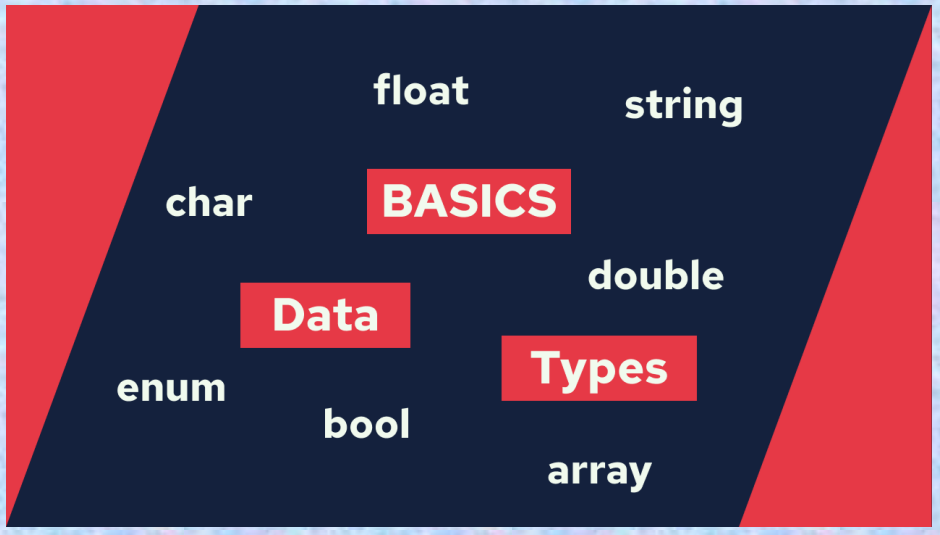


Variables Methods and Operators

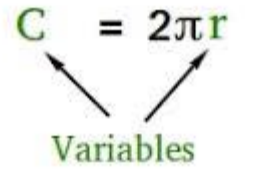
Ashutosh Jha,
Technical Officer,
C-DAC Patna

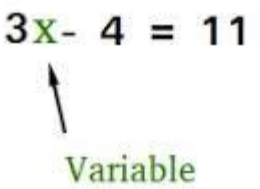
Agenda

- Introductions
- Variables
- Data Types
- Scope
- Methods
- Operators
- Writing a Program

Variables, Methods, and Operators

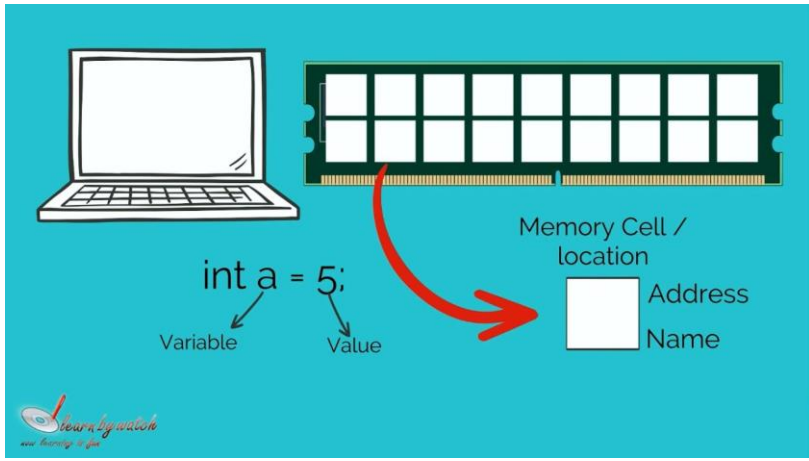
- Variables: Variables are symbolic names that represent values or data stored in memory.
- Methods: Methods are blocks of code that perform a specific task or function.
- Operators: Operators are symbols that perform operations on variables or values.





Variables

- A variable is a named container, held by the computer in a memory location.
- By naming a variable, you can access it easily throughout the program.
- A variable can hold one value (piece of data) at a time.
- This value can change throughout the execution of a program. When a new value is placed in (or assigned to) a variable, it replaces the last value.
- Each variable used in a program is given a unique identifier (name).

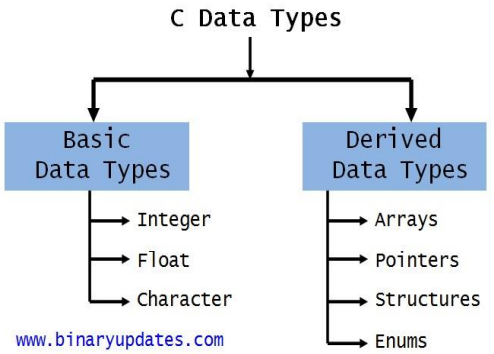


Declaring and Initializing variables

- When you create a variable, the process is known as **declaring** a variable.
- You specify its **name** and the **type** of data it can contain
- A **memory location** will be reserved for the data and you will be able to access the location by specifying the name of the variable
- Some languages such as Python do not require you to declare variables before you use them.
- When you set the initial value for a variable, you are initialising that variable.
- A variable must be initialised before it is referenced (used).
- A constant is used in programming when a value needs to be available to the program but it will not change during the execution of the program.

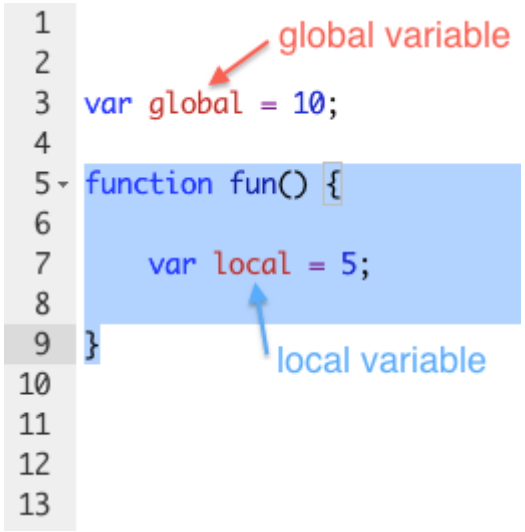
Data Types

- A data type is a formal classification of the type of data being stored or manipulated within a program.
- There are four ‘primitive’ or basic data types, from which all others can be created.
  - Integer - Whole number. They can positive or negative but they must be whole. (5, -7, 111)
  - Real - Real is also referred to as float. Real numbers include numbers with decimal places (1.1, -1.9, 382.4)
  - Boolean - based on logic and can have one of only two values, True or False
  - Char - Char refers to a single character. The character can be any letter, digit, or symbol. (c, A, X, £)
- *String - collection of chars



Variable Scope: Global vs Local

- Global variables are accessible from anywhere in the program and are declared outside of any function.
- Local variables are accessible only within the function they are declared in and are declared inside a function.
- Global variables can be accessed and modified by any function in the program.
- Local variables can only be accessed and modified by the function they are declared in.
- It's important to use the appropriate scope for variables to avoid naming conflicts and to make your code more maintainable.



Best practices for naming variables

- Use descriptive names that reflect the purpose of the variable.
- Use camelCase naming convention, starting with a lowercase letter and capitalizing each new word.
- Avoid using single-letter names or abbreviations unless they are well-known and widely used.
- Don't use reserved keywords or special characters in variable names.
- Be consistent in your naming conventions throughout your program.

Methods

- Methods are like recipes for solving specific problems in programming.
- They take input, perform operations, and return output.
- Built-in methods, such as toUpperCase() and length(), are part of the language and can be called on values of specific types.
- User-defined methods are created by the programmer and can be called to perform specific tasks in a program.
- Methods can have parameters, which allow the programmer to pass data into the method for processing.
- Methods can also have return values, which allow the method to pass data back to the calling code.

Calling methods

- To call a method, you need to provide its name, followed by parentheses.
- Inside the parentheses, you can provide any required parameters, separated by commas.
- If the method has a return value, you can assign it to a variable or use it in other expressions.
- When calling a method on an object, you use the dot notation, which separates the object name and the method name with a period.
- If the method is a static method, you call it on the class name instead of an object instance.

Built-in methods

- Programming languages provide built-in methods as part of their standard library to help programmers accomplish common tasks.
- Built-in methods are part of the language and can be called on values of specific types.
- Some common built-in methods in programming include toUpperCase(), length(), parseInt(), and toString().
- The main() function is a special method used in many programming languages to mark the entry point of a program.
- The main() function is called automatically when the program starts, and it usually contains the code that initializes the program and starts its execution.

Custom methods

- Custom methods are created by programmers to solve specific problems in a program.
- Custom methods can take input parameters, perform operations on them, and return output results.
- Custom methods help us write modular, reusable, and efficient code that is easier to understand, test, and maintain.
- Custom methods also allow us to hide implementation details and focus on the high-level logic of a program.

```
function calculateTax(price, rate) {
 tax = price * rate;
return tax;
}
```

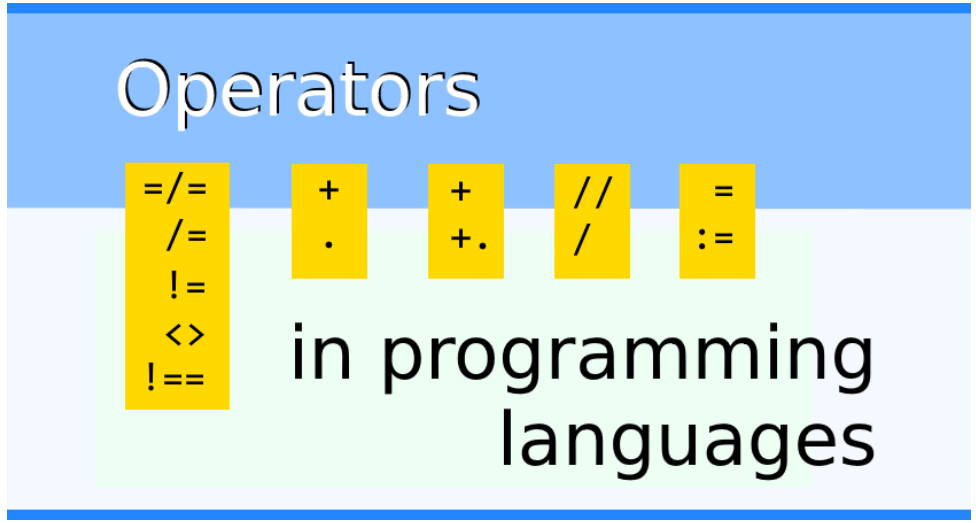
Parameters and Return values

- Parameters are input values passed to a method, and return values are the output results returned by the method.
- Parameters and return values help us create flexible and reusable methods that can handle different inputs and produce different outputs.
- Parameters and return values are essential tools in programming that help us solve complex problems and write better code.

```
public double calculateArea(double radius) {
double area = Math.PI * radius * radius;
return area;
}
```

Operators

- Operators are symbols that perform operations on variables and values in a program.
- Operators are used to perform arithmetic, comparison, logical, assignment, and other operations.
- Understanding and using operators is essential in programming to perform mathematical calculations, compare values, and manipulate data.



Arithmetic operators

- **+** , Addition
- **–**, Subtraction
- ***,** Multiplication
- **/,** Division
- **^,** To the power
- **%,** Modulo (Remainder)
- **++,** (Increment)
- **––** , (Decrement)

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Raised to a power

Comparison operators

- == : Equals
- != : Not Equals
- > : Greater Than
- <: Less Than
- >= : Greater Than or Equal To
- <= : Less Than or Equal To

Operators	Meaning	Example
>	Greater than	x > y
<	Less than	x < y
==	Equal to	x == y
!=	Not equal to	x != y
>=	Greater than equal to	x >= y
<=	Less than equal to	x <= y

Logical operators

- &&: AND
- ||: OR
- ! : NOT

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

Assignment operators

- = (Simple Assignment)
- += (Add and Assign)
- -= (Subtract and Assign)
- *= (Multiply and Assign)
- /= (Divide and Assign)
- %= (Modulo and Assign)

Assignment Operators

Operator	Example	Equivalent Expression (m=15)	Result
+=	m +=10	m = m+10	25
-=	m -=10	m = m-10	5
*=	m *=10	m = m*10	150
/=	m /=	m = m/10	1
%=	m %=10	m = m%10	5

Writing your own program

- Write a program to determine whether a number entered by the user is even or odd. The program should ask the user to enter a number, store this value in a variable, and then check if the number is even or odd using the modulo operator %. If the number is even, the program should print "The number is even" to the console, and if it is odd, the program should print "The number is odd" to the console.

Recap

- In today's session, we learned about Variables, Methods, and Operators in programming
- Overall, today's session gave us a deeper understanding of how programming works and the essential role Variables, Methods, and Operators play in building robust programs. With practice and experimentation, we can continue to improve our programming skills and tackle more complex challenges.

Coming UP...

- In our next session, we will discuss about Conditional and Looping Statements. We'll also dive into recursion.
- Be sure to Complete the assignments that are assigned to you, as we will be discussing that first thing in the morning.

Questions

Thank You!!!

https://www.linkedin.com/company/c-dac-patna

https://twitter.com/CDAC_Patna

https://www.facebook.com/PatnaCDAC/

https://www.youtube.com/c/CDACPatna