1 Introduction

Friday, August 23, 2024 7:27 AM

- Computer Science:

¿What is Computer Science?

The subject of study is **Problems**

Fundamental Question:

¿what problems can be solved in an automated way and how?

Dijkstra.- "Comp. Science is no more about computers as astronomy is about telescopes or biology about microscopes"

"Automated way" → "Algorithm"

Algorithm.- a sequence of unambeguous instructions to solve a problem.

Solve.- Take some "input" data and produce some expected "output" data

E.G.- "Recipe" = Algorithm for human Cooking.

E.G.-

736 * 52

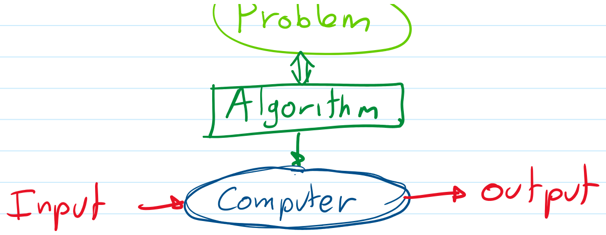
input.

```
   ' 3
   736
 *  52
 -----
  1472
 36800
 -----
 38272   output
```

Problem

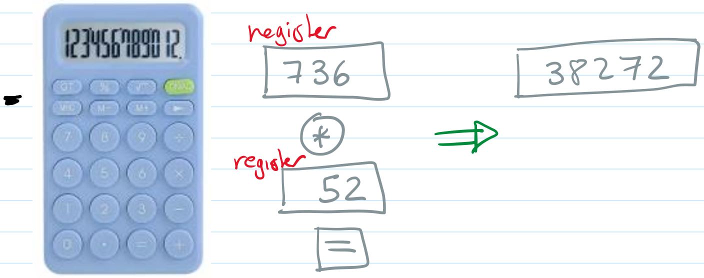
⇕

Algorithm

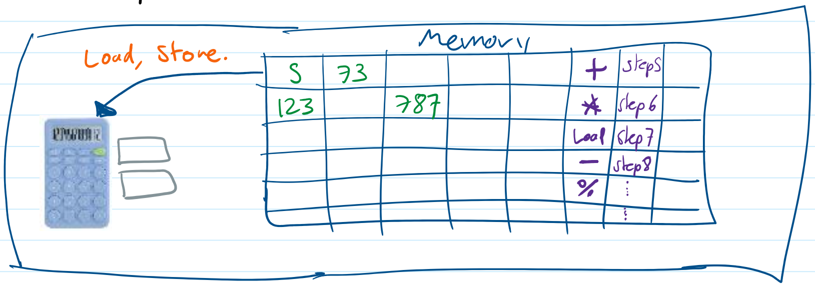


- Digital Computers

- Mental Model.



- A computer:



- The Algorithm is stored in nemary
- The input data is in memory
- The Algorithm modifies memory producing partial resuts
- The output data is produced into memory

Memory in modern Computers is "digital"

8   0001000

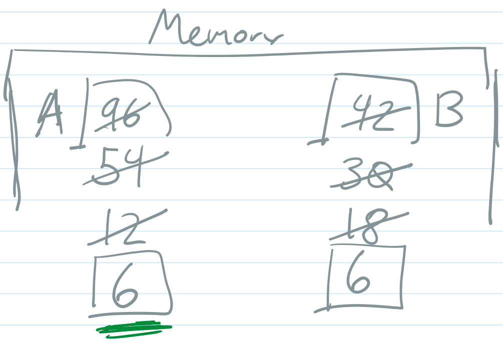
A   0110011
+   1110001

Recap:

- A computer is a "calculator w/ memory"
- A computer manipulates memory following a sequence of instructions: "program"
- The program itself is stored in memory
- A program encodes an "Algorithm"
- An algorithm is a sequence of unambiguous instructions to solve a problem.

E.G. An Algorithm:

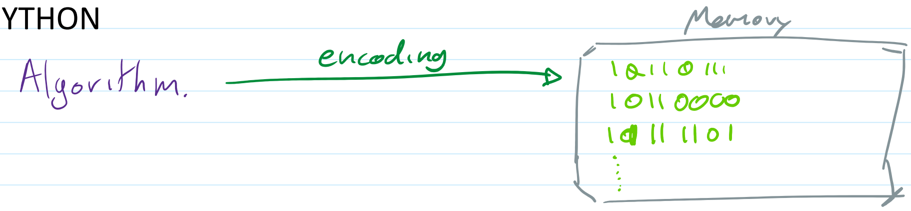
```
1.- Given 2 numbers, A and B
Repeat:
  2.- If A is equal to B, STOP
  3.- If A is greater than B:
        assign A-B to A.
      Else
        assign B-A to B.
Print A.
```

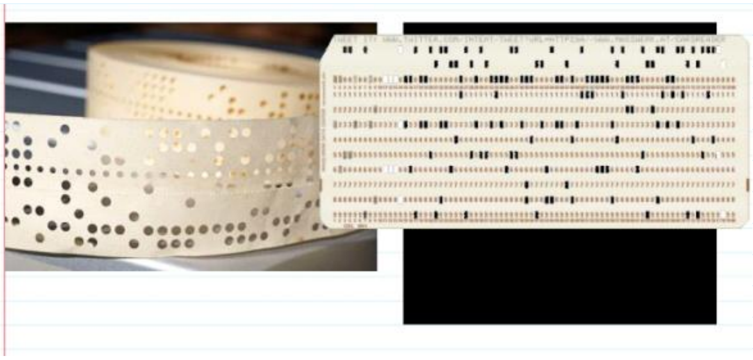


6 is the greatest common denominator of 96 and 42.

Euclid's Algorithm, (300 BC.)

- PYTHON





☒ "Programming Languages"

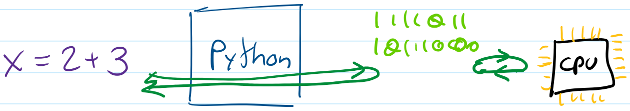
- A Synthetic language intended to write computer instructions
- A separate program (Compiler / Interpreter) that translates programs in the programming language to binary instructions.

'59 FORTRAN by John Backus.

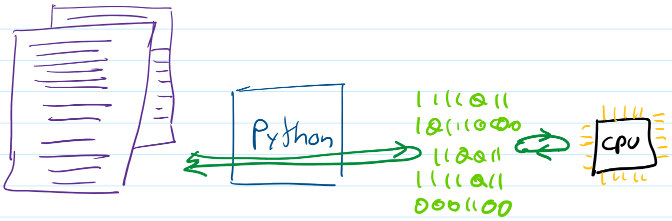
☒ '98 Python. Guido Van Rossum.

'2008 Python-3

- Interactive Mode.



- Batch Mode



"The quick brown fox swifty jumps over the lazy dog"

article noun adverb verb.

- Basic Concepts of Python:
  - Literals
    constant values which are part of the language.
    - Numbers
    - Strings.
  - Expressions

- Strings.

- Expressions

  Sentences made of operands and operators that are part of the language.

  Some operators. + - * / ( )

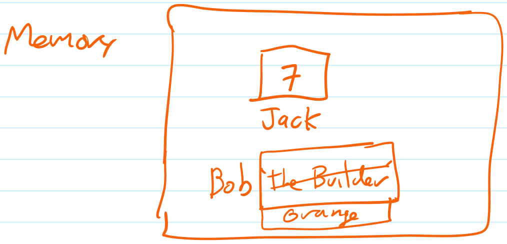
  // %

- Assignment

  Assigns a value to a named piece of memory
  ↳ a "variable"

  Syntax:

  name = expression.



  variables can be used as operands in expressions.

- Basic input/output.

  Syntax:

  print(expression) dispays vale of expression

  $\underbrace{\text{name = input()}}_{\text{Function.}}$ reads from keyboard and stores what is typed in name.

- Conditionals. (chap-4)

  Sentences that allow a program to make decisions about what instructions to execute.

- Repetition (chap-5)

  Sentences that allow a program to repeat some instructions, a fixed number of times, or until some condition is satisfied.

- Demo: Python batch mode

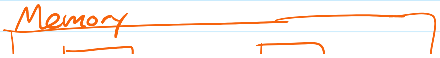
- A point about Assignment

  = Assignment is not algebra

  = is not "equals" in math

  e.g.

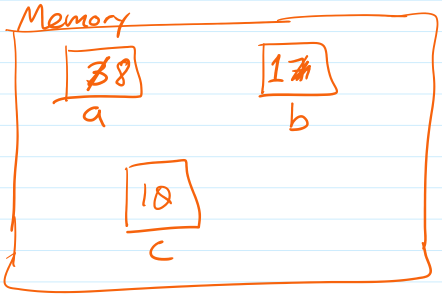
  A = 3



= is not equals in math

e.g.

```
a = 3
b = 7
c = a + b
a = 8
print(c)
b = 1
```



10

<=

:=

c is a + b

- Rules for names:

a c bob apple.

- Names can consist of letters, number and '_' underscore
- Names cannot begin with a number
- Names are case sensitive

Apple apple ~~2p~~ p23

_a_p_27

area a

vol_circle v

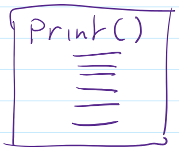
- There is a list of reserved names. int True

- A point about "functions":

print( )

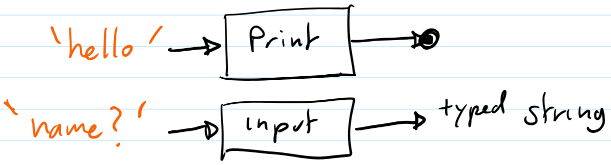
input( )

int( )



A named piece of code.

like a black box



int(x) takes a value x and returns an integer representation of x, if possible.

- A function can be used inside an expression the return value of the function will be used in-place of the function.

- More arithmetic operators:

+ - * / //

a ** b exponentiation.

+= "apply operator and assign"

+=   "apply operator and assign"
-=
/=
*=

eg. $X = X + 2 \iff x += 2$ "increment x by 2"

$X -= 2$ "decrement x by 2"

$X /= 2$ "half x" $\quad \underbrace{X = \underbrace{x}}\,/\,2$

- Boolean Expressions:

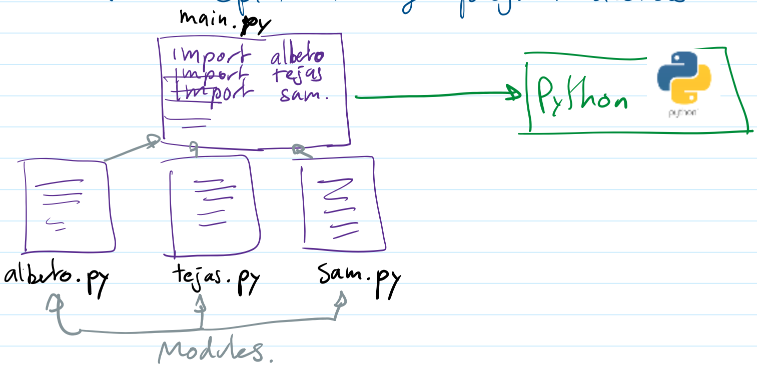
True False

operators

<	>	or
<=	>=	and
==	!=	not

- Basics of Modules:

idea: Split a large program accross multiple files

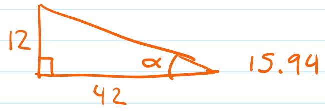


- Python has a laaaaarge collection of "Standard Modules"

- E.g. math module

cos( )	acos( )	radians( )
sin( )	asin( )	degrees( )
tan( )	atan( )	
		sqrt( )
factorial( )		log( )
gamma( )		pow( )

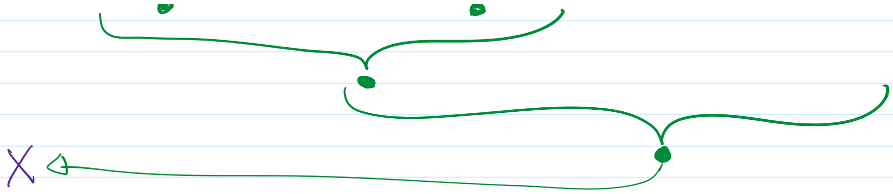
- Problem:



a = 12
b = 42

nonsense but an expression

$$x = \underbrace{\underbrace{\sin(\underbrace{a/b}_{\bullet})}_{\bullet} * \underbrace{\tan(\underbrace{a/3}_{\bullet})}_{\bullet}} + \underbrace{\mathrm{sqrt}(b)}_{\bullet}$$



- Eg: The Random modle.

import random

functions

arguments

random()	random number	[0..1)
randrange(a,b)	random number	[a..b)
randrange(b)	random number	[0..b)
randint(a,b)	random integer	[a...b]

- on "Pseudo-Random" numbers.

e.g. Marsenne Twister.

These numbers depend on an initial number called the seed

by default, this is the value of the clock.

—o— EOF