



Dictionary of Computer and Internet Terms

Sixth Edition



-
- More than 2500 key computer terms with definitions
 - Includes hundreds of words and expressions that apply specifically to the Internet
 - User-friendly descriptions of programming concepts, desktop and other applications, and much more
 - Filled with illustrations
-

Douglas Downing, Ph.D., Michael Covington, Ph.D., and
Melody Mauldin Covington

Dictionary of Computer and Internet Terms

Sixth Edition



© Copyright 1998 by Barron's Educational Series, Inc.
Prior editions © copyright 1996, 1995, 1992, 1989, and 1986
by Barron's Educational Series, Inc.

All rights reserved.

No part of this book may be reproduced in any form, by photostat, microfilm,
xerography, or any other means, or incorporated into any information
retrieval system, electronic or mechanical, without the written permission
of the copyright owner.

All inquiries should be addressed to:

Barron's Educational Series, Inc.
250 Wireless Boulevard
Hauppauge, New York 11788
<http://www.barronseduc.com>

Library of Congress Catalog Card No. 98-6984

International Standard Book No. 0-7641-0094-7

Library of Congress Cataloging-in-Publication Data

Downing, Douglas.

Dictionary of computer and Internet terms / Douglas A. Downing.

Michael A. Covington, Melody Mauldin Covington—6th ed.
p. cm.

First-4th eds. published under title: Dictionary of computer
terms.

ISBN 0-7641-0094-7

1. Computers—Dictionaries. 2. Internet (Computer network)—
Dictionaries. I. Covington, Michael A., 1957– II. Covington,
Melody Mauldin. III. Downing, Douglas. Dictionary of computer
terms. IV. Title.

QA76.15.D667 1998
004'.03—dc21

98-6984
CIP

PRINTED IN THE UNITED STATES OF AMERICA

9876

10	43	8	91	-5
X(1)	X(2)	X(3)	X(4)	X(5)

FIGURE 15. ARRAY (ONE-DIMENSIONAL)

You can store numbers in these elements with statements such as:

```

X(1) = 10
X(2) = 43
X(3) = 8
X(4) = 91
X(5) = -5

```

just as if each element were a separate variable. You can also use INPUT and READ statements on array elements just as if they were ordinary variables.

Arrays are useful because they let you use arithmetic to decide which element to use at any particular moment. For example, you can find the total of the numbers in the five-element array **X** by executing the statements:

```

TOTAL = 0
FOR I = 1 TO 5
  TOTAL = TOTAL + X(I)
NEXT I
PRINT TOTAL

```

Here **TOTAL** starts out as 0 and then gets each element of **X** added to it.

Arrays can have more than one dimension. For example, the declaration **DIM Y(3,5)** creates a 3×5 array whose elements are:

Y(1,1)	Y(1,2)	Y(1,3)	Y(1,4)	Y(1,5)
Y(2,1)	Y(2,2)	Y(2,3)	Y(2,4)	Y(2,5)
Y(3,1)	Y(3,2)	Y(3,3)	Y(3,4)	Y(3,5)

31	17	95	43	60
32	95	86	72	40
27	45	93	43	81

FIGURE 16. ARRAY (TWO-DIMENSIONAL, 3×5)

Multidimensional arrays are useful for storing tables of data, such as three test grades for each of five students. *See also* DATA STRUCTURES; SORT.

arrow keys keys that move the cursor up, down, or to the left or right. The effect of these keys depends on the software being used. In a GUI environment, the arrow keys are basically an alternative to a mouse. Some drawing environments let you *NUDGE* the selected object with the arrow keys, giving you greater precision. Touch typists sometimes prefer the arrow keys to a mouse because it allows them to keep their hands on the keyboard. *See* *NUDGE*; *KEYBOARD* (diagram); *MOUSE*.

Arrow keys do not have ASCII codes. Instead, each of them is read in a way that depends on the particular computer and software. Be cautious about using arrow keys when communicating with another computer; even if your cursor moves in the intended way, the other computer may not understand what you are doing, unless you are emulating a standard type of terminal (such as VT-100) and have identified your terminal type to it.

artificial intelligence (AI) the use of computers to simulate human thinking. Artificial intelligence is concerned with building computer programs that can solve problems creatively, rather than simply working through the steps of a solution designed by the programmer.

For example, consider computer game playing. Some games, such as tic-tac-toe, are so simple that the programmer can specify in advance a procedure that guarantees that the computer will play a perfect game. With a game such as chess, however, no such procedure is known; the computer must instead use a *heuristic*, that is, a procedure for discovering and evaluating good moves.

One possible heuristic for chess would be for the computer to identify every possible move from a given position and then evaluate the moves by calculating, for each one, all the possible ways the game could proceed. Chess is so complicated that this would take an impossibly long time (millions of years with present-day computers).

A better strategy would be to take shortcuts. Calculating only five or six moves into the future is sufficient to eliminate most of the possible moves as not worth pursuing. The rest can be evaluated on the basis of general principles about board positions. In fact, an ideal heuristic chess-playing machine would be able to modify its own strategy on the basis of experience. Like a human chess player, it would realize that its opponent is also following a heuristic and would try to predict her behavior.

One of the main problems of AI is how to represent knowledge in the computer in a form that can be *used* rather than merely reproduced. In fact, some workers define AI as the construction of computer programs that utilize a knowledge base. A computer that gives the call number of a library book is not displaying artificial intelligence; it is merely echoing back what was put into it. Artificial intelligence would