Programming Fundamentals

University Computer Science · Vocabulary in context

Word	Example
programming fundamentals /ˈprəʊgræmɪŋ ˌfʌndəˈmentlz/	Learn **programming fundamentals** by writing small programs.
type /taɪp/	The number 8 has the **type** int. 8 int
value /ˈvæljuː/	After x = 5, the **value** of x is 5. x = 5 x 5
variables /ˈveərɪəblz/	Name and Age are **variables**. Name Age
string /strɪŋ/	“Hello” is a **string**. “Hello”
selection /sɪˈlekʃn/	An IF statement is a kind of **selection**. IF
iteration /ˌɪtəˈreɪʃn/	A loop is a kind of **iteration**.
infinite loop /ˈɪnfɪnət luːp/	An **infinite loop** keeps running without stopping.
trace table /treɪs ˈteɪbl/	A **trace table** shows the value of each variable at each step.
function /ˈfʌŋkʃn/	A **function** returns a value.
return value /rɪˈtɜːn ˈvæljuː/	The function’s **return value** is the total.
parameter /pəˈræmɪtə/	A function’s **parameter** is the name for the input it receives.
argument /ˈɑːgjuːmənt/	In area(5), 5 is the **argument**. area(5) 5

exercises.lol exercises.com.cn

Word	Example
local scope /ˈləʊkl skəʊp/	A variable made inside a function has **local scope**: it only exists there.
list /lɪst/	[3, 5, 7] is a **list** of three numbers. [3, 5, 7]
dictionary /ˈdɪkʃənəri/	A **dictionary** stores pairs of keys and values.
tuple /ˈtuːpl/	A **tuple** is one row of a table.
set /set/	A **set** holds values with no repeats.
aliasing /ˈeɪlɪəsɪŋ/	With **aliasing**, two names point to the same list, so changing one changes both.
file /faɪl/	Save the scores in a **file**.
exceptions	Catch **exceptions**, such as a missing file, so the program does not crash.
decomposition /ˌdiːkɒmpəˈzɪʃn/	**Decomposition** breaks a big problem into small parts.
testing /ˈtestɪŋ/	Use **testing** to compare actual and expected answers.
debugging /ˈdiːbʌgɪŋ/	**Debugging** means finding and fixing mistakes in a program.