

Introduction to the basic data types in programming

^[1] Nagendra Singh, ^[2] Manish Kumar, ^[3] Vidhan Singh, ^[4] Dayanand Choudhary

^[1] Asst. Professor
Computer Science Engineering
Arya Institute of Engineering and Technology, Jaipur

^[2] Asst. Professor
Dept. of Applied Science
Arya Institute of Engineering Technology & Management, Jaipur

^[3] Science student
A.N.D Public School Khalpura Chapra, Bihar

^[4] Science student
Vivekanand Sr. Sec. School Fatehpur Shekhawati(Sikar) Raj.

Abstract: In the era of technology, the need for developing a language to communicate with computers led to the development of different programming languages. Computers cannot understand the commands in the normal human languages so we needed programming languages to communicate with computers and develop different apps, website and software programs. A crucial part for programming languages are data types. Data types are the classifications of data which tells the compiler or interpreter how the developer wants to insert the data.

Keywords: Data types, numeric data types, text data types, Boolean data types, composite data types, special data types, type casting and type coercion, dynamic and static typing.

1. Introduction

Programming languages are a type of medium used to communicate and give command to computers. To use the different types of programming languages effectively and make the compiler understand the task more efficiently one should know how data is classified in languages in the form of data types. Basic data types are the introduction to the most primary classification of data. This review paper gives a brief knowledge of these basic data types.

1. Numeric Data Types

- i) Integer type
- ii) Floating points type
- iii) Complex numbers

2. Text Data Types

- i) Characters
- ii) String

3. Boolean Data Types

- i) Boolean algebra
- ii) Logical operations
- iii) Conditional statements

4. Composite Data Types

- i) Arrays
- ii) Lists
- iii) Tuples

5. Special Data Types

- i) None
- ii) Enumerations
- iii) Custom data types

6. Type Casting and Type Coercion

- i) Implicit and explicit casting

ii) Type coercion in expressions

7. Type Safety and Compatibility

i) Strongly typed and weakly typed languages

ii) Type inference

8. Dynamic Typing and Static Typing

i) Dynamic typing

ii) Static typing

2. Previous Review

- 1) **Early Research:** The theory of data types has been an elemental for initial programming languages like Fortran and cobol, Integers, floating point numbers, and characters were the form of classification of data. The introduction and defining of the data types and its representation in the systems was the only focus in the past researches.
- 2) **Type Systems:** Static and dynamic were the two systems explored during the past researches. The enforcement of type checking at the moment of compilation in languages like C and Java is known as Static typing, whereas, the inspection of type at runtime in languages like Python and JavaScript is known as dynamic typing. The exchange between the approaches of these systems were being examined by the researchers on the basis of performance and safety.
- 3) **User-Defined Data Types:** The user-defined data types were offered limited support in the initial programming languages. The invention of custom data types with the help of structures, objects and classes through succeeding researches. To advance in this field, Object-oriented programming languages such as C++ and Small-talk play a crucial role.
- 4) **Type Inference:** In the evolution of type inference blueprint, compiler itself deduces the type of a variable being a remarkable research area. This excludes all the explicit type annotations by allowing expressive and compressed code. The significant examples that use type inference are Haskell and ML.
- 5) **Data Abstraction:** Abstract data types (ADT's) and encapsulation was the main focus in the data abstraction research. Besides, providing a well-defined interface for data manipulation, it hides the elementary implementation details. Barbara Liskov and Stephen Zilles were the two personalities who popularized the concept of adts, in their 1974 paper.
- 6) **Generic Types:** The research on generic programming was escorted through the need for generic data type capable of working with various data while working with type safety. Templates and generics were introduced by languages like C++ and Java, respectively, to address above.
- 7) **Unicode and Character Encoding:** Research centered on the representation and character coding, due to the globalization of software. Unicode became a standard, resulting in the exploration of efficient methods to handle international character sets in coding languages.
- 8) **Big Data and Data Structures:** Efficient data structures and algorithms were being addressed in the researches to handle big data. Substantial study of topics such as balanced trees, hash tables and graph structures helped in optimization of data processing.
- 9) **Parallel and Concurrent Data Types:** In the present world, there has been a center of attention on synchronization for concurrent and parallel programming and data types as compared to multi-core processors and distributed systems. This encompasses lock-free data structures and concurrency control.
- 10) **Type Safety and Security:** Highlighting of type safety, type inference for security, as well as the prevention of programming with the increasing concern about software security, emphasized type safety, type inference for security, and the prevention of common programming helplessness such as buffering overflow and type-related security issues.

3. Working

Data types are very important for the smooth functioning and representation of the data in programming languages. Data types organize data sets given by the user to the interpreter in different-different

values, these values are then further used and categorized in different variables, these values are then utilized according to different functions and operations being applied on them, each value acquire unique storage space based on their data types and information they hold. Data types works differently for different programming languages but they mainly work on the following principles in programming languages:

- A. Type declaration
- B. Data storage
- C. Operations
- D. Type compatibility
- E. Type safety
- F. User-defined data types
- G. Data type conversions

4. Future Scope

Operations performed among several variables in programming languages, was made efficient and easy with the help of data types. Every data type has a specific and defined calculation as well as transformation, which are used with different variables of similar data type, to be executed without any difficulty. To reduce all the runtime errors, occurring during the operation of multiple variables, type system is used.

References

- [1] Horowitz, E. (2012). Fundamentals of programming languages. Springer Science & Business Media.
- [2] Summerfield, M. (2010). Programming in Python 3: a complete introduction to the Python language. Addison-Wesley Professional.
- [3] Sestoft, P. (2017). Programming language concepts. Springer.
- [4] Sharan, K. (2014). Beginning Java 8 Fundamentals: Language Syntax, Arrays, Data Types, Objects, and Regular Expressions. Apress.
- [5] Lee, K. D. (2017). Foundations of programming languages. Springer.
- [6] Loudon, K. C., & Lambert, K. A. (2011). Programming languages: principles and practices. Cengage Learning.
- [7] Pace, L. (2012). Beginning R: An introduction to statistical programming. Apress.
- [8] Ekmekci, B., McAnany, C. E., & Mura, C. (2016). An introduction to programming for bioscientists: a Python-based primer. PLoS computational biology, 12(6), e1004867.
- [9] Guttag, J. V. (2016). Introduction to computation and programming using Python: With application to understanding data. MIT press.
- [10] Liang, Y. D. (2013). Introduction to programming using Python.
- [11] Kochan, S. G. (2015). Programming in C. Pearson education.
- [12] Chapman, S. J. (2016). Essentials of MATLAB programming. Cengage Learning.
- [13] Kraveva, R., Kravev, V., & Kostadinova, D. (2019). A Methodology for the Analysis of Block-Based Programming Languages Appropriate for Children. J. Comput. Sci. Eng., 13(1), 1-10.
- [14] Kunkle, W. M., & Allen, R. B. (2016). The impact of different teaching approaches and languages on student learning of introductory programming concepts. ACM Transactions on Computing Education (TOCE), 16(1), 1-26.
- [15] Donovan, A. A., & Kernighan, B. W. (2015). The Go programming language. Addison-Wesley Professional.
- [16] R. K. Kaushik Anjali and D. Sharma, "Analyzing the Effect of Partial Shading on Performance of Grid Connected Solar PV System", 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-4, 2018.
- [17] R. Kaushik, O. P. Mahela, P. K. Bhatt, B. Khan, S. Padmanaban and F. Blaabjerg, "A Hybrid Algorithm for Recognition of Power Quality Disturbances," in IEEE Access, vol. 8, pp. 229184-229200, 2020.
- [18] R. Kaushik, O. P. Mahela and P. K. Bhatt, "Events Recognition and Power Quality Estimation in Distribution Network in the Presence of Solar PV Generation," 2021 10th IEEE International

Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2021, pp. 305-311

- [19] Sharma, R. and Kumar, G. (2017) "Availability improvement for the successive K-out-of-N machining system using standby with multiple working vacations," International journal of reliability and safety, 11(3/4), p. 256. doi: 10.1504/ijrs.2017.089710.
- [20] Gireesh, K., Manju, K. and Preeti (2016) "Maintenance policies for improving the availability of a software-hardware system," in 2016 11th International Conference on Reliability, Maintainability and Safety (ICRMS). IEEE.