# Python Level 1

Table of Contents

[Python Level 1 1](#_Toc69909277)

[Course Readings 2](#_Toc69909278)

[Reserved Words 2](#_Toc69909279)

[Your First Program 2](#_Toc69909280)

[Variables 3](#_Toc69909281)

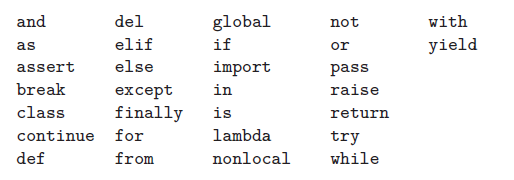
[Getting Input from the User 3](#_Toc69909282)

## Course Readings

* <https://www.py4e.com/>
  + Lessons => <https://www.py4e.com/lessons>
  + Book => <https://www.py4e.com/book>
* Youtube Course => <https://youtu.be/_uQrJ0TkZlc?list=PLF-r7uGjIN7CZ-zMa-YOcU4Bi4otMnTgL>

## Reserved Words

* Python’s reserve words are as follows:



## Your First Program

* Our first program will be a simple “Hello World” program:
  + print ('Hello World')
* Let’s add a few more print statements:
  + print ('Hello World')
  + print ('And hello Theo')
  + print ('And hello Pisha')
* If we run the following again, you will see the order in which the code was executed – in this case, line by line, starting from the top.
* Now, let’s add the following line of code to our list:
  + print ('\*' \* 10)
  + When we run the code we see that the “\*” icon has been duplicated x10. The code itself is called an **expression** – a piece of code that produces a value

## Variables

* **Note: Variables in python are *case sensitive*.** It is considered best-practices to define all variables in lower cases and, if made up of multiple words, separated by an underscore.
* Variables in python are written using the following syntax:
  + x = 10
  + x=10
  + (spaces will not affect the running of the code)
* What if we were to update the value of our x-variable, like so, what would get printed:
  + x=10
  + x=11
  + print(x)
  + Because we *know* python processes code line-by-line, from the top down, the x-variable will be *reset* to it’s *latest* value – in this case 11
* So far we have been dealing with whole integers, but python can also process float-point operators:
  + x=1.1
  + y=1.2
  + print(x\*y)
  + This will still produce the valid result of 1.32
* Python can also use Boolean logic for variables:
  + # Boolean logic (True and False need to be capitalized)
  + is\_this\_a\_program = True
  + print(is\_this\_a\_program)
  + **Note: True and False** **need to be capitalized variables when Boolean logic is employed**

## Getting Input from the User

* To gather input from the user we use the following syntax:
  + name = input('What is your name? ')
  + print('Hi ' + name)
  + In this example:
    - the input value is stored to the variable name
    - We then print a message “Hi” and concatenate the name that the user input

## Type Conversion

* To illustrate this section, we are going to write a program that will:
  + Ask the year we were born in
  + Calculate our age

Print it in the terminal

* year = input('What year were you born? ')
* ageCalculator = 2021 - year
* print('Your are ' + ageCalculator + ' years old')
* If we run this program, we will get the following error: “Type Error. Unsupported operand…” This is because, though the user will input a number, anything that is typed into the termina **is treated as a string** – even if it is a number.
* In this case our year variable, to python, looks like the following:
  + year = ‘1987’
* To properly process a mathematical expression, we need to first ensure that *both* our variables are integers, and so we need to convert the **string** value to an **integer** value, and there are a variety of built-in conversion functions we can leverage
  + **int()** function
    - To convert a **string** to an **integer**
  + **str()** function
    - To convert an **integer** to a **string**
  + **float()** function
    - To convert a **string** to a **float-point number**
  + **bool()** function
    - To convert a **string** to a **Boolean** value
* So, to have our code run properly, we need to amend it in the following way:
  + year = input('What year were you born? ')
  + age = 2021 - int(year)
  + print(age)

### Using Python to Discover the Variable Type

* In addition to converting a given variable to a type, we can use python to tell us what variable type the current variable exists as. To do this we can use the **type()** function
  + year = input('What year were you born? ')
  + print(type(year))
  + age = 2021 - int(year)
  + print(type(age))
* If we run this code, we will get the following result:

