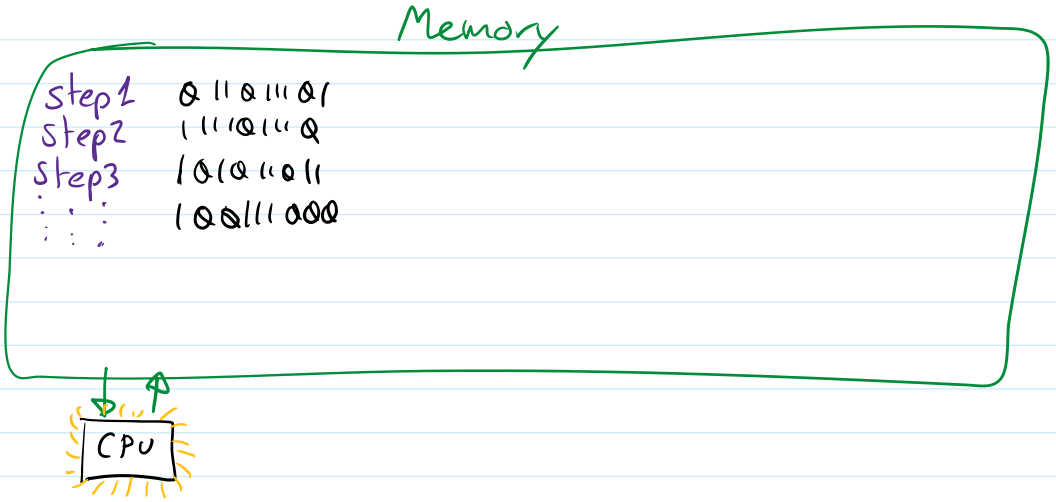
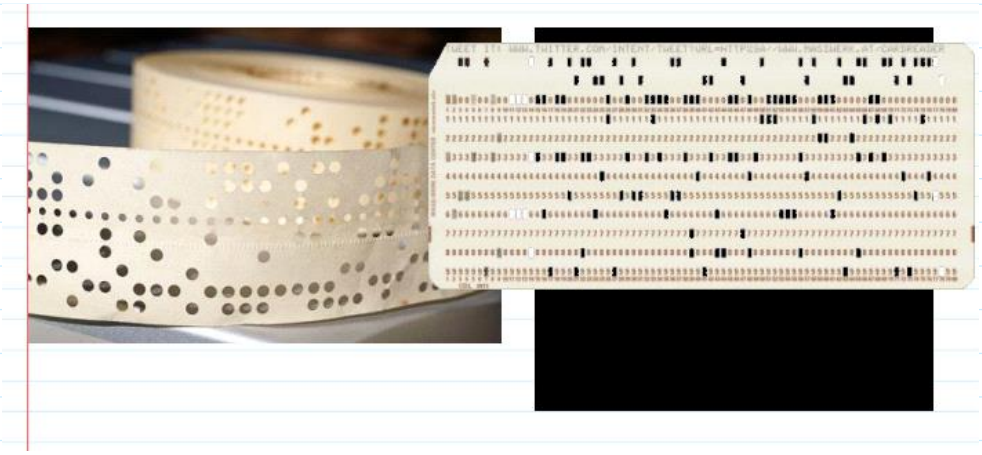
2 Python Basics

Friday, January 26, 2024 1:56 PM





- "Programming Languages"
  - Syntletic Language intended to write computer instructions.
  - A separate programs (compiler / Interpreter) translates the language to binary instructions.

'59 IBM FORTRAN
by John Backus.

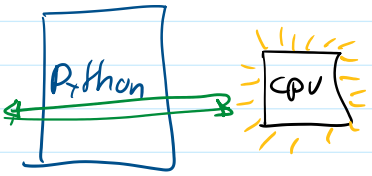
- Python
  '98 Guido Van Rossum.

'98 Guido Van Rossum.

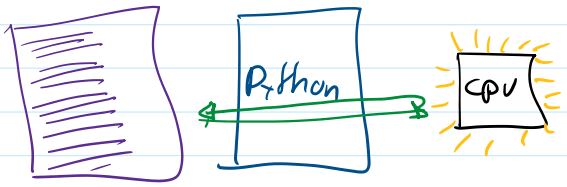
'2008 Python-3

Interactive Mode

$x = 2+3$



Batch Mode.



- Most Basic Python:
  - Literal: constant values part of the language.
    - Numbers
    - Strings.
  - Expressions:
    Sentences made of operands and operators that are part of the language.

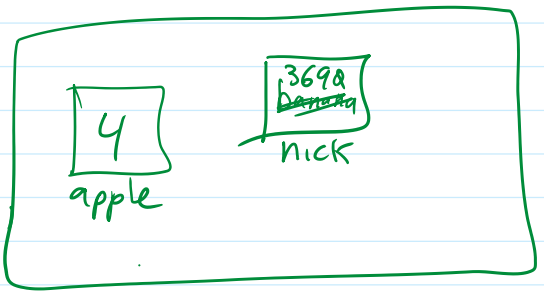
    + - * / ( )

    // % ← remainder
  - Assignment.
    assign a value to a (named) piece of memory

Syntax: name = expression

Syntax: name = expression

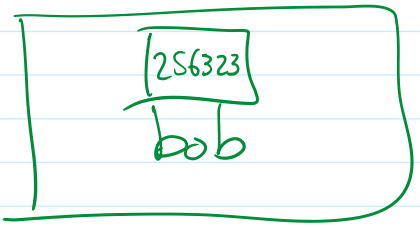
variable.



- basic input/output.

print(expression)
displays expression into screen.

name = input()
read from keyboard and store in name.



- Conditionals (Chap-4)

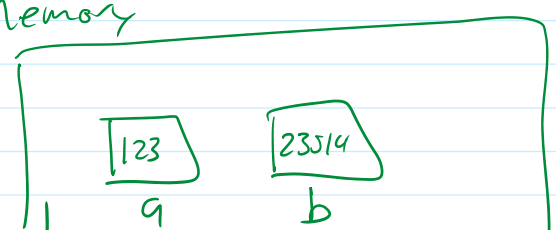
Programs can make decisions.

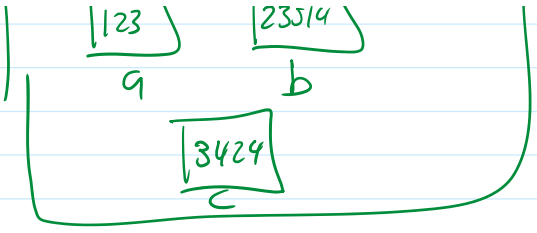
- Repetition (Chap-5)

Programs can reapeat a block of instructions
a fixed number of times
or until some condition is true/false.

- Demo: batch/file mode.

Memory

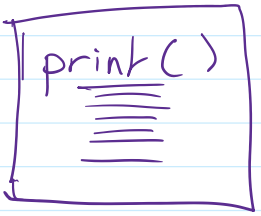




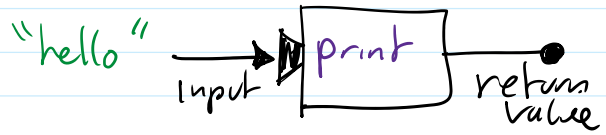
- A Little about "functions"

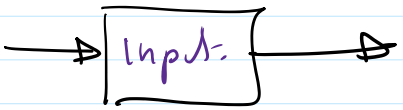
print()
input()

A named piece of code
A "black box"



print('hello')





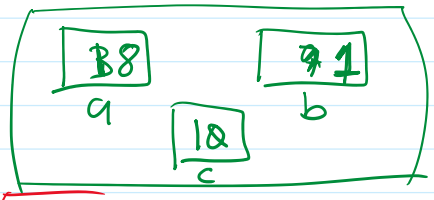
- function int(x)

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

- Assignment is not Algebra.

= is not the "equals" from Math. ← :=

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



10

- Rules for names.

  a b c apple

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

  Apple apple ~~2p~~ p23

  orange

  a_p_27

  - There is a list of reserved names.

- More Arithmetic operators.

  a ** b exponentiation.

  +=
  -=
  /=
  *=

  apply operation and assign.

  e.g. x = x + 2 ≡ x += 2
  "increment x by 2"

  x -= 2 "decrement x by 2"
  x *= 2 "multiply x by 2"

- Boolean Expressions. < True / False.

  operators

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

- Entering Numbers.

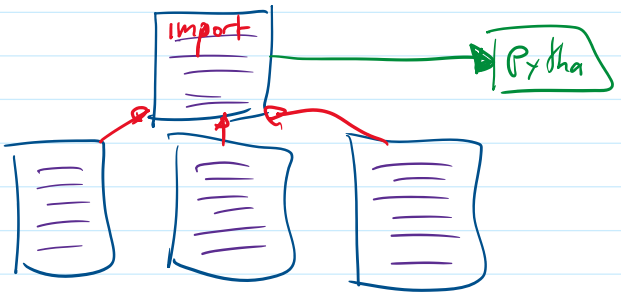
  x = input(msg)

  ~~35~~ 33
  age

  x = int(input(msg))

Basics of Modules:

- Split your program accross multiple files.



- "Standard Modules" pre-created program files.

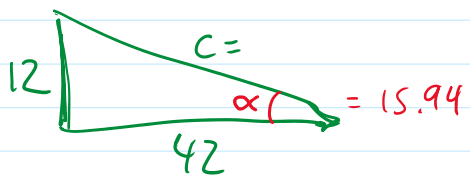
Eg. the math module.

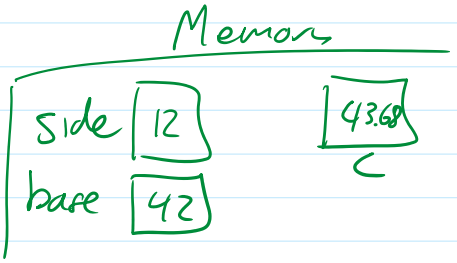
```
import math
=====

a = math.pi * r * r
=====
```

cos( )	acos( )	radians( )	converts degrees to radians
sin( )	asin( )	sqrt( )	square root
tan( )	atan( )	degrees( )	converts radians to degrees
factorial( )		log( )	natural logarithm.
gamma( )		pow( )	

eg.



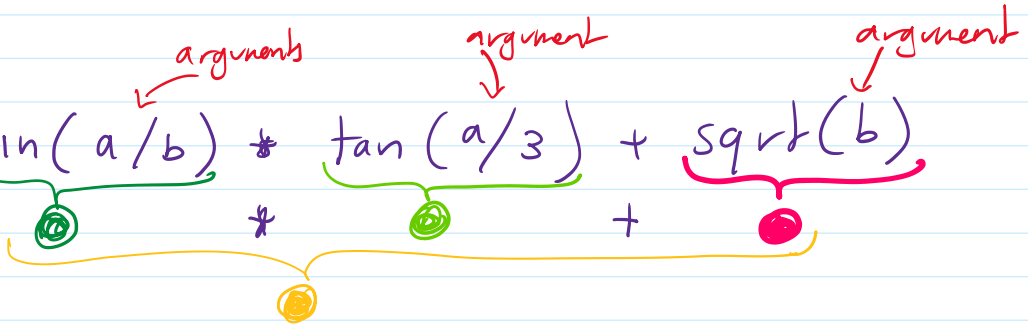


Note: In an expression, a function is replaced by its return value.

a = 12 argument argument

replaced by its return value.

```
a = 12
b = 42
x = sin(a/b) * tan(a/3) + sqrt(b)
X =          *          +
X =
```



Eg: The Random Module

```
import random
```

`random()` random number in [0..1)

- `randrange(a, b)` random integer in [a...b)
- `randrange(b)` random integer in [0..b)

`randint(a,b)` random integer in [a..b]

- on "Pseudo-Random" numbers.
  e.g "Mersenne Twister"

These formulas depend on an initial number called the "seed"

↑ by default, this is the value of the clock.

EOF