



Chapter 25



VARIABLES IN PYTHON

VARIABLES are used to store information. Storing information is useful when you want to reuse that information many times or use it at different times while running a program. They save you from entering the same block of code over and over.

In Python, the basic **DATA TYPES** that can be stored by variables include:

DATA TYPE

The kind of value the data is

- strings
- integers
- lists
- Boolean values



Variables can store values (content or data), like a **STRING**, number, **LIST**, or **BOOLEAN**.

LIST

An ordered sequence of values

USING VARIABLES

To assign a value to a variable, use this format:

identifier = value

The **IDENTIFIER** is the name of the variable.

STRING

A data type that represents text

BOOLEAN

A data type that represents either "True" or "False"

FOR EXAMPLE, in a game where the player gets three lives, you could name a variable "lives" and assign it a value of 3:

lives = 3

VALUE

IDENTIFIER

In Python, the "=" sign means **is assigned to**.

Read "lives = 3" as "3 is assigned to lives."

After creating the variable and assigning it the value of 3, you can reuse the lives variable throughout the program.

Rules for Naming Python Variables

When coming up with the variable identifier, or name, you must follow these rules:

GOOD



- * Use numbers, uppercase letters, lowercase letters, and underscores (_).
- * Start the identifier with a letter.
- * Keep your identifier short and simple, and have it describe the variable it's used for.

Score



Mice_4

lives

When coming up with the variable identifier, or name, **do not** do this:

BAD



- * Include spaces or symbols (like -, /, #, and @).
- * Use key words that already mean something else in Python (like "print").
- * Use the capital letters *O* or *I* or a lowercase *l*, because those can get confusing—a capital *O* looks like the number 0.
- * Start your variable with a number or symbol.

Key words in Python (not allowed as identifier names):

and	def	finally	in	pass	while
as	del	for	is	print	with
assert	elif	from	lambda	raise	yield
break	else	global	not	return	
class	except	if	or	try	
continue	exec	import			

When naming a variable, try to describe what it is.

FOR EXAMPLE, while you could name a variable for storing the number of lives a player has "j," you might forget what "j" means after you add in many more variables.



But if you name the variable "lives," you'll remember that the variable stores the number of lives for the player.

Also, keep variable names short; longer names lead to more typos and wasted time typing out the extra characters.

FOR EXAMPLE, although a variable called `the_number_of_lives_the_player_has_left` describes the data it stores, it's better to shorten it to just `lives`.



Formatting Variable Names

Programmers often use **NAMING CONVENTIONS** to make their code easier to read. A naming convention is a format for naming things like variables. Python will run correctly no matter which naming convention you pick. However, it's best to stick to one way in order to make your code easier to read.

Common naming conventions include:

DESCRIPTION	EXAMPLES
Single lowercase letter	x
Single uppercase letter	X
All lowercase	treasure
Lowercase and underscores	x_position
All uppercase	CAPTAIN



For Python, many people prefer to use **lowercase_and_underscores** when naming variables.

Uppercase and
underscores

SHIP_NAME



Capitalized words
(Pascal case)

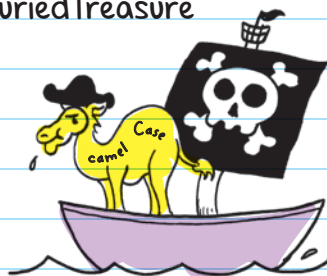
JollyRoger

Capitalize the first letter
of every word in the name.

Mixed case (camel case)

buriedTreasure

Capitalize only the
first letter of the
second word in a
name. Think of it like
a camel's hump in the
middle of the name.



Capitalized words with
underscores

Polly_Want_A_Cracker



CHECK YOUR KNOWLEDGE

1. Declare a variable called `winning_number` and give it a value of 1001.
2. Which of these are NOT allowed as Python variable names?
 - A. `@UserName`
 - B. `handle`
 - C. `Num_posts`
 - D. `FriendCount`
3. In Python, what does the `"="` symbol mean?
4. Write a two-word variable using camel case.
5. Why might it be a good idea to stick with the same naming convention in a program you're making?

6. What's wrong with the following variable name?
date/Time

7. What's wrong with using the following variable name?
continue

8. What's wrong with the following variable name?
1001I1



CHECK YOUR ANSWERS



1. `winning_number = 1001`

2. A

3. It means "is assigned to."

4. One answer would be: `exampleVariable`

5. Using the same naming convention will help you stay organized and make your code easier to read because all your variables will follow the same format.

6. Python does not allow slashes in variable names.

7. "continue" is a key word in Python. Key words are reserved and cannot be used as variable names.

8. It's hard to read because the 1s, Os, 0s, Is, and /s look similar.