Functions

a named sequence of statements

Accumulator

↳ repetition.

Function Decomposition

↳ like composite functions.

↳ f(g(x)) ex: .def square(x):
y = x * x
return y
• def sum_of_square(x,y,z):
a = square(x)
b = square(y)
z = square(z)
return a+b+c

Unit testing

↳ test case
↳ "assert"

types

1. Fruitful fx: return a value
2. Procedure: doesn't " "

→ local vs global variables
↓ within the fx
↓ outside the fx

→ Defining a fx

def ... ( ):
↳ parameters
For i in ...

→ Call a function

→ require arguments
(value that allows fx to do their job)

Flow of Execution summary

↳ order.

$y = b^p$.

square(x):
a = x^2
a.

result(square(5)
5^2

1, 4