3

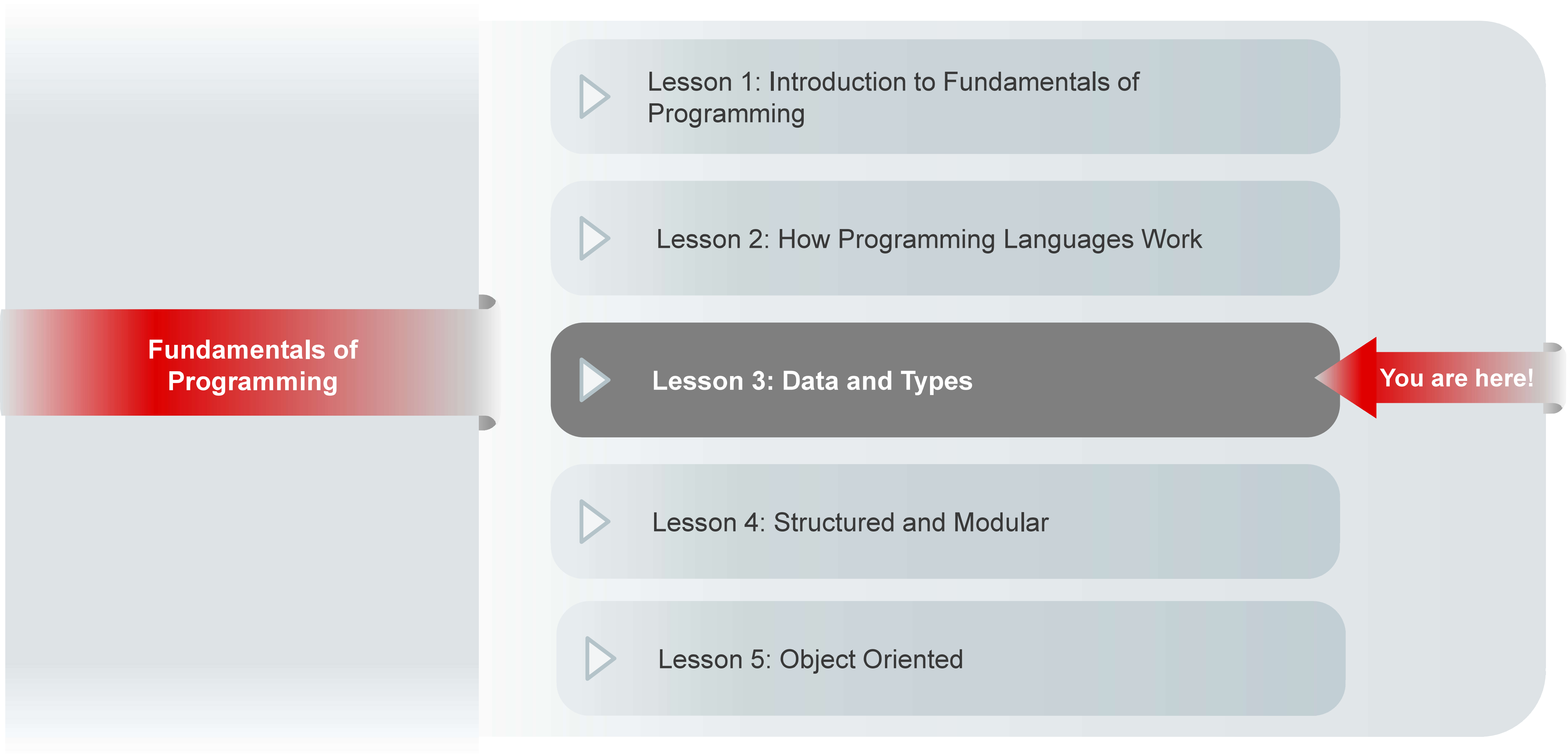
Data and Types

Objectives

After completing this lesson, you should be able to:

- Understand key terms and definitions.
- Understand basic data types and how operators manipulate data.
- Given example pseudocode, flowcharts, and source code, create a program that uses appropriate data types and operators to solve a given problem.

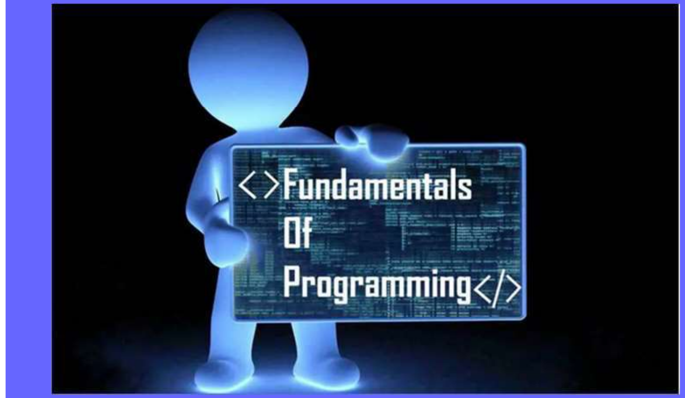
Course Roadmap



Fundamentals of Programming

- Lesson 1: Introduction to Fundamentals of Programming
- Lesson 2: How Programming Languages Work
- **Lesson 3: Data and Types**
- Lesson 4: Structured and Modular
- Lesson 5: Object Oriented

You are here!

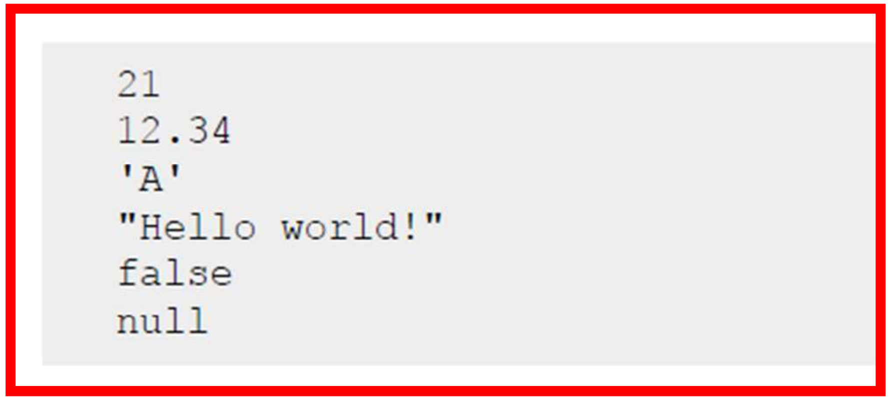


Constants and Variables

- A **constant** is a value that cannot be altered by the program during normal execution, i.e., the value is constant. When associated with an identifier, a constant is said to be “named,” although the terms “constant” and “named constant” are often used interchangeably.
- This is contrasted with a **variable**, which is an identifier with a value that can be changed during normal execution, i.e., the value is variable.

- A **constant** is a data item whose value cannot change during the program’s execution. Thus, as its name implies – the value is constant.
- A **variable** is a data item whose value can change during the program’s execution. Thus, as its name implies – the value can vary.
- Constants are used in two ways.
  - Literal constant
  - Defined constant
- Constant
  - A data item whose value cannot change during the program’s execution.
- variable
  - A data item whose value can change during the program’s execution.

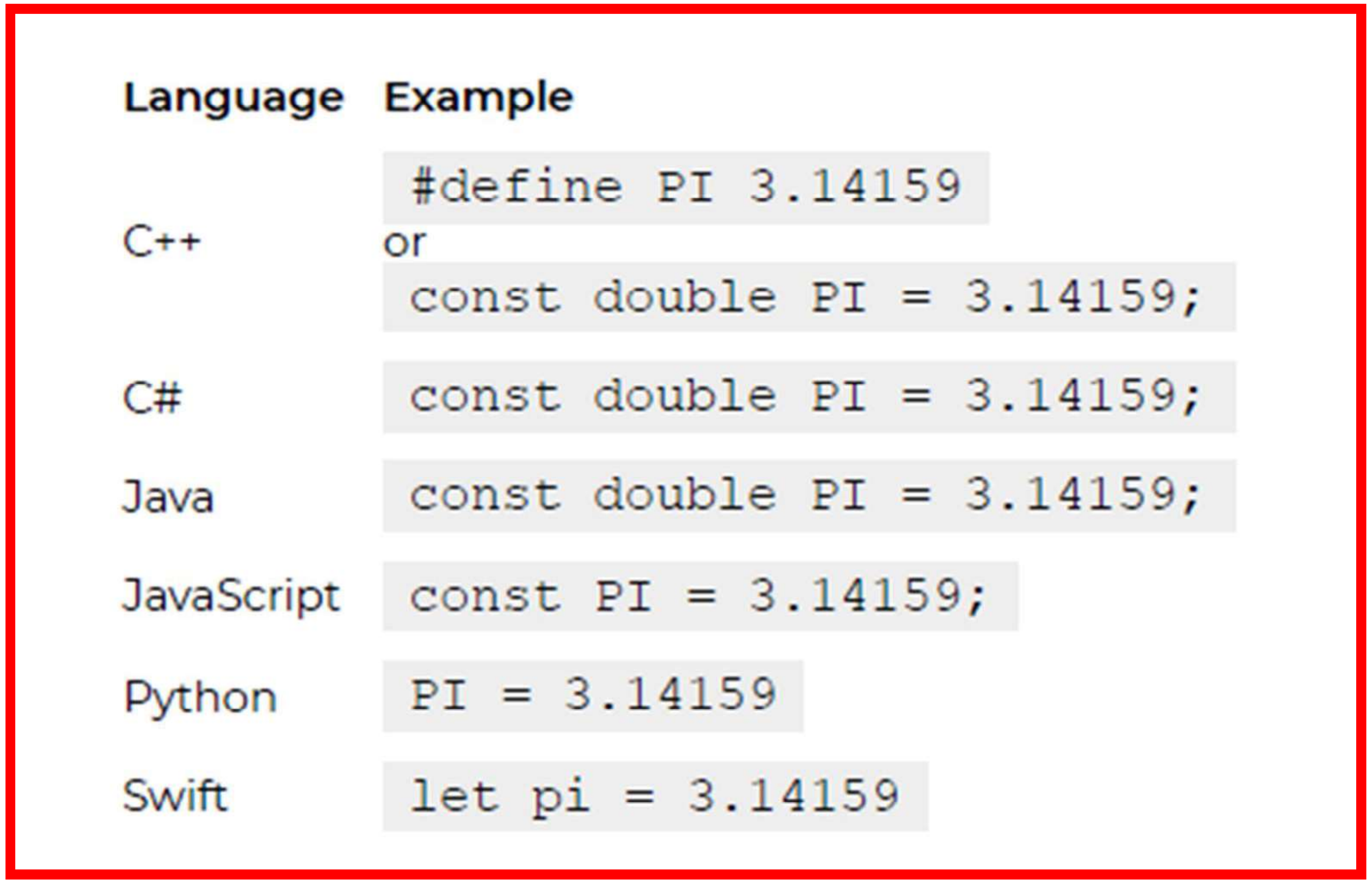
- A literal constant is a **value** you type into your program wherever it is needed.



```
21
12.34
'A'
"Hello world!"
false
null
```

- In addition to literal constants, most textbooks refer to symbolic constants or named constants as a constant represented by a name. Many programming languages use ALL CAPS to define named constants.

Examples



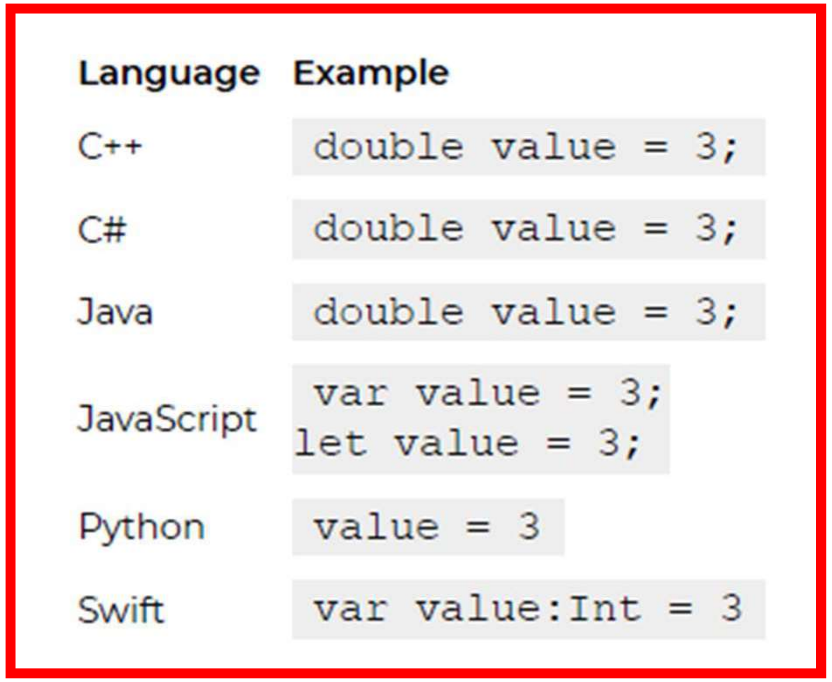
Language	Example

C++	`#define PI 3.14159`
	or
	`const double PI = 3.14159;`

C#	`const double PI = 3.14159;`
Java	`const double PI = 3.14159;`
JavaScript	`const PI = 3.14159;`
Python	`PI = 3.14159`
Swift	`let pi = 3.14159`

Defining Constants and Variables

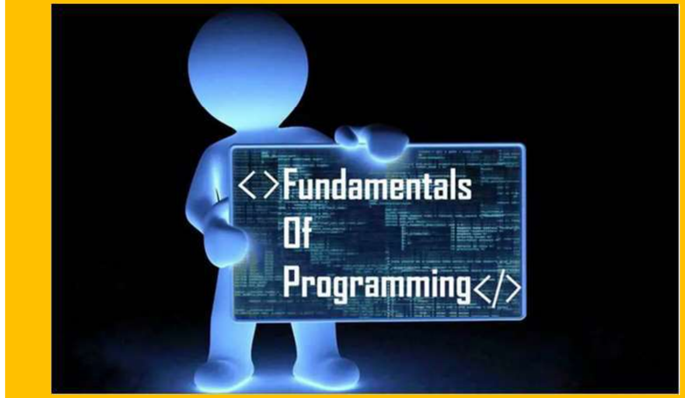
- Named constants must be assigned a value when they are defined. Variables do not have to be assigned initial values. Variables once defined may be assigned a value within the instructions of the program.



Language	Example
C++	`double value = 3;`
C#	`double value = 3;`
Java	`double value = 3;`

JavaScript	`var value = 3;`
	`let value = 3;`

Python	`value = 3`
Swift	`var value:Int = 3`

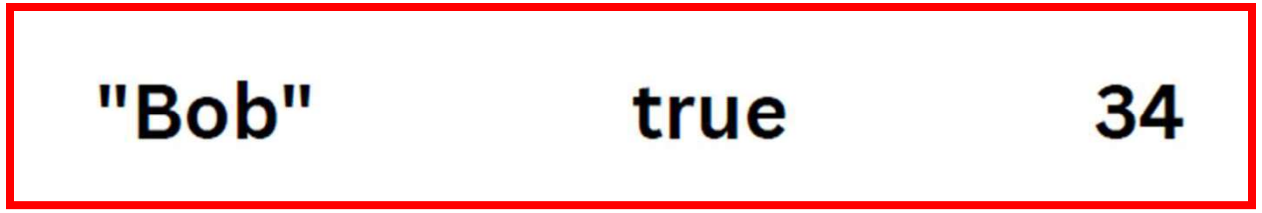


Identifiers

What are variables?

- What is a variable? And why is it the MOST important?

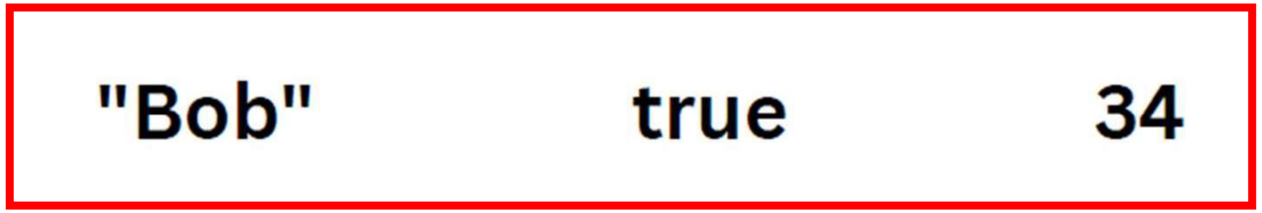
- Variables
  - Something that can store information
  - Can be referenced and manipulated



"Bob" true 34

What is a Variable: Variable Basics

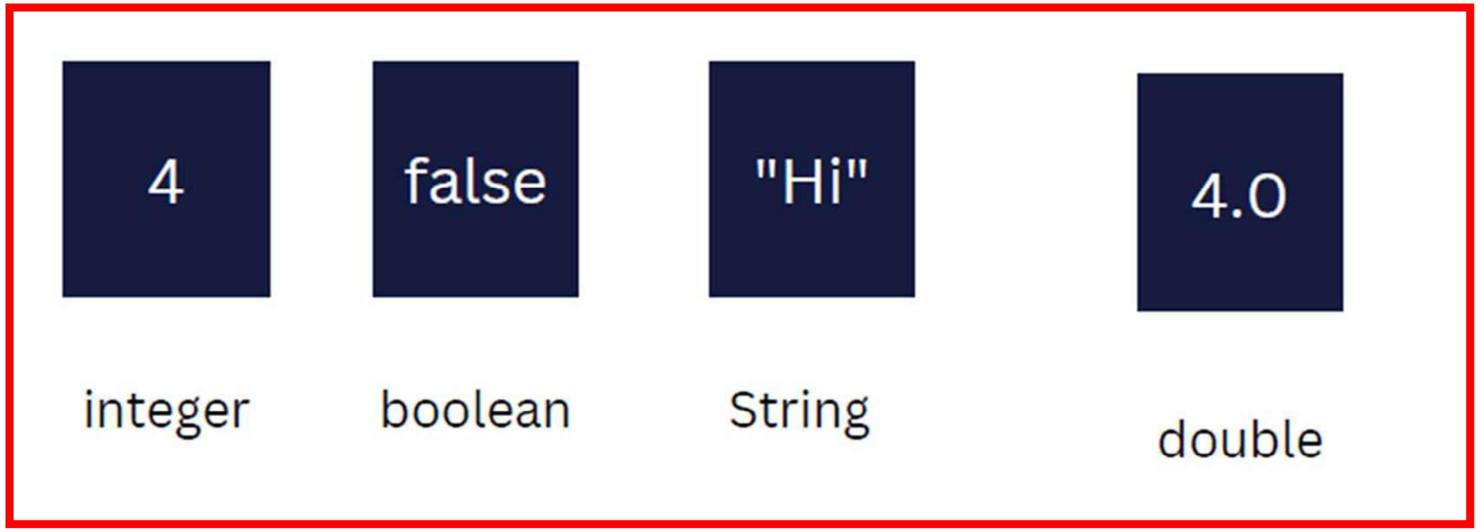
- Each variable has a type, a name, and a piece of information stored inside.



"Bob" true 34

What is a Variable: Types of Variables

- Many type of variables, but we will be focussing on primitive type variables



4

integer

false

boolean

"Hi"

String

4.0

double

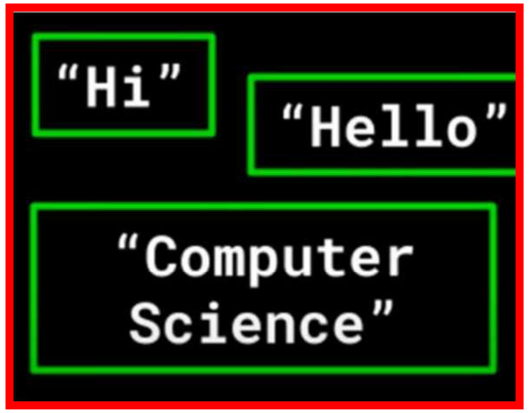
- Integer's
  - A variable that can store integer values
  - Can't and will not hold any decimal values

What is a Variable: Floats and Double

- Both are types of floating point data types can store numbers with decimal places
  - 4.23
- Float Variables
  - Can store upto 32 bits of information
- Double Variables
  - Can store up to 64 bits of information

What is a Variable: Strings

- What we talked about before, except stored somewhere in a variable
- Useful for displaying text and storing input information
- Also useful for outputting information in a readable format for the user



What is a Variable: Chars

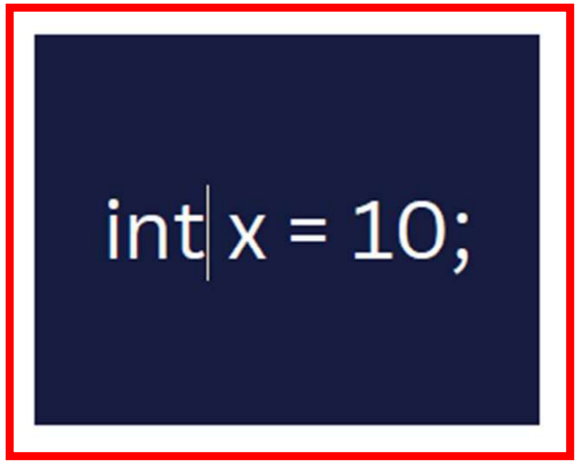
- Char -> Character
- Each hold one character
- Useful when a programmer wants to read one character at a time

What is a Variable: Why are Variables so useful -

- Often times you are going to want to keep track of things such as user's name or score
- By creating a variable ***user_name*** or score, you can store this useful information
- Other important uses of variable
- Taking input from the user
- Manipulating variables is necessary for any tasks in programming

How do we manipulate variables? What happens when we define a variable?

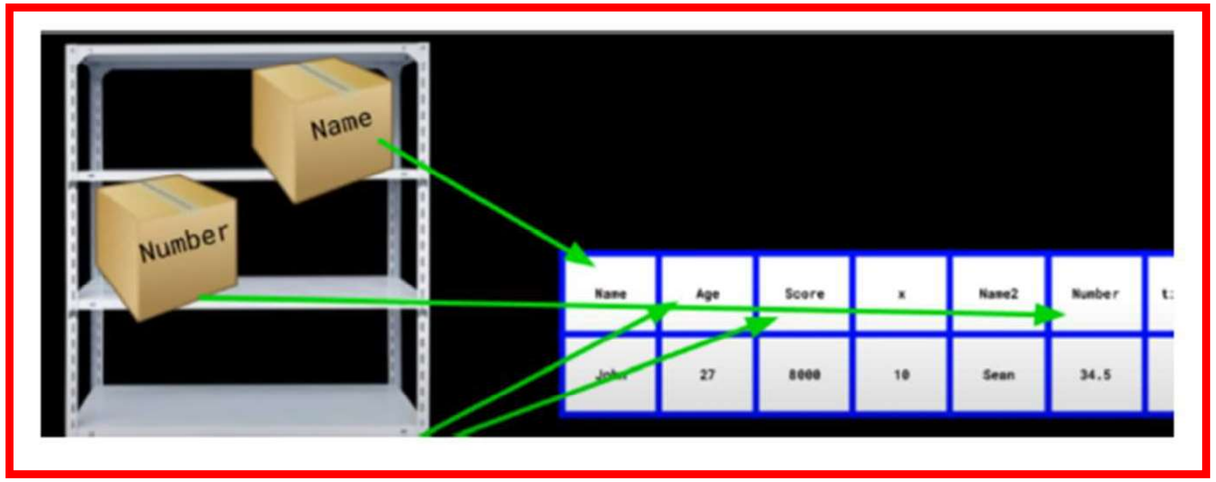
- Create a little space in memory that stores your variable name and its contents



```
int x = 10;
```

How do we manipulate variables? Warehouse example?

- Consider space in a computer as a space in a warehouse and a variable is occupying space in the ware house

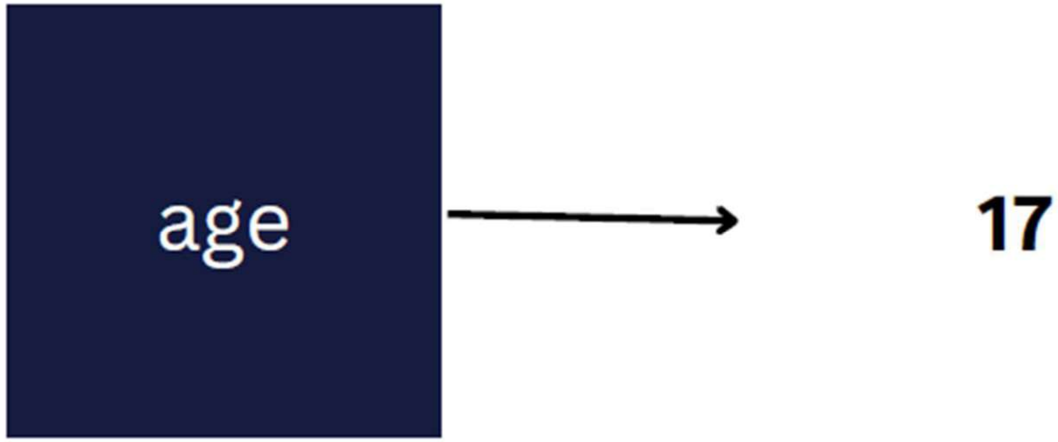


Name	Age	Score	x	Name2	Number	t:
John	27	8000	10	Sean	34.5	

How do we manipulate variables? Blank variable example?

- Reasons you would do this
  - Because you want to store information in the variable down the road
  - Because you are going to use it to store information given to you by the user
- Trying to reference a blank variable will result in a NullPointerException

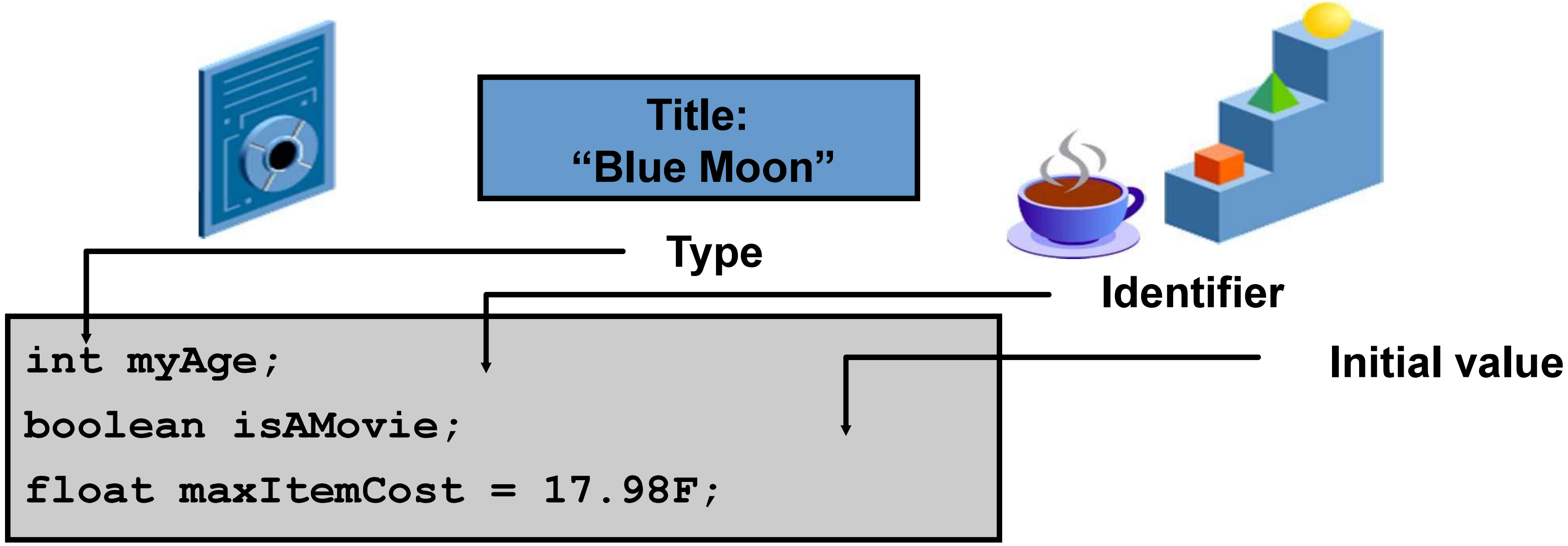
How do we manipulate variables? Changing Variable Example



```
age = 18;
```

Variables

- A variable is a basic unit of storage.
- Variables must be explicitly declared.
- Each variable has a type, an identifier, and a scope.



```
int myAge;
boolean isAMovie;
float maxItemCost = 17.98F;
```

How do we manipulate variables? The Fate of variable

deleted in memory

GONE

How do we manipulate variables? Naming convention

- There is one big rule when naming variables
  - They must be one continuous string
- Most programmers name variables according to camel case
  - A CamelCaseVariable
  - Another CamelCaseVariable
    - userName
    - firstName etc.

- Within programming a variety of items are given descriptive names to make the code more meaningful to us as humans.
- These names are called “Identifier Names”. Constants, variables, type definitions, functions, etc. when declared or defined are identified by a name.

1. the language’s technical limitations
2. good programming practices
3. common industry standards for the language

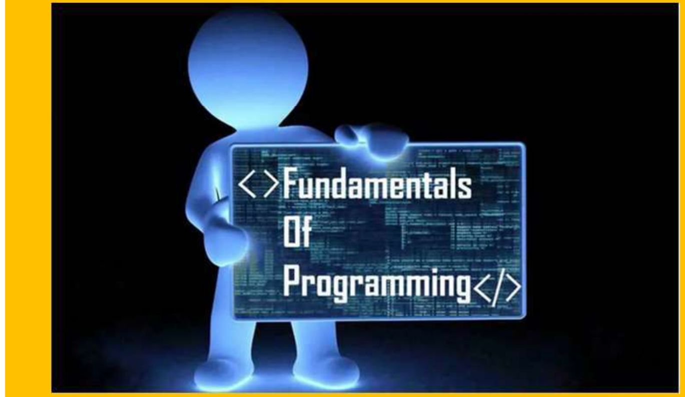
- Use only allowable characters (in many languages the first character must be alphabetic or underscore, can continue with alphanumeric or underscore)
- Can’t use reserved words
- Length limit

- These attributes vary from one programming language to another. The allowable characters and reserved words will be different.
- The length limit refers to how many characters are allowed in an identifier name and often is compiler dependent and may vary from compiler to compiler for the same language.
- However, all programming languages have some form of the technical rules like
- **Good Programming Techniques**
  1. Meaningful
  2. Be case consistent

Industry Rules

- Almost all programming languages and most coding shops have a standard code formatting style guide programmers are expected to follow. ‘
- Among these are three common identifier casing standards:
  - **camelCase** – each word is capitalized except the first word, with no intervening spaces
  - **PascalCase** – each word is capitalized including the first word, with no intervening spaces
  - **snake_case** – each word is lowercase with underscores separating words

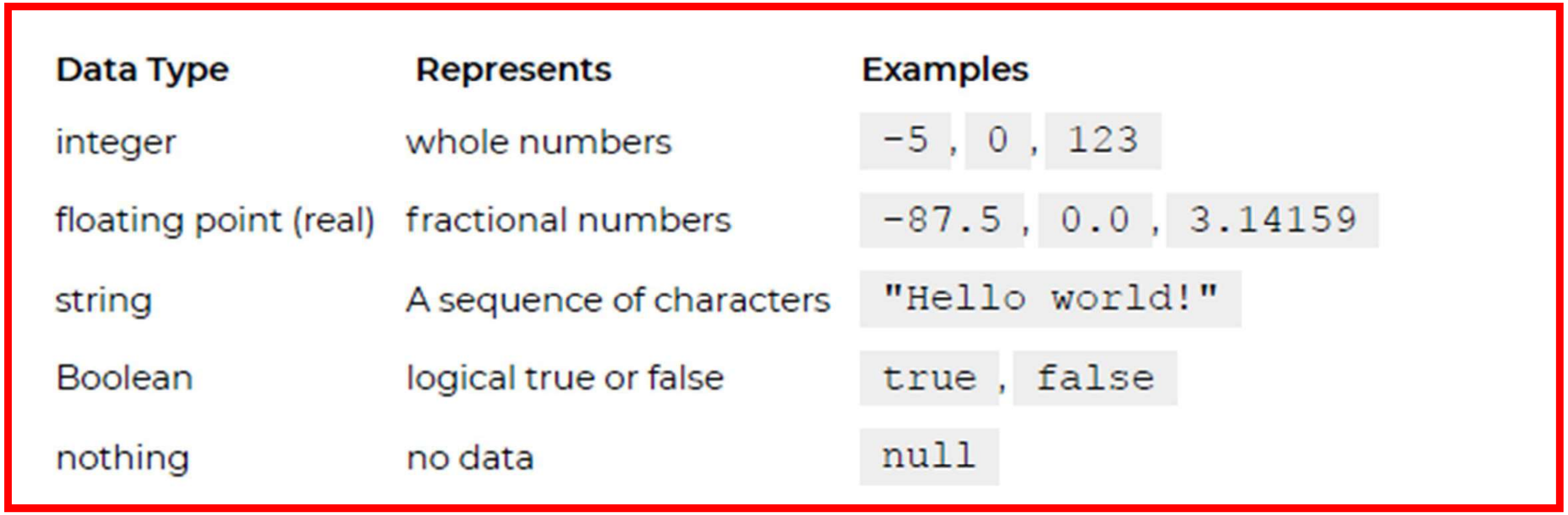
- C++, Java, and JavaScript typically use camelCase, with PascalCase reserved for libraries and classes. C# uses primarily PascalCase with camelCase parameters. Python uses snake_case for most identifiers. In addition, the following rules apply:
  - Do not start with an underscore (used for technical programming)
  - CONSTANTS IN ALL UPPER CASE (often UPPER_SNAKE_CASE).



Data Types

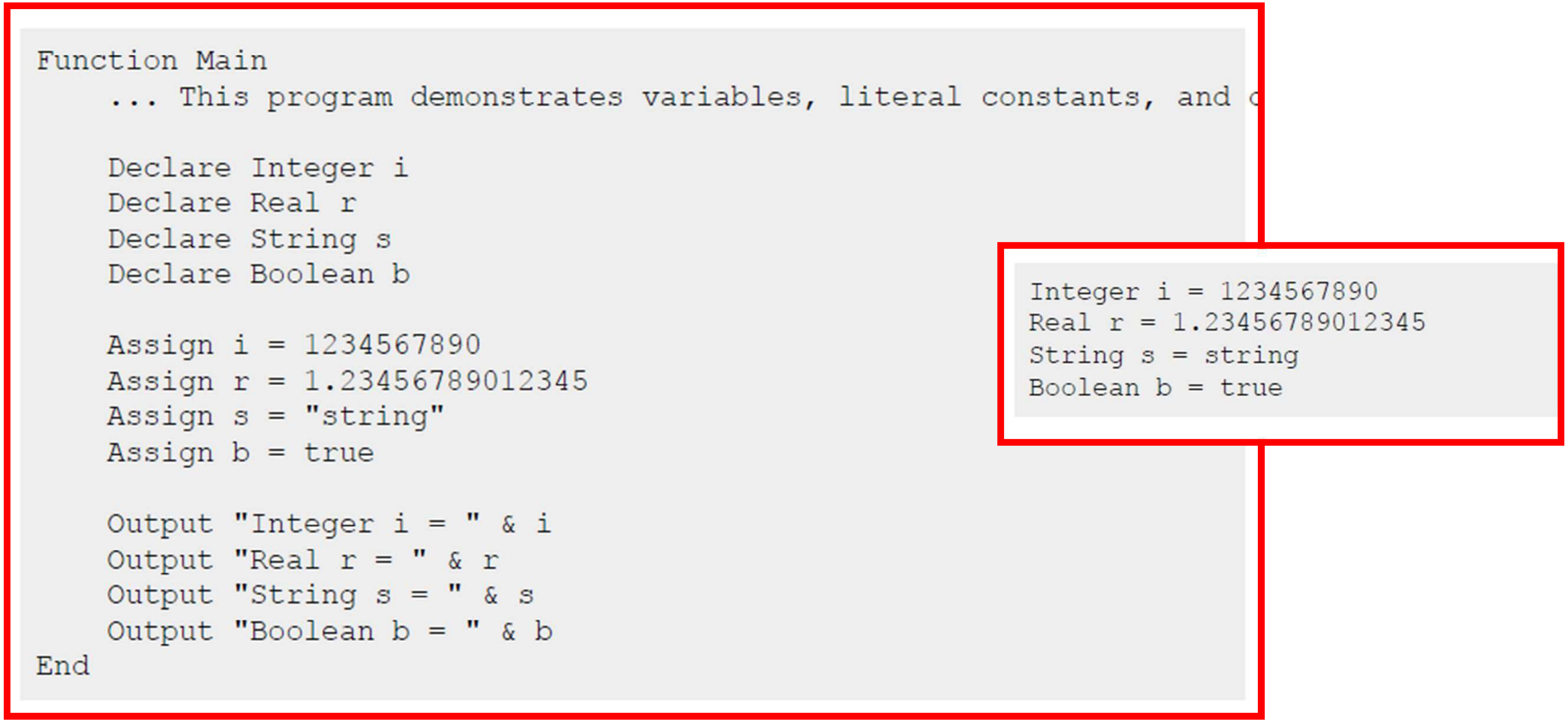
- A **data type** is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.
- Most programming languages support various types of data, including integer, real, character or string, and Boolean.

  - **your name** – a string of characters
  - **your age** – usually an integer
  - **the amount of money in your pocket** – usually a value measured in dollars and cents



Data Type	Represents	Examples
integer	whole numbers	`-5` , `0` , `123`
floating point (real)	fractional numbers	`-87.5` , `0.0` , `3.14159`
string	A sequence of characters	`"Hello world!"`
Boolean	logical true or false	`true` , `false`
nothing	no data	`null`

Pseudocode



```
Function Main
    ... This program demonstrates variables, literal constants, and c

    Declare Integer i
    Declare Real r
    Declare String s
    Declare Boolean b

    Assign i = 1234567890
    Assign r = 1.23456789012345
    Assign s = "string"
    Assign b = true

    Output "Integer i = " & i
    Output "Real r = " & r
    Output "String s = " & s
    Output "Boolean b = " & b
End
```

```
Integer i = 1234567890
Real r = 1.23456789012345
String s = string
Boolean b = true
```

Summary

In this lesson, you should have learned that:

- Definition of Program, Computer Programming, and Computer Programmer.
- Generations of Programming Language
- Types of Programming Language