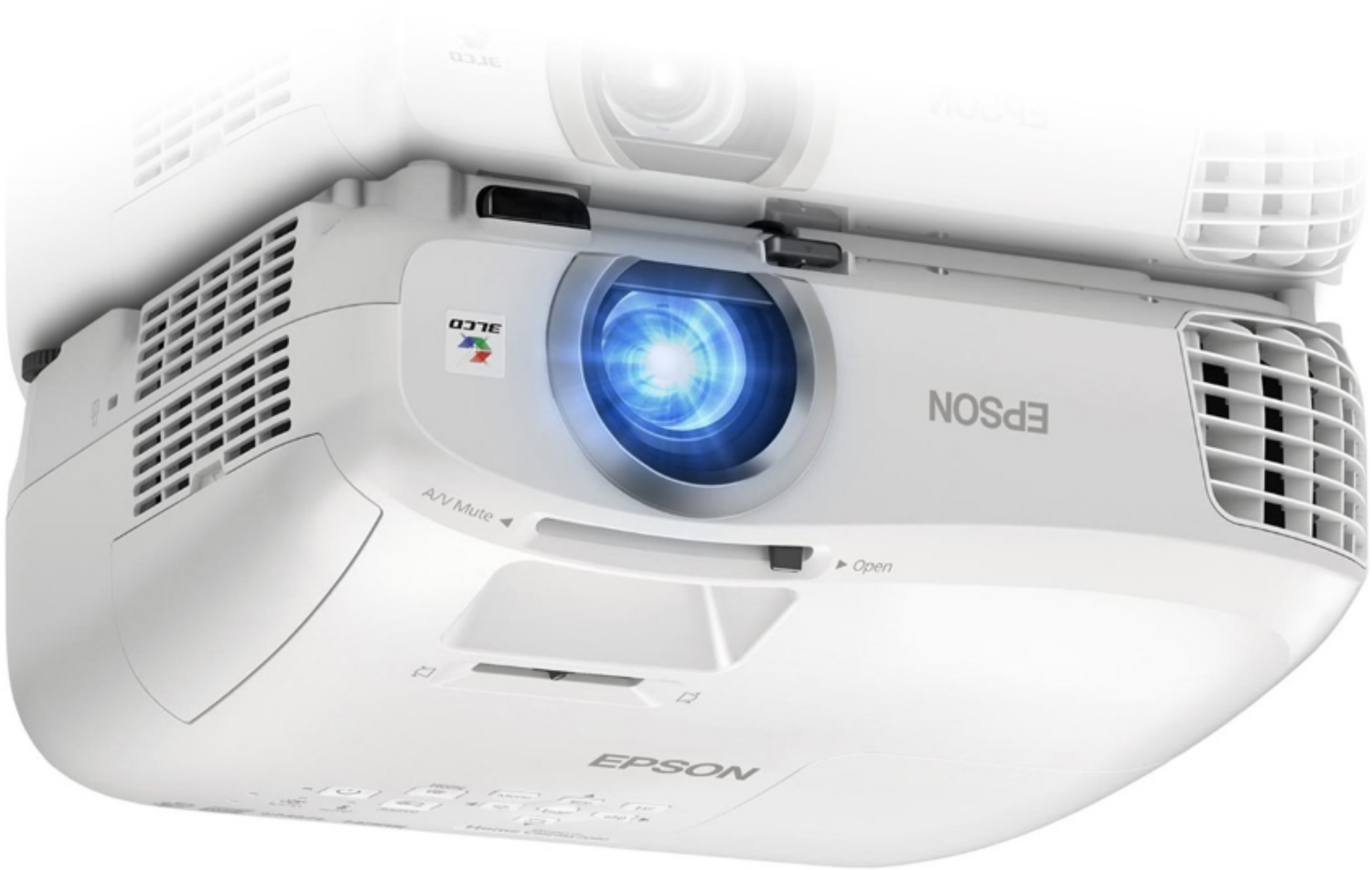
s = ‘abc’
s == ‘abc’
s > ‘def’
len(s)
s[0]
s[3]

s[::-1]
s[0:2]
s[-1]

s[start:end:step]

strings and loops


3LCD
EPSON
A/V Mute
Open
EPSON

Abstraction - suppressing the details

Decomposition - breaking down the problem

Modules

used to break up code
self-contained
re-usable

Functions

name
parameters
doc string
body - the sequence of instructions

function

```
def <name>(<parameters>):
    """docstring"""
    |
    |block of instructions
```

function

is_even(<int>)

variable scope

formal parameter gets bound to value of actual parameter when function is called

new scope/frame/environment is created when enter a function

scope is mapping of names to objects

Global scope

f

some code

x

3

z

4

f scope

x

3 which becomes 4

return vs print

return	print
only has a meaning inside a function	can be used outside a function
only one can be executed inside function	multiple print statements can be executed within function
code is not executed after return function	code is executed even after print statement
has an associated value given to function caller	has an associated value printed to console

some examples for functions

some important aspects for scope

variables defined outside a function
can be accessed inside

variables defined outside a function
cannot be modified inside

last example for scopes before the break

BREAK