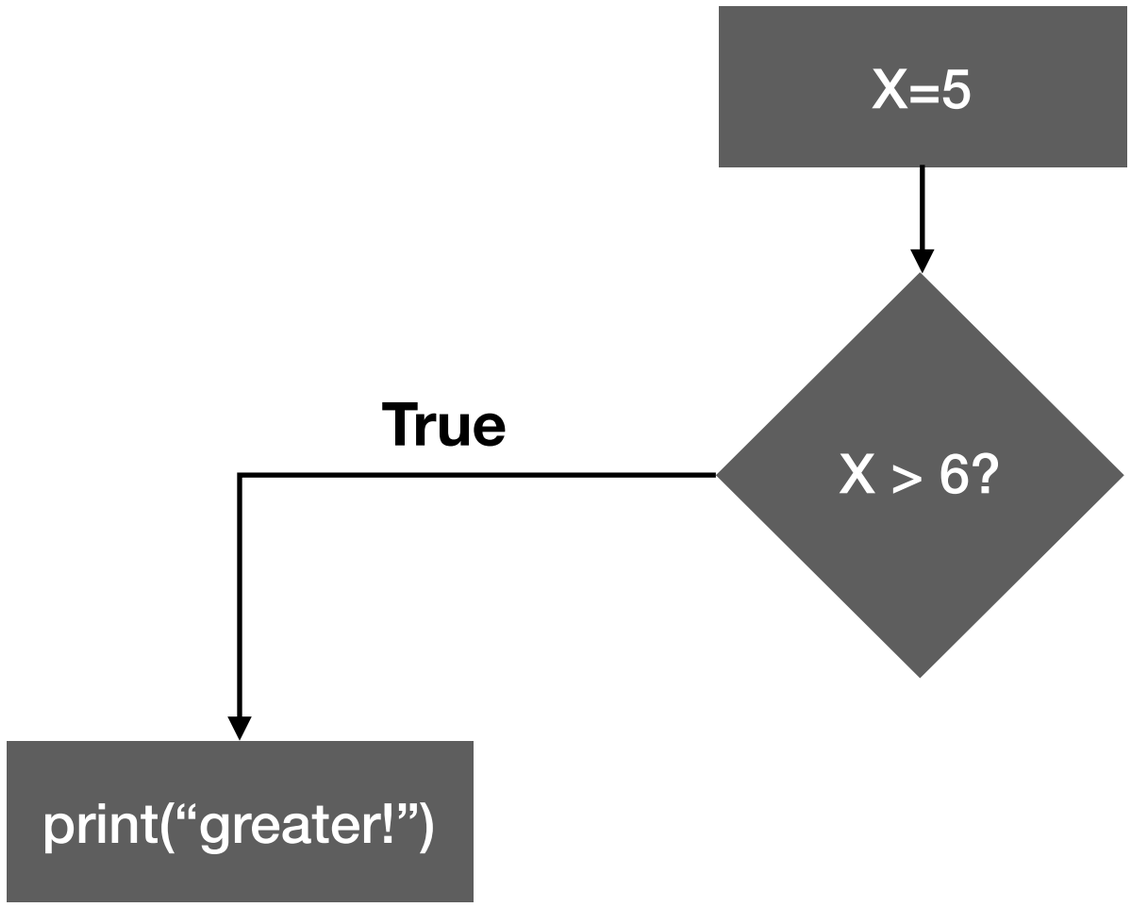
Conditional Statements

- If condition is met, execute following lines within indentation

- If condition is not met, just execute the following statements

Conditional Statements

- Execute a piece of code only if a set of one or more conditions is satisfied



Conditional Statements

- code

```
X=5

if X >6 :
    print(“great”)

print(“done!”)
```

Comparison Operators

- Compare left side with right
- Produce True/False output

Python	English Interpretation
<	Strictly Less than
<=	Less than or equals
==	Equals
>=	Greater than or equals
>	Strictly Greater than
!=	Not equals

if-elif-else

- if statement
- The structure of if statement is

    if <some condition>:

        do something

if-elif-else

- else statement
- else statement only appears when there is a if-statement

```
if <some condition>:
    do A
else:
    do B
```

exercise

- get a number of donation from user input
- if the input is bigger than 100, say “you are generous”
- if the input is smaller than 100 or equal to 100, say “thank you!”

if-elif-else

- elif statement
- elif only appears when there is a if-statement
- We can not use else statement to hold condition, but we can use elif to do this
- we can write multiple elif-statement following if-statement

if-elif-else

- elif statement

```
if <some condition>:
    do A
elif <some condition>:
    do B
elif <some condition>:
    do C
elif <some condition>:
    do D
```

if-elif-else

- elif statement and else statement can be used at the same time

```
if  X > 5:
    do A
elif X < 0:
    do B
elif X < 2:
    do C
else:
    do D
```

if-elif-else

- and & or

- I get a number, if it is larger than 15 or smaller than 0, I print “find it”.

```
if  X > 15:
    print(“find it”)
if X < 0:
    print(“find it”)
```

```
if  X > 15 or X < 0:
    print(“find it”)
```

if-elif-else

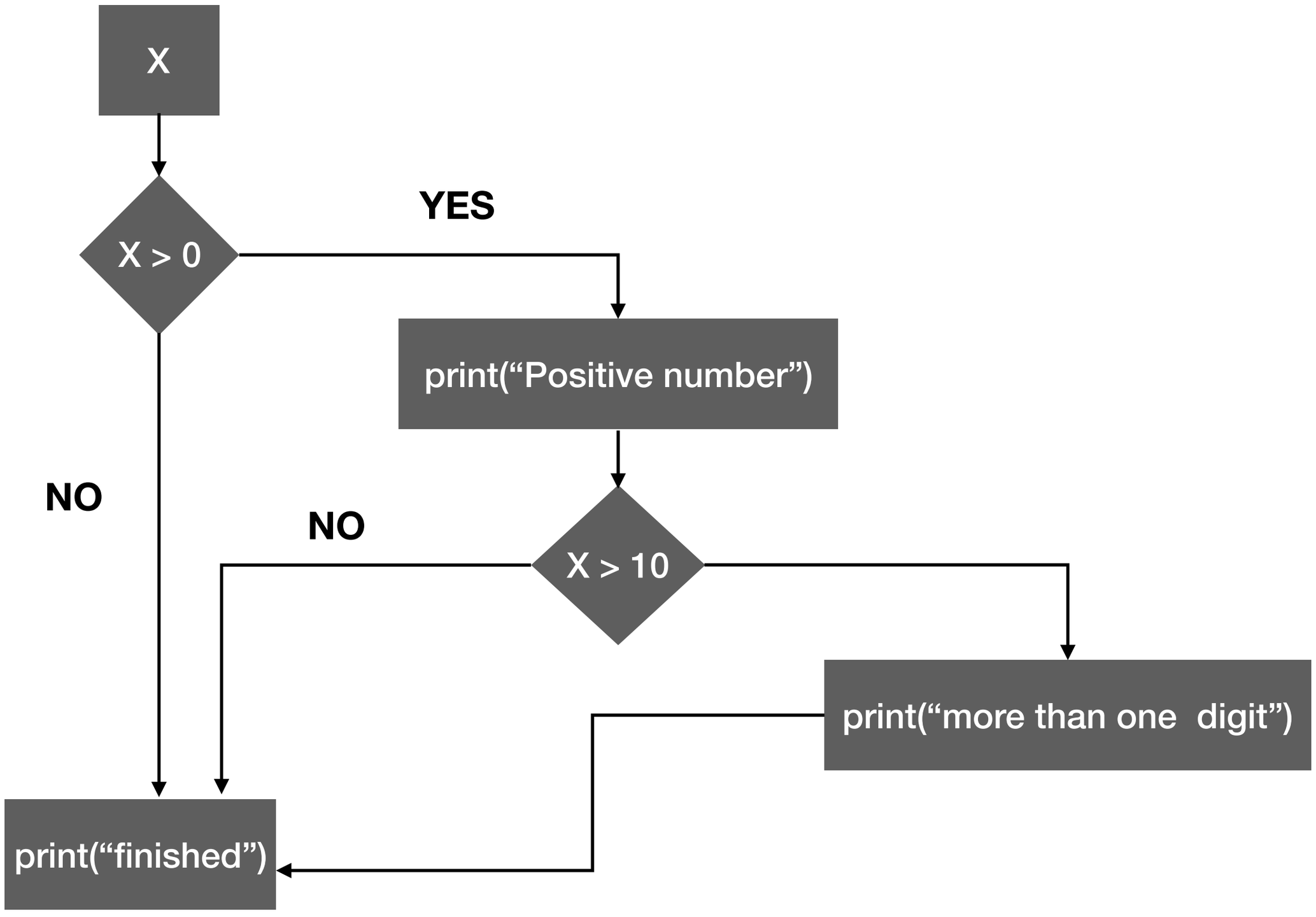
- and & or

- decide whether a number is smaller than 100 and bigger than 0, and it is an odd number

```
input = raw_input(“input an number:”)
X = int(input)

if  X > 0 and X < 100 and X%2 == 1:
    print(“find it”)
```

Nested decision



Nested decision

Which statements will print ?

```
x = 5

if x>0:
    print “Positive number”

    if x>=10:
        print “More than 1 digit”

print “Finished”
```

exercise

- I have a number X
- If X is bigger than 10
    - if X is bigger than 100, print(“>100”)
    - if X is smaller than 100,print(“10-100”)
- If X is smaller than 10
    - if X is bigger than 0, print(“0-10”)
    - if X is smaller than 0, print(“<0”)

loop

- for loop
- while loop
- continue & break

loop

- continue
- what is continue?
- Stop this iteration, just go ahead and do the next iteration
- code following continue won’t be executed

```
for i in range(0,10):
    if i == 5:
        continue
    print(i)
```

loop

- break
- what is break
- Whenever a ‘break’ is encountered, this loop stops.

```
for i in range(0,10):
    if i == 5:
        break
    print(i)

print(“done”)
```

Nested Loop

- loop in loop
- given a 2-D list, print list
- L = [[1,2,3], [2,3,4], [5,6,7]]

Nested Loop

- continue in nested loop
- break in nested loop

```
for i in range(len(L)):
    value = L[i]
    for j in range(len(value)):
        print( L[ i ][ j ] )
        if L[ i ][ j ] == 5:
            break
print(“done”)
```