Functions & abstraction

Python Reference · Vocabulary in context

Word	Example
function /ˈfʌŋkʃn/	A **function** returns a value.
define /dɪˈfaɪn/	Use def to **define** a function. def
call /kɔːl/	To **call** the function, write its name with brackets: greet(). greet()
return /rɪˈtɜːn/	**Return** sends a value back from the function.
parameter /pəˈræmɪtə/	In def area(r):, r is a **parameter**. def area(r): r
argument /ˈɑːgjuːmənt/	In area(5), 5 is the **argument**. area(5) 5
local /ˈləʊkl/	A **local** variable exists only inside its function.
scope /skəʊp/	The **scope** of a variable is where it can be used.
procedural abstraction /prəˈsiːdʒərəl əbˈstrækʃn/	**Procedural abstraction** lets you use a function without knowing how it works inside.
module /ˈmɒdjuːl/	math is a **module** full of useful functions. math
import /ɪmˈpɔːt/	Write import math to **import** the math module. import math math
default value /dɪˈfɒlt ˈvæljuː/	In def greet(name="friend"):, "friend" is the **default value**. def greet(name="friend"): "friend"