

Plotting and Programming in Python (../)



Variables and Assignment



Teaching: 10 min Exercises: 10 min Questions

• How can I store data in programs?

Objectives

- Write programs that assign scalar values to variables and perform calculations with those values.
- Correctly trace value changes in programs that use scalar assignment.

Use variables to store values.

- Variables are names for values.
- In Python the = symbol assigns the value on the right to the name on the left.
- The variable is created when a value is assigned to it.
- Here, Python assigns an age to a variable age and a name in quotes to a variable first_name .

Python		
age = 42 first_name = 'Ahmed'		

- Variable names
 - can only contain letters, digits, and underscore _ (typically used to separate words in long variable names)
 - cannot start with a digit
 - are case sensitive (age, Age and AGE are three different variables)
- Variable names that start with underscores like __alistairs_real_age have a special meaning so we won't do that until we understand the convention.

Use print to display values.

- Python has a built-in function called print that prints things as text.
- Call the function (i.e., tell Python to run it) by using its name.
- Provide values to the function (i.e., the things to print) in parentheses.

- To add a string to the printout, wrap the string in single or double quotes.
- The values passed to the function are called arguments

Python

```
print(first_name, 'is', age, 'years old')
```

Output

```
Ahmed is 42 years old
```

- print automatically puts a single space between items to separate them.
- And wraps around to a new line at the end.

Variables must be created before they are used.

• If a variable doesn't exist yet, or if the name has been mis-spelled, Python reports an error. (Unlike some languages, which "guess" a default value.)

Python

print(last_name)

Error	
<pre>NameError <ipython-input-1-c1fbb4e96102> in <module>> 1 print(last_name)</module></ipython-input-1-c1fbb4e96102></pre>	Traceback (most recent call last) •()
NameError: name 'last_name' is not defined	i

- The last line of an error message is usually the most informative.
- We will look at error messages in detail later (../15-scope/#reading-error-messages).

✗ Variables Persist Between Cells

Be aware that it is the *order* of execution of cells that is important in a Jupyter notebook, not the order in which they appear. Python will remember *all* the code that was run previously, including any variables you have defined, irrespective of the order in the notebook. Therefore if you define variables lower down the notebook and then (re)run cells further up, those defined further down will still be present. As an example, create two cells with the following content, in this order:

Python

print(myval)

Python

myval = 1

If you execute this in order, the first cell will give an error. However, if you run the first cell *after* the second cell it will print out 1. To prevent confusion, it can be helpful to use the Kernel -> Restart & Run All option which clears the interpreter and runs everything from a clean slate going top to bottom.

Variables can be used in calculations.

- · We can use variables in calculations just as if they were values.
 - Remember, we assigned the value 42 to age a few lines ago.

Python

```
age = age + 3
print('Age in three years:', age)
```

Output

Age in three years: 45

Use an index to get a single character from a string.

- The characters (individual letters, numbers, and so on) in a string are ordered. For example, the string 'AB' is not the same as 'BA'. Because of this ordering, we can treat the string as a list of characters.
- Each position in the string (first, second, etc.) is given a number. This number is called an **index** or sometimes a subscript.
- Indices are numbered from 0.
- Use the position's index in square brackets to get the character at that position.

print(atom_name[0])



Python

atom_name = 'helium'
print(atom_name[0])

Output

h

Use a slice to get a substring.

- A part of a string is called a **substring**. A substring can be as short as a single character.
- An item in a list is called an element. Whenever we treat a string as if it were a list, the string's elements are its individual characters.
- A slice is a part of a string (or, more generally, any list-like thing).
- We take a slice by using [start:stop], where start is replaced with the index of the first element we want and stop is replaced with the index of the element just after the last element we want.
- Mathematically, you might say that a slice selects [start:stop).
- The difference between stop and start is the slice's length.
- Taking a slice does not change the contents of the original string. Instead, the slice is a copy of part of the original string.

Python

atom_name = 'sodium'
print(atom_name[0:3])

Output

sod

Use the built-in function len to find the length of a string.

Python	
<pre>print(len('helium'))</pre>	
Output	
6	
Output 6	

• Nested functions are evaluated from the inside out, like in mathematics.

Python is case-sensitive.

- Python thinks that upper- and lower-case letters are different, so Name and name are different variables.
- There are conventions for using upper-case letters at the start of variable names so we will use lower-case letters for now.

Use meaningful variable names.

• Python doesn't care what you call variables as long as they obey the rules (alphanumeric characters and the underscore).

Python

```
flabadab = 42
ewr_422_yY = 'Ahmed'
print(ewr_422_yY, 'is', flabadab, 'years old')
```

- Use meaningful variable names to help other people understand what the program does.
- The most important "other person" is your future self.

Swapping Values

Fill the table showing the values of the variables in this program after each statement is executed.

Python				
# Command	# Value of x	# Value of y	# Value of swap	#
x = 1.0	#	#	#	#
y = 3.0	#	#	#	#
swap = x	#	#	#	#
x = y	#	#	#	#
y = swap	#	#	#	#

Predicting Values
What is the final value of position in the program below? (Try to predict the value without running the program, then
check your prediction.)
Python
initial = 'left' position = initial initial = 'right'
Solution
,✔ Challenge
If you assign $a = 123$, what happens if you try to get the second digit of a via $a[1]$?
Solution 🖸
🖍 Choosing a Name
Which is a better variable name, m, min, or minutes ? Why? Hint: think about which code you would rather inherit from someone who is leaving the lab:
1. $ts = m * 60 + s$
2. tot_sec = min * 60 + sec 3. total_seconds = minutes * 60 + seconds
Solution Solution
Slicing practice
What does the following program print?
Python
atom_name = 'carbon' print('atom_name[1:3] is:', atom_name[1:3])
Solution 🖸

✓ Slicing concepts

- 1. What does thing[low:high] do?
- 2. What does thing[low:] (without a value after the colon) do?
- 3. What does thing[:high] (without a value before the colon) do?
- 4. What does thing[:] (just a colon) do?
- 5. What does thing [number: some-negative-number] do?
- 6. What happens when you choose a high value which is out of range? (i.e., try $atom_name[0:15]$)

Solutions	◙
-----------	---

Key Points

- Use variables to store values.
- Use print to display values.
- Variables persist between cells.
- Variables must be created before they are used.
- Variables can be used in calculations.
- Use an index to get a single character from a string.
- Use a slice to get a substring.
- Use the built-in function len to find the length of a string.
- Python is case-sensitive.
- Use meaningful variable names.

< (../01runquit/index.html)

(../03 types conv /inde

Licensed under CC-BY 4.0 () 2018–2021 by The Carpentries (https://carpentries.org/) Licensed under CC-BY 4.0 () 2016–2018 by Software Carpentry Foundation (https://software-carpentry.org)

Edit on GitHub (https://github.com/swcarpentry/python-novice-gapminder/edit/gh-pages/_episodes/02-variables.md) / Contributing (https://github.com/swcarpentry/python-novice-gapminder/blob/gh-pages/CONTRIBUTING.md) / Source (https://github.com/swcarpentry/python-novice-gapminder/) / Cite (https://github.com/swcarpentry/python-novice-gapminder /blob/gh-pages/CITATION) / Contact (mailto:team@carpentries.org)

Using The Carpentries style (https://github.com/carpentries/styles/) version 9.5.3 (https://github.com/carpentries/styles /releases/tag/v9.5.3).