

Splufic Python Training

Welcome to the instructor-led Training for week one day two

Python Modes

- **Interactive Mode:** used when you simply want to test a code or a statement on a line-by-line basis or when you're exploring its features.
 - Using command line
 - Using python IDLE (Integrated Development and Learning Environment)
- **Script Mode:** used to interpret an entire file of statements or application program.

Topics Covered

- Identifiers, Variables, operators, expressions and assignments
- Basic data types (Boolean, int, float, string)
- The list container type
- Implicit and Explicit type conversion
- 1-way and 2-way conditional statements

Comments

- Comments makes it helpful to put some notes within your code to describe what it does.
- It is very handy when you have to review or revisit your program.
- It will also help another programmer who might need to go over the source code.
- '#' is used to make comments in Python
- You can also wrap multiline comments in triple quotes

algebraic operators

- Subtraction operator -,
- Addition +,
- multiplication *,
- Division /,
- Exponentiation **,
- Floor division //,
- Remainder %

Algebraic Functions/Expressions

- An algebraic function is a function that involves only algebraic operations, like, addition, subtraction, multiplication, and division, as well as fractional or rational exponents. Think of an algebraic function as a machine, where real numbers go in, mathematical operations occur, and other numbers come out.
- It always evaluate to a number, whether of type int or float or real or imaginary
- We call the numbers going into an algebraic function the input, x , and the number going out as the output, y .
- Any number can go into a function as long as it is not divided by zero or does not produce a negative square root.
- E.g.: abs, max, min

Boolean Expressions and Operators

- Boolean expressions are expressions that evaluate to one of two Boolean values: True or False.
- Comparison operators (such as < or >) are commonly used operators in Boolean expressions.
- E.g.:
 - Greater than >
 - Less than <
 - Equal to ==
 - Less or equal to <=
 - Greater or equal to >=
 - Not
 - And
 - Or

Class Work 1

Write Python algebraic expressions corresponding to the following statements:

- (a) The sum of the first five positive integers
- (b) The average age of Sara (age 23), Mark (age 19), and Fatima (age 31)
- (c) The number of times 73 goes into 403
- (d) The remainder when 403 is divided by 73
- (e) 2 to the 10th power
- (f) The absolute value of the difference between Sara's height (54 inches) and Mark's height (57 inches)
- (g) The lowest price among the following prices: \$34.99, \$29.95, and \$31.50

Class Work 2

Translate the following statements into Python Boolean expressions and evaluate them:

- (a) The sum of 2 and 2 is less than 4.
- (b) The value of $7 // 3$ is equal to $1 + 1$.
- (c) The sum of 3 squared and 4 squared is equal to 25.
- (d) The sum of 2, 4, and 6 is greater than 12.
- (e) 1387 is divisible by 19.
- (f) 31 is even. (Hint: what does the remainder when you divide by 2 tell you?)
- (g) The lowest price among \$34.99, \$29.95, and \$31.50 is less than \$30.00.

Keywords

- keywords are reserved words in Python that should not be used as variable, constant, function name, or identifier in your code.
- E.g.
 - And
 - If
 - Else
 - Try
 - In
 - For
 - From
 - Not
 - Or
 - Pass
 - Print
 - raise
 - Return
 - try
 - While
 - With
 - Yield

Variables

- A variable is like a container that stores values that you can access or change.
- It is a way of pointing to a memory location used by a program.
- You can use variables to instruct the computer to save or retrieve data to and from this memory location.
- It helps us to uniquely identify and differentiate a variable from another

variable names: naming convention

- An identifier can be a combination of uppercase letters, lowercase letters, underscores, hyphen and digits (0-9).
- variable names cannot start with numbers
- Special characters such as %, @, and \$ are not allowed
- whenever a variable is Constant, use capital case
- variable names should be specific and detailed
- Python is a case-sensitive language and this behavior extends to identifiers
- You cannot use Python keywords as identifiers.

Variable Declaration, Initialization and Assignment

- A variable is **declared** when it is given a name
- A variable is **initialized** when it is **assigned** a value for the first time
- The general format of an **assignment** statement is: `<variable> = <expression>`
- In python, variables are declared and initialized at the same time
- E.g.:
 - `name = 'victor'`
 - `print(type(name))`

Class Work 3

Write Python statements that correspond to the actions below and execute them:

- (a) Assign integer value 3 to variable a.
- (b) Assign 4 to variable b.
- (c) Assign to variable c the value of expression $a * a + b * b$.

The background consists of a teal upper section and a black lower section, separated by a jagged horizontal line. The teal section has a fine, diagonal hatching pattern.

Questions?