Selection and Iteration

AP Computer Science A · Vocabulary in context

Word	Example
Algorithms /ˈælgərɪðəmz/	**Algorithms** are step-by-step methods for solving a problem.
control structures /kənˈtrəʊl ˈstrʌktʃəz/	Sequence, selection and loops are **control structures**. 、
sequence /ˈsiːkwəns/	In a **sequence**, the steps run one after another.
selection /sɪˈlekʃn/	An IF statement is a kind of **selection**. IF
iteration /ˌɪtəˈreɪʃn/	A loop is a kind of **iteration**.
boolean expression /ˈbuːlɪən ekˈspreʃn/	x > 3 is a **boolean expression**. x > 3
relational operators /rɪˈleɪʃənl ˈɒpəreɪtəz/	<, > and == are **relational operators**. <、> ==
primitive values	== compares **primitive values** such as int and double. == int double
object references	For objects, == compares **object references**, not contents. ==
if statement /ɪf ˈsteɪtmənt/	An **if statement** runs code only when the condition is true.
condition /kənˈdɪʃn/	The **condition** x > 10 decides whether the if block runs. x > 10 if
block /blɒk/	A **block** is the code inside curly braces { }. { }

exercises.lol exercises.com.cn

Word	Example
branch /bræntʃ/	Each **branch** of an if statement does something different. if
Logical operators /ˈlɒdʒɪkl ˈɒpəreɪtəz/	&&, || and ! are **logical operators**. &&、|| !
short-circuit evaluation /ʃɔːt ˈsɜːkɪt ɪˌvæljuːˈeɪʃn/	**Short-circuit evaluation** stops once the answer is known.
guard against	Short-circuit evaluation can **guard against** division by zero.
De Morgan's laws /də ˈmɔːgənz lɔːz/	**De Morgan's laws** change AND into OR. AND OR
equivalent /ɪˈkwɪvələnt/	!(a && b) is **equivalent** to !a || !b. !(a && b) !a || !b
truth table /truːθ ˈteɪbl/	A **truth table** shows the output for every input.
reference /ˈrefrəns/	A variable holds a **reference** to an object, not a copy.
aliases	a and b are **aliases**: two names for one list. a b
null /nʌl/	A reference set to **null** points to no object. null
while loop /waɪl luːp/	A **while loop** repeats while the condition is true.
infinite loop /ˈɪnfɪnət luːp/	while True: with no break makes an **infinite loop**. while True: break
initialization	A for loop's **initialization** runs once, before the first pass. for
update /ˈʌpdeɪt/	count = count + 1 will **update** the value of count. count = count + 1 count

Word	Example
integer /ˈɪntɪdʒə/	An **integer** is a whole number, which can be positive, negative or zero.
digits	n % 10 gives the last of a number's **digits**. n % 10
divisibility	For nonzero d, n % d == 0 tests **divisibility**. d n % d == 0
counter /ˈkaʊntə/	The **counter** goes up by 1 each time. 1
frequency /ˈfriːkwənsi/	A **frequency** of 50 Hz means 50 waves every second. 50 Hz 50
criterion /kraɪˈtɪərɪən/	Count the values that meet the **criterion**.
running total	A **running total** adds each value to a sum.
flag /flæg/	A **flag** variable records whether something has happened.
index /ˈɪndeks/	Use **index** 3 to read the third item in the array. 3
character /ˈkærɪktə/	"A" is a single **character**. "A"
occurrences	Count the **occurrences** of "a" in a word. "a"
nested loop /ˈnestɪd luːp/	A **nested loop** is a loop inside another loop.
inner loop /ˈɪnə luːp/	The **inner loop** runs fully each time the outer loop runs once.
outer loop /ˈaʊtə luːp/	The **outer loop** goes through each row of the grid.
grids	Nested loops process **grids** row by row.

exercises.lol exercises.com.cn

Word	Example
Run-time analysis /rʌn taɪm əˈnæləsɪs/	**Run-time analysis** shows how the time grows with input size.
linear /ˈlɪnɪə/	The relationship looks **linear**.
quadratic /kwɒˈdrætɪk/	Counting every pair among n items takes **quadratic** work. n
efficiency /ɪˈfɪʃənsi/	A lamp with 20% **efficiency** turns only a fifth of its energy into light. 20%
boundary conditions	Test the **boundary conditions**, such as an empty list.
off-by-one	Using i <= a.length can cause an **off-by-one** error. i <= a.length
bug /bʌg/	The team found a **bug** that made the score go below zero.
loop variable	Update the **loop variable** so the loop ends.