

Programming

Fundamentals

What is programming?

Programming is the process of converting ideas into instructions that a computer can understand and execute.

These instructions are **specific** and **sequential**.

- The instructions we provide to computers need to be specific so that we can get our desired result.
- Computers are very literal. They try to execute our commands exactly.
- When we give them bad instructions, we might introduce bugs or even make the computer crash.

Bug

A bug is when something unexpected happens, whereas a crash is when your programming stops early or freezes.

Sequential

Order matters in programming. Because programming is just as much about finding errors and preventing crashes as it is about writing instructions or code that the computer understands.

What is a programming language?

When we want to provide instructions to a computer, we use what's called programming language.

Programming languages have **unique syntax rules**.

C++ Syntax

```
#include  
int main()  
{  
    std::cout << "Hello, world!";  
    return 0;  
}
```

Javascript syntax - `document.write("Hello, world!");`

Python syntax - `print "Hello world!"`

Programming is how we communicate with the digital world.

Programming Basics

Why so many languages?

Each language comes with its own set of strengths and weaknesses.

Some are for programming small devices with limited memory, whereas others were made to handle complex mathematical computations.

Language has to be broken down into the only one that computers understand, machine language.

Machine language

Machine language is extremely complex for us to write directly because it's mostly just a series of numbers.

High-level languages

They are closer to human languages.

Source code

It's the instructions we have for the computer, and it's written in plain text.

- Word processing applications are not suitable for writing code because by default they insert bits of information in files that prevent them from being plain text. Instead we can use a text editor.

- Each programming language has its own file extension.

Javascript → .js perl → .pl kotlin → .kt

Running code

Python is an interpretive language that means we have to interpret its file into machine code.

There are three ways to translate source code into machine code —

1 → Compile it 2 → Interpret it 3 → A combination of both

Compiler

Compilers take our high-level programming language and turn it into an executable that contains low-level machine code. In this way users can run their code on their machine without ever needing your original source code.

Interpreter

Computer interpreters process our source code each time it's run, line by line and it's up to the other users to have the needed interpreter available on their machine.

Compiled languages: C, C++ and objective C

Interpreted languages: php and javascript

Combination: Java, C#, Python

IDE

Integrated Development Environment

An IDE is an application that provides the special tools needed to write, debug and compile code.

Xcode

Xcode is used to develop applications for Apple products.

VS code

It is more lightweight editor and initially designed for scripting languages like JS, Typescript. It supports many languages through powerful extensions.

One of its unique features is IntelliSense, allows to get code suggestions while typing.

Android Studio

for Android apps

Statements, Troubleshooting issues, Variables, data type

Error Categories

- Syntax** - language rules broken
- Runtime** - unable to execute (happens when program is actually running)
- Semantic** - unexpected output

Statements

Statements are the building blocks of any program and are the individual actions that we want the program to take.

Each statement can be made up of **Keywords**, **Expressions**, **Operators**

Operators

Operators are symbols that tell the computer to perform an action with some input.

Arithmetic operators

+ (Addition) **-** (Subtraction) ***** (Multiplication) **/** (Division)

These are known as **arithmetic operators**. - They take numbers for input and perform an arithmetic operation.

$3 + 2 =$ 3, 2 - operands/inputs
 + - operator

Expressions

In programming, the combination of operators and operands that break down to a single value are called expressions.

In programming, the default order of operations is the same as in mathematics.

Order of operations

P - Parentheses first

E - Exponents

MD - Multiplication and division (left to right)

AS - Addition and subtraction (left to right)

Computer programs are made up of statements.

Variable

A variable is simply a container for a value. When we run our programs, the computer gives us space in its memory where we can put data that we want to use as a reference for later. This data is called variable.

In python, we declare variables by giving them a name and then setting their value. This is called **assigning value** and this assignment is done with the **assignment operator**.

In python $\rightarrow (=)$ sign

Data Types

We want the variables to represent different data types.

A data type allows us to put our variable in a particular category so that the computer knows how much space to give us in its memory.

Integer - is a whole number, no decimal places.

String - letters and numbers inside a double quotes called string because it's made up of a string of characters.

- We can check any type of any variable by **type()** function in python.

Variables in other languages, Conditional Code

Some languages require that you define your variables and their types before you can use them — **Java**, **C#** and **C++**

Java `String cookie = "Sugar";` — first declare its type is string
`System.out.println(cookie);`

Python we don't need to declare a variable's type before using it.
`name = "Rahul"` Python interpreter is able to figure out the type based on the value that we provided.

Variable Rules

- 1 Variable names should only contain letters, numbers and underscores. The name should not start with a number.
- 2 Spaces are not allowed in variable names.
- 3 Names are case-sensitive.
- 4 Variables can't be keywords.

Strings, numbers, comments

Whenever we have to deal with text, we use strings.

— letters, numbers, symbols, spaces
" " and ' ' depends on the content of the string.

Comments are notes to us for future or for other to understand the code.

Making Decisions in code

Conditional or Boolean expression

Any expression that breaks down to either **true or false**.

→ is 4 equals to $2 * 2 \Rightarrow \text{True}$ → is 4 greater than 5 $\Rightarrow \text{False}$

Relational operators Relational operators work with two operands, and return a value **True or False** based on their relation to each.

Equality operators → `x == y` $\Rightarrow \text{False/True}$

`==` Equal to `!=` not equal to `<` less than `>` greater than `>=` greater than or equal to
`<=` less than or equal to

If statement

The most common way for a computer to make decisions is by means of an if statement.

If condition
do work

end if

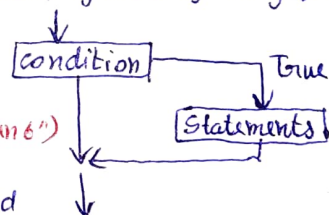
Python

`if 5 < 6:`

`print("Yes, 5 is less than 6")`

`print()`

we have a group name of statements that are grouped together this way in programming. it's called block.



If-else statement

if condition:

do work

else:

do something else

Keywords are reserved words that mean something special to Python interpreter.
— break or try.

Import keyword
`print(keyword, kwlist).`

In JS there's no difference between int and float
Float — Any number with a decimal point.

Conditions in other languages, Functions

Python

```
if 5 > 6:
    print("no")
else:
    print("yes")
```

Java

```
if (5 > 6) {
    System.out.println("no");
} else {
    System.out.println("yes");
}
```

After if keyword, there's a set of parentheses. put condition inside parentheses. instead of a colon after condition - a pair of curly brackets

Ruby

Ruby is known for its ease of use and flexibility

```
if 5 > 6
    puts "no"
else
    puts "yes"
end
```

Functions

A function is a block of code packaged together with a name.

We used some built in functions - `print()` `input()`

Functions help us to avoid writing the same lines of code again and again.

We have to give functions an unique name just like variables.

```
def hello();
```

Creating function

For function its group of statement is a **body**.

body (statements)

calling → `hello()`

Parameters

The variable inside a function has a special name, it's called parameter

Parameters allow function to ~~choose~~ change their behaviour based on some input.

For the variables we use in our function definition, we have arguments as the name we use for the values that we give to our functions.

arguments

We give our functions arguments by placing a value between the open and close parenthesis in the function call.

```
def add(x, y)
```

parameters

```
add(1, 2)
```

arguments

Return

In Python, when creating a function using the `def` statement, we can specify what the return value should be with a `return` statement.

`return()`

No parameters and no return value

```
def hello();
```

```
    print("Hello")
```

have parameters but don't return value

```
def check(g):
```

```
    if g == "A":
```

```
        print("First")
```

Four categories

No parameters but do return value

```
def food():
```

```
    f = input("Enter")
```

```
    return f
```

have both parameters and return the value

```
def (total, out):
```

```
    if (total >= out):
```

```
        total = total - out
```

```
    return total
```

Functions across languages, Exploring languages

Java

```
void hello() {
```

```
    System.out.println("Hello friends");
}
```

void means function does not return a value.

Java requires that we always specify what the return type is of a function.

Kotlin

```
fun checkGrade(grade: String) {
```

```
    if (grade == "A")
```

```
        println("First")
```

```
}
```

We have to specify what our type of parameter is when we define the function.

Ruby

```
puts "Paycheck calculator"
```

```
def calculate_chuck (hourly_rate)
```

```
    hourly_rate * 40
```

```
end
```

```
puts "You made $ #{calculate_chuck(32)} this week!"
```

C++

```
#include <iostream>
```

```
using namespace std;
```

```
int wm (int cb, int am) {
```

```
    if (cb >= am) {
```

```
        cb = cb - am;
```

```
    }
```

```
    return cb;
```

```
}
```

```
int main() {
```

```
    int balance;
```

```
    balance = wm (100, 50);
```

```
    cout << "Balance is: " << balance << endl;
```

```
    return 0;
```

```
}
```

JavaScript

```
function isEven(n) {
```

```
    if (n % 2 == 0)
```

```
        return "Yes";
```

```
    else
```

```
        return "No";
```

```
}
```

```
console.log (isEven(2));
```

C

C was developed in 1970. used for programming operating systems, games, smart devices, compilers for other languages.

C low level language

Java Python C++

Library

A grouping of variables and functions that someone else has written and verified.

Java

It is a cross platform programming language — write — compile — run on many devices. Desktop apps, ~~Android~~ Android apps. C based language, Everything in java is in what's called a class.

C++

C based language Garbage collection cleans up unused variables for us.

For games and game engines.

Swift and

ObjectiveC → For ios development

Ruby and Javascript

For web Apps