

Talking to the machine

Lecture 01.03

By Marina Barsky

Implementation: modern computers

- The basic set of primitive operations is implemented using **digital circuits**
- We have digital memory where we load **the data and the instructions**
- There is a processing unit which executes these instructions **one at a time**
- All the work of a computing machine is in reading next instruction from a known place in memory and sticking the result into another place

Language that machine understands

- Computers understand only basic instructions expressed as binary code: **sequence of zeroes and ones**
- High-level programming languages offer larger set of primitives and human-readable code
- We use [Python programming language](#) for this course to communicate with the computer

Why Python

- Clean syntax that is concise:

You can say a lot more with fewer words

- Compact language

You can carry all the language constructs in your head

- Powerful library of useful functions

This means that you can be productive quite easily and fast.

You will be spending more time solving real problems and writing code than grappling with the idiosyncrasies of the language

- Python is used by Google, Industrial Light & Magic, YouTube, and NASA to name a few

Python is **interpreted** language

- Each line is read and interpreted (executed) in turn
- There is no conversion of the entire program into machine language (unlike in **compiled** languages C and C++)
- That is why we can run separate commands in IDLE – interactive shell
- In IDLE we can also edit programs, save to a file and run

Like any new language

- Words
- Sentences
- Paragraphs
- Stories

Python language

- Words: values and variables
- Sentences: expressions and statements
- Paragraphs: blocks of code
- Stories: programs with a purpose

Sample story (dialog)

```
print("Welcome!")  
g = input("Guess the number: ")  
guess = int(g)  
if guess == 5:  
    print("You win!")  
else:  
    print("You lose!")  
print("Game over!")
```


Sentences

x = **2** ← Assignment statement

x = **x** + **2** ← Assignment with expression

print (**x**) ← Print statement

Variable

Operator

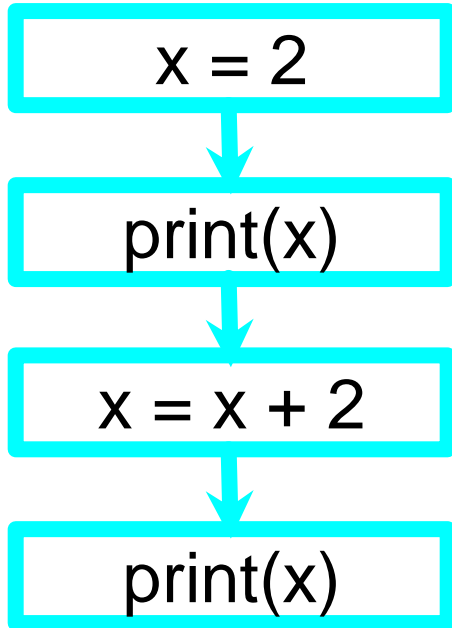
Value

Function

Program Flow

- Program is a sequence of instructions to be executed in order
- Some instructions are conditional - they may be skipped
- Sometimes instructions are to be repeated

Sequential Programs



Program:

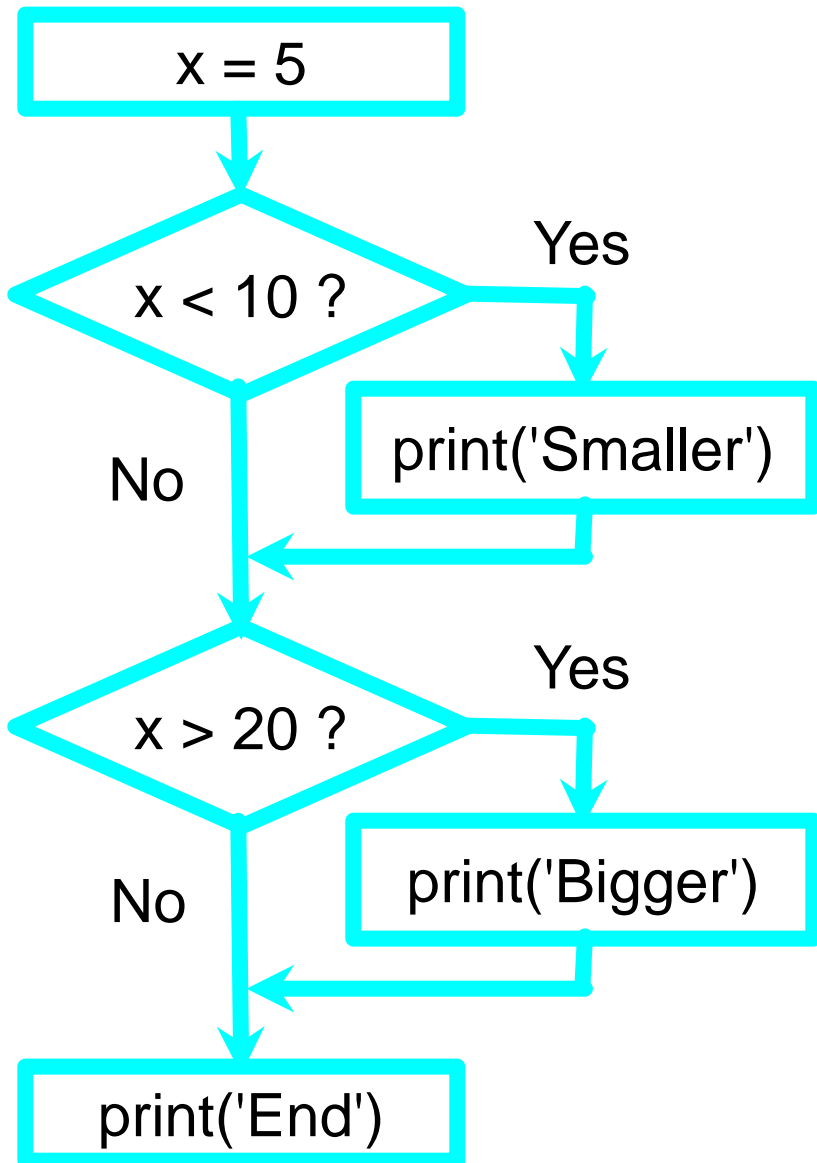
```
x = 2  
print(x)  
x = x + 2  
print(x)
```

Output:

2
4

When a program is running, it flows from one step to the next. As programmers, we set up “paths” for the program to follow.

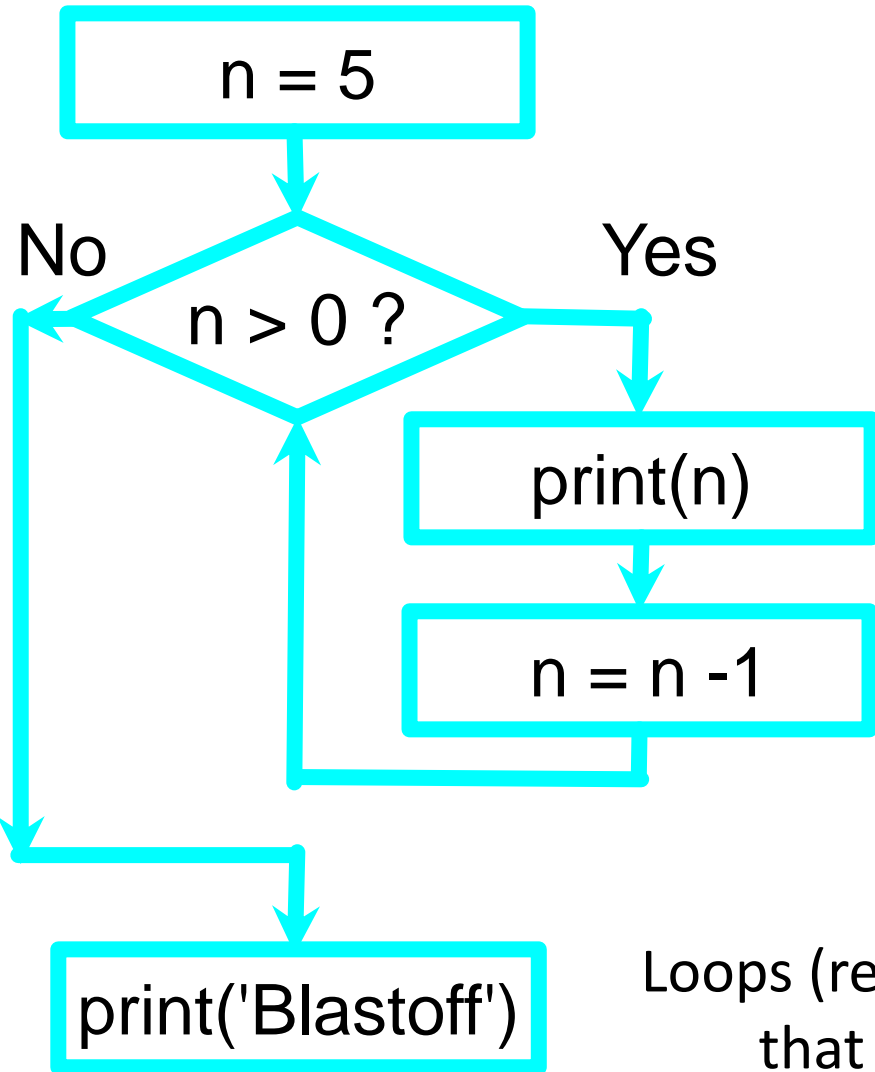
Conditional Steps



Program:

```
x = 5
if x < 10:
    print('Smaller')
if x > 20:
    print('Bigger')
print('End')
```

Repetitions



Program:

```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Blastoff!')
```

Output:

5
4
3
2
1
Blastoff!

Loops (repeated steps) have iteration variables that change each time through a loop

Summary

- This is a quick overview of the next two weeks
- We will revisit these concepts in the next lectures
- For now, focus on the big picture which is:

We are actually not conversing with the machine, but with the programmer who has put a piece of its intelligence into a program