

COL1000: Introduction to Programming

Nuts & Bolts of Python – Variables, Expressions, Assignments

Subodh Sharma | Lec 3 | Aug 05



Reminders!

- Do NOT share passwords!
 - Change it frequently
- Remember to check <https://moodlenew.iitd.ac.in>
 - For lab and lecture content

(Python) Programming Essentials: Variables

RECAP

- Storage locations in memory that holds a **value**
 - Can be changed in the program
 - Identified by a **name**. Eg: **x**, **userName**, ...
 - Values could be integers, strings, boolean etc.
 - The *kind of* values that a variable holds is called a **type** (eg: Int, Float, String, etc.)
 - Examples of **Primitive types**
 - 5 is of Int type, True is of Bool type, 5.6 is of Float type, '5' is of String type
 - Examples of **Compound types**
 - [1, 3, 7.2] is a List, (5, "svs) is a Tuple, {"name": "svs", "age": 100} is a Dictionary, {"R", "G", "B"} is a Set,
 - **Scope:** Until which part of the program can one access the variable

(Python) Programming Essentials: Statement

- Statement is an instruction that performs an action
- **Assignment Statement:** `x := 9, userName := "svs", ...`
- **I/O Statement:** `print("Hello"), name = input("enter your name")`
- **Control Statement:** `if C: <stmt>`
- **Function Statement:** `svs()`
 - **Function** is a reusable code of the program — to avoid repetition
 - `X := foo(y)` here **X** is the return value, **foo** is the name of the function, **y** is the argument
 - `input(..), print(..)` were function calls
- Expressions are computable parts of the statement. E.g: `y:= "svs" + "is" + "cool"` – RHS is an expression

(Python) Programming Essentials: Operators

- Arithmetic: `+, -, *, //, /, %`
- Relational: `==, <, >, <=, >=, !=, ...`
- Logical: `and, or, not`
- Membership: `in`
- Build compound: `(), [], {}, ' ', "`
- Operator precedence: Order in which expressions are to be evaluated!
 - $2 + 3^*5 \rightarrow r = (3^*5)$ followed by $r' = 2 + r$
 - $(2+3)^*5 \rightarrow$ explicitly specified grouping of expressions

(Python) Programming Essentials: Type Conversion

- Implicit conversion:

```
x, y = 2 , 3.5
```

```
z = x + y # implicit conversion – x is treated as a Float
```

- Explicit conversion:

```
s = "123"
```

```
print(s+1) # error
```

```
n = int(s)
```

```
print(n+1) # correct
```

- **Exercise:** `print(chr(52)), print(ord('F'))`

(Python) Programming Essentials: I/O & Function Statements

-
- The diagram illustrates the components of a Python function call. It features three rectangular boxes at the top: "Function name" (light gray), "Function Argument" (white), and "Function return value" (dark gray). Arrows point from each component to its corresponding part in the code below.
- Console input: `varName = input("some-prompt-string")`
 - Always returns a string type value
 - One could do type-conversion, if possible

```
age = int(input("enter your age: ")) # input validation? things can go wrong
```
 - Console output: `print("Name:", name, "Age:", age)`
 - **File I/O – Later**
 - Function statement: `range(stop)`, `range(start, stop)`, `range(start, stop, step)`
 - **User-defined functions – Later**
 - **Native functions:** `abs()`, `pow(x, y)`, `max()`, `min()`, `sum()`, `type()`,

Computing History Trivia

- **Variables:** inspired by the variables in Maths (FORTRAN 1957)
- **Functions:** As first-class citizens to be passed around as data (LISP 1958)
- **Compound Statements/Expressions:** With begin .. end or { .. } (ALGOL 1960)
- **Language, Recursion & Transformation:** Panini ~500 BC
- **First General-purpose Computer:** Konrad Zuse 1941 called Z3
- **First computer program:** Ada Lovelace 1840 on Analytical Engine (by Charles Babbage)