

CPE1009/FOE1001

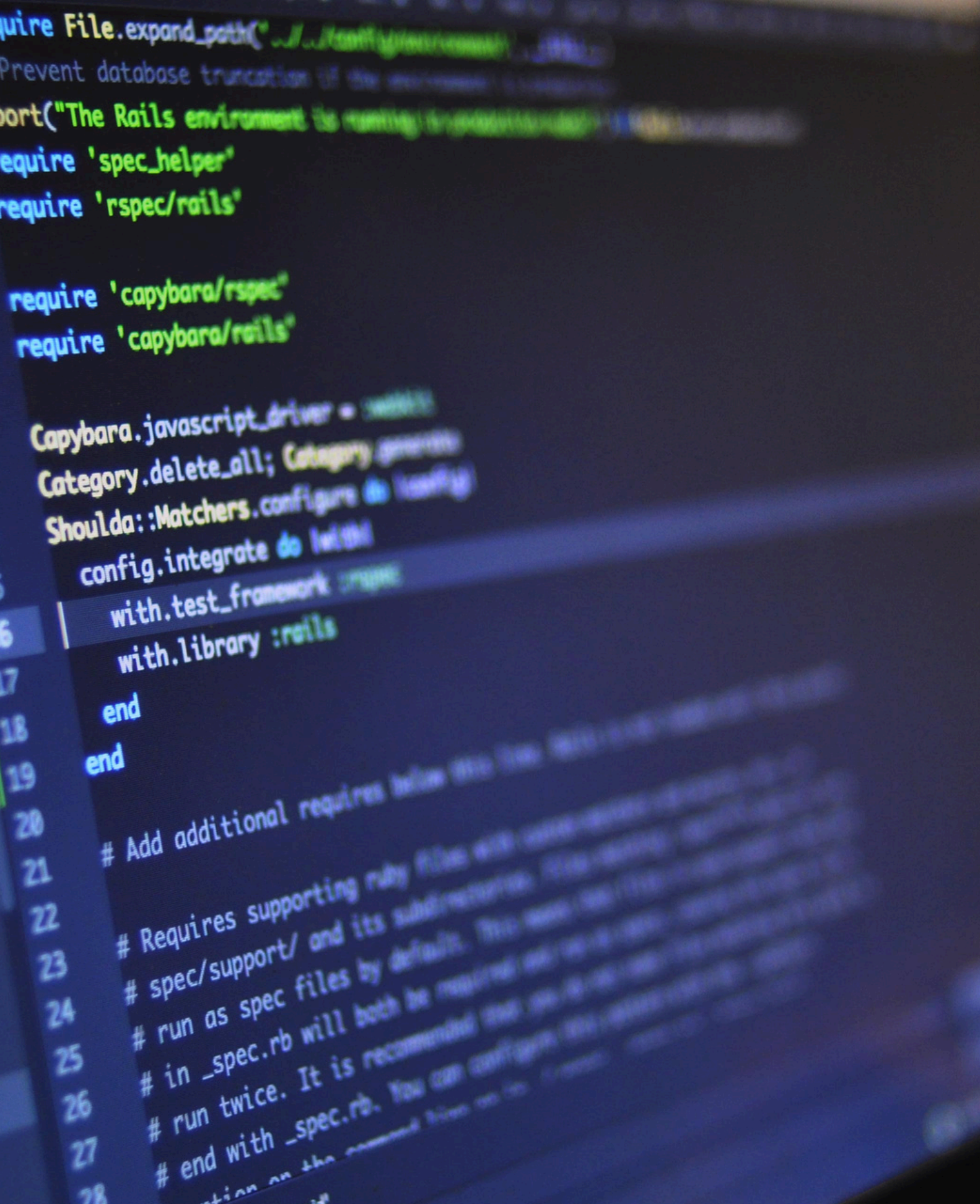
Computer Programming

Dr. Pongrapee Kaewsaiha

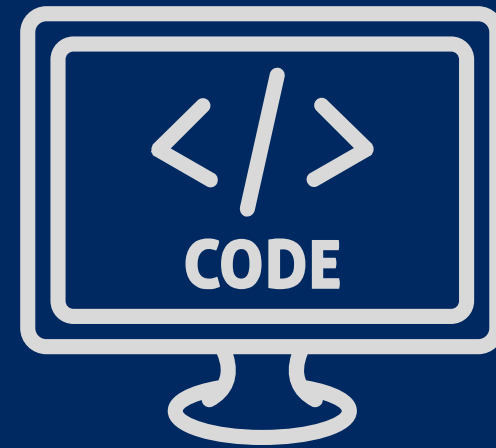


Faculty of
Engineering and
Industrial Technology
Suan Sunandha Rajabhat University





Overview of Programming Languages



1. Why programming?

Programming is the backbone of the digital age.

It allows humans to communicate with computers and command them to perform a variety of tasks.

- Calculate and solve math problems
- Store and find data
- Send and receive files over the internet
- Etc.



```
$result !== false) {  
    $distArray = array();  
    $row = mysqli_fetch_assoc($result);  
    $correctAnswer = $row['Correct'];  
    $distArray['A'] = $row['Anum'];  
    $distArray['B'] = $row['Bnum'];  
    $distArray['C'] = $row['Cnum'];  
    $distArray['D'] = $row['Dnum'];  
    $distArray['Correct'] = $correctAnswer;  
    $distArray['Answer'] = rtrim($row[$correctAnswer], ".");  
    $distArray['Query'] = "SELECT * FROM TechTerms WHERE Date='$date'";  
    return $distArray;  
}  
else {  
    $distArray['Error'] = 'Quiz load query failed';  
    return $distArray;  
}
```



2. Major types of programming language

- Procedural programming language
- Functional programming language
- Object-oriented programming language
- Scripting language
- Logic programming language



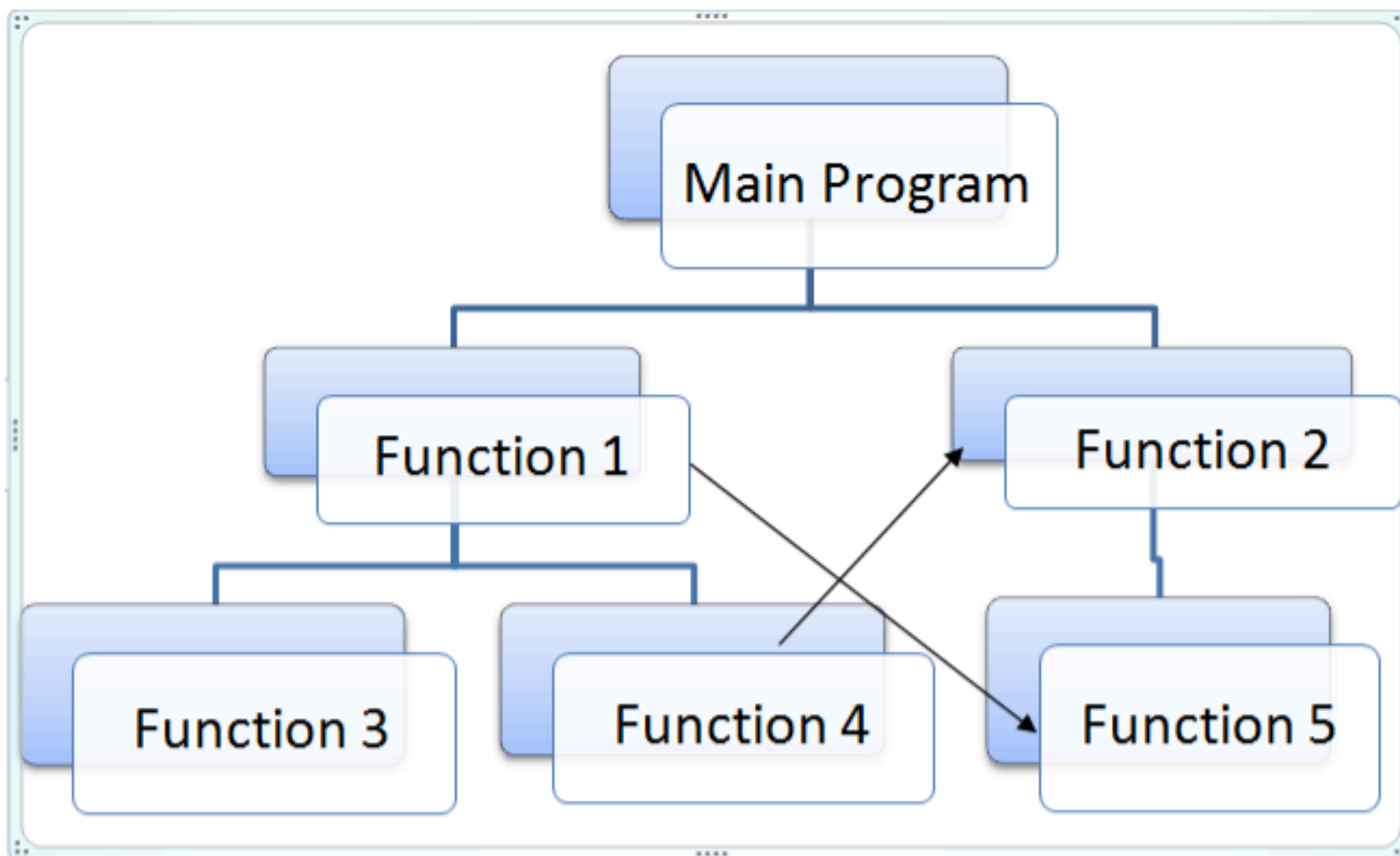
* Some languages may be associated with multiple types.





2.1 Procedural programming language

- Define a sequence of commands to achieve the desired result.
- Each set of steps is called a “**procedure**” or “**function**”.
- Each program has one or more procedures.



Do (1) and (2)

After (1),

if .. do (3) and (4)

else .. do (2)

...

Examples

- C
- Pascal
- Fortran
- Basic

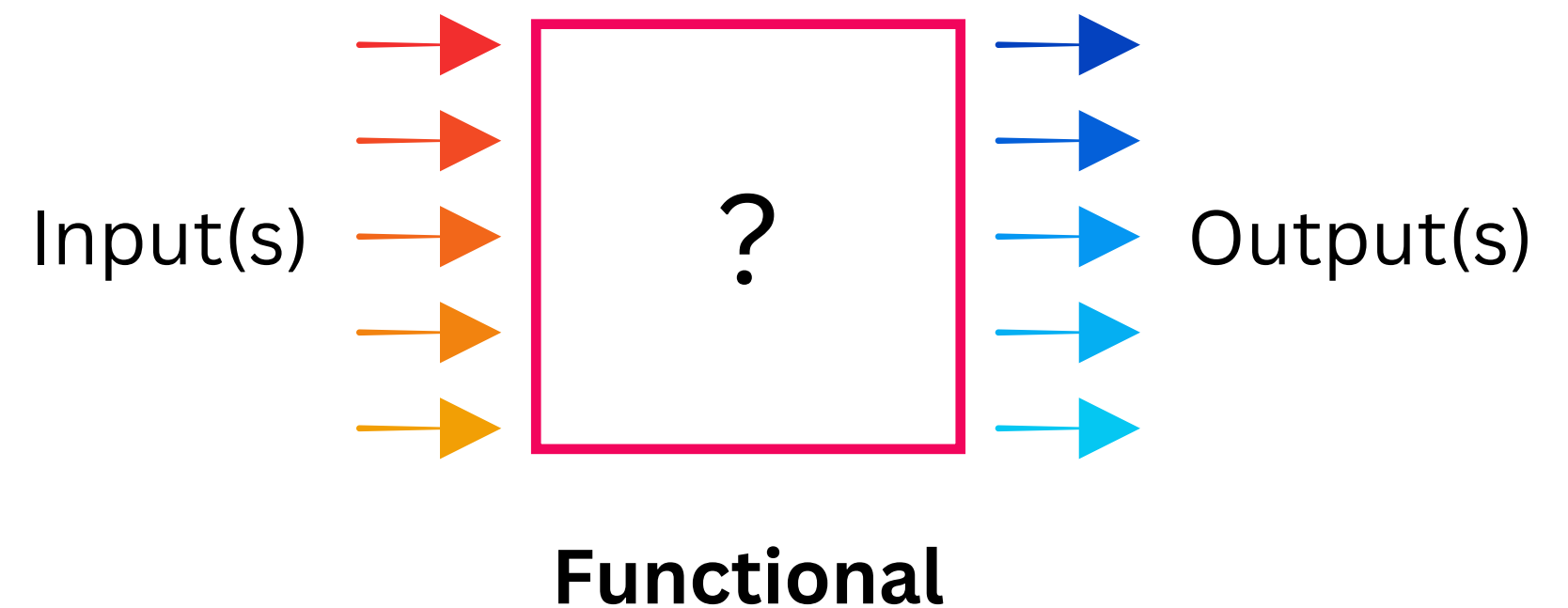
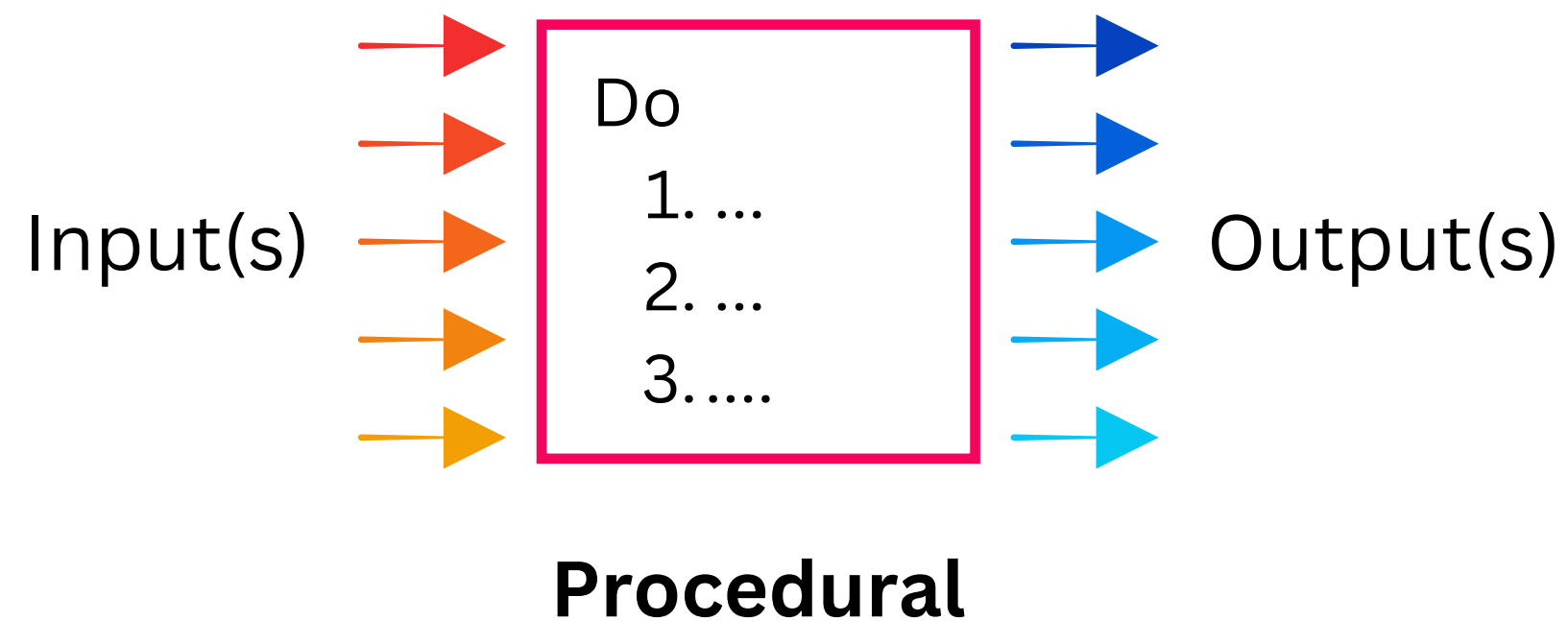


2.2 Functional programming language

- Focus on output rather than process.
- Each function receives input(s) and returns a result.

Examples

Scala, Erlang, Wolfram, Elixir, F#



Example: Calculate the Euclidean distance $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

C (Procedural)

```
double euclideanDistance(double x1, double y1, double x2, double y2) {  
    double dx = x2 - x1;  
    double dy = y2 - y1;  
    return sqrt(dx*dx + dy*dy);  
}
```

Wolfram (Functional)

```
point1 = {3, 4};  
point2 = {6, 8};  
  
distance = EuclideanDistance[point1, point2];
```



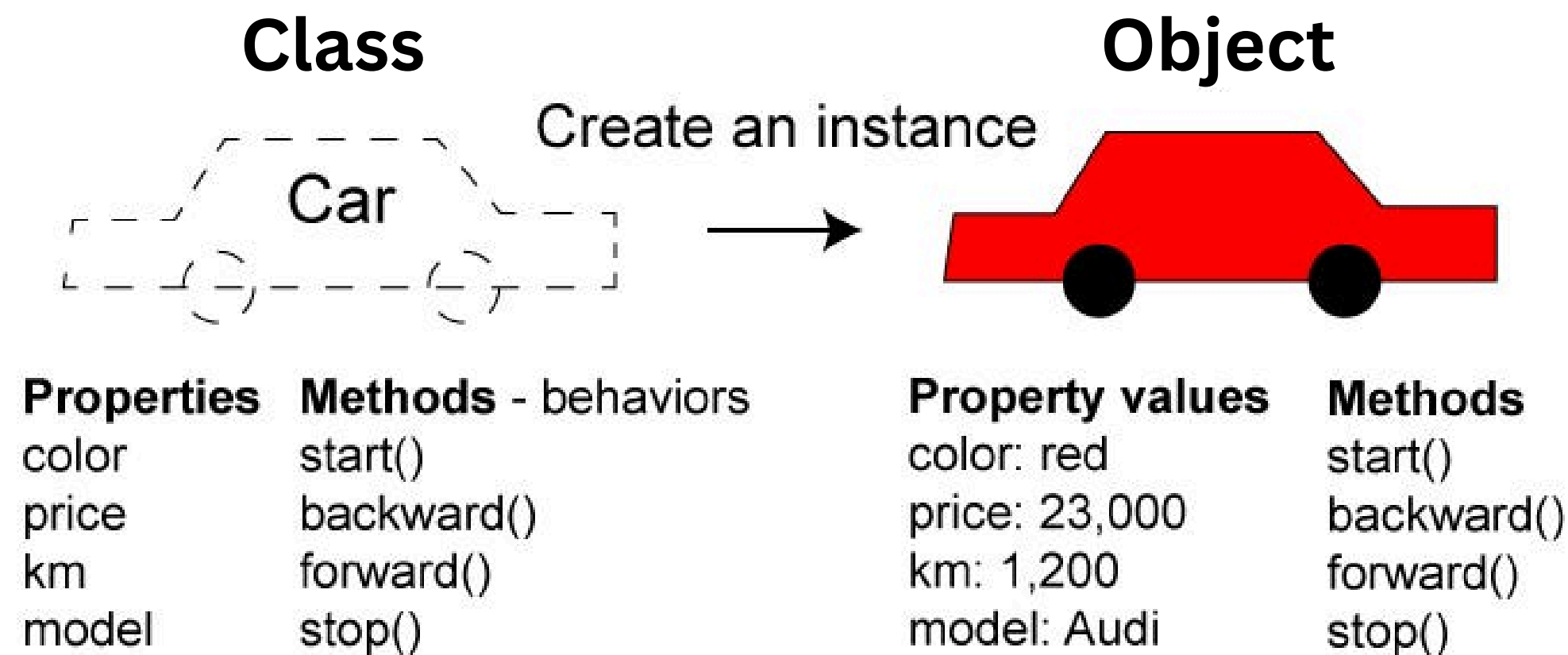


2.3 Object-oriented programming language

- An object contains data (attribute) and how to process those data (method).
- A class is a prototype that can be used to create objects (instances).
- Objects can be reused within a program or in other programs.
- Useful for complex programs because the code can be reused.

Examples

- Java
- Python
- PHP
- C#
- Ruby





2.4 Scripting language

- Used to manipulate, customize, and automate repetitive tasks.

Examples

Bash, PowerShell, AWK, Lua, PHP, Python, Perl

The example on the right is a bash script used to change all the file names in the working directory to all lowercases.



```
# Iterate over all files in the directory
for file in "$directory"/*; do
    # Check if it's a file (not a directory)
    if [ -f "$file" ]; then
        # Get the base name and directory name of the file
        base_name=$(basename "$file")
        dir_name=$(dirname "$file")

        # Convert the filename to lowercase
        lower_case_name=$(echo "$base_name" | tr 'A-Z' 'a-z')

        # Create the new file name with the lowercase name
        new_name="${dir_name}/${lower_case_name}"

        # Rename the file
        mv "$file" "$new_name"

        # Print the rename action
        echo "Renamed $file to $new_name"
    fi
done
```



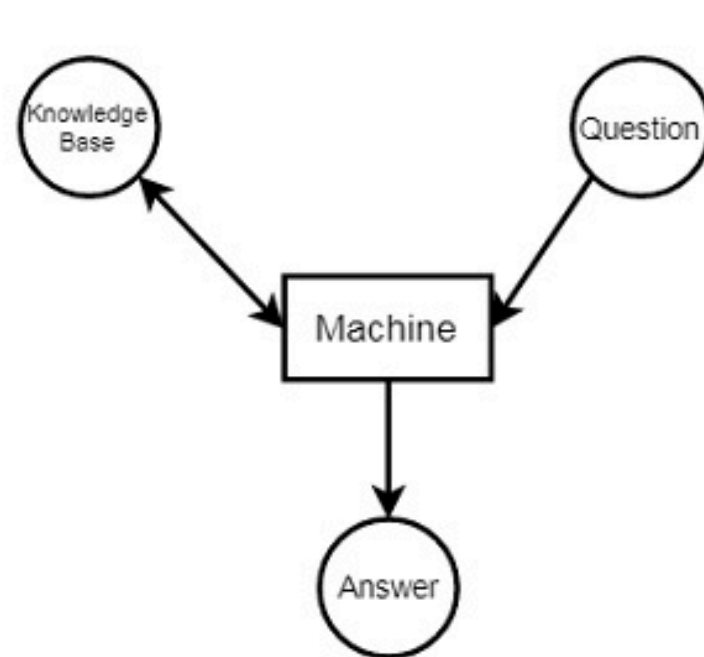
2.5 Logic programming language

Instead of telling a computer what to do, a logic programming language expresses a series of facts and rules to instruct the computer on how to make decisions.

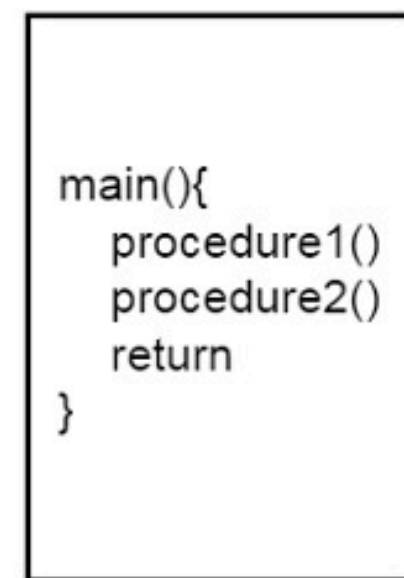
Logic programming languages are well-suited for developing logic-based artificial intelligence applications.

Examples

- Prolog
- Absys
- Datalog
- Alma-0



Logical Programming



Functional Programming

3. Low- and high-level language

Low-level language

- Machine code
- Assembly language

High-level language

- C
- Java
- Python
- ...

Natural language

- Human language





3.1 Low-level language

Machine code

- Executable binary code (0 and 1)
- The only language that computers understand (at the hardware level)
- Normal human cannot read

```
00000000: 01001101 01011010 10010000 00000000 00000011 00000000
00000006: 00000000 00000000 00000100 00000000 00000000 00000000
0000000c: 11111111 11111111 00000000 00000000 10111000 00000000
00000012: 00000000 00000000 00000000 00000000 00000000 00000000
00000018: 01000000 00000000 00000000 00000000 00000000 00000000
0000001e: 00000000 00000000 00000000 00000000 00000000 00000000
00000024: 00000000 00000000 00000000 00000000 00000000 00000000
0000002a: 00000000 00000000 00000000 00000000 00000000 00000000
00000030: 00000000 00000000 00000000 00000000 00000000 00000000
00000036: 00000000 00000000 00000000 00000000 00000000 00000000
0000003c: 10000000 00000000 00000000 00000000 00011110 00011111
00000042: 10111010 00001110 00000000 10110100 00001001 11001101
00000048: 00100001 10111000 00000001 01001100 11001101 00100001
0000004e: 01010100 01101000 01101001 01110011 00100000 01110000
00000054: 01110010 01101111 01100111 01110010 01100001 01101101
0000005a: 00100000 01100011 01100001 01101110 01101110 01101111
00000060: 01110100 00100000 01100010 01100101 00100000 01110010
00000066: 01110101 01101110 00100000 01101001 01101110 00100000
0000006c: 01000100 01001111 01010011 00100000 01101101 01101111
00000072: 01100100 01100101 00101110 00001101 00001101 00001010
00000078: 00100100 00000000 00000000 00000000 00000000 00000000
0000007e: 00000000 00000000 01010000 01000101 00000000 00000000
```

Assembly language

- Among the oldest programming languages
- Human-readable
- To be translated into machine code
- Programmers must know the system memory layout and positions

```
section .text
_start:  mov     rax, 1
         mov     rdi, 1
         mov     rsi, message
         mov     rdx, 13
         syscall
         mov     rax, 60
         xor     rdi, rdi
         syscall
```





3.2 High-level language

- Highly human-readable
- To be translated into machine code
- Deal with variables, loops, arrays, functions, objects, and complex expressions.
- Focus on usability over optimal program efficiency.

```
a = 10
b = 20
if a == b:
    print('yes')
else:
    print('no')
```



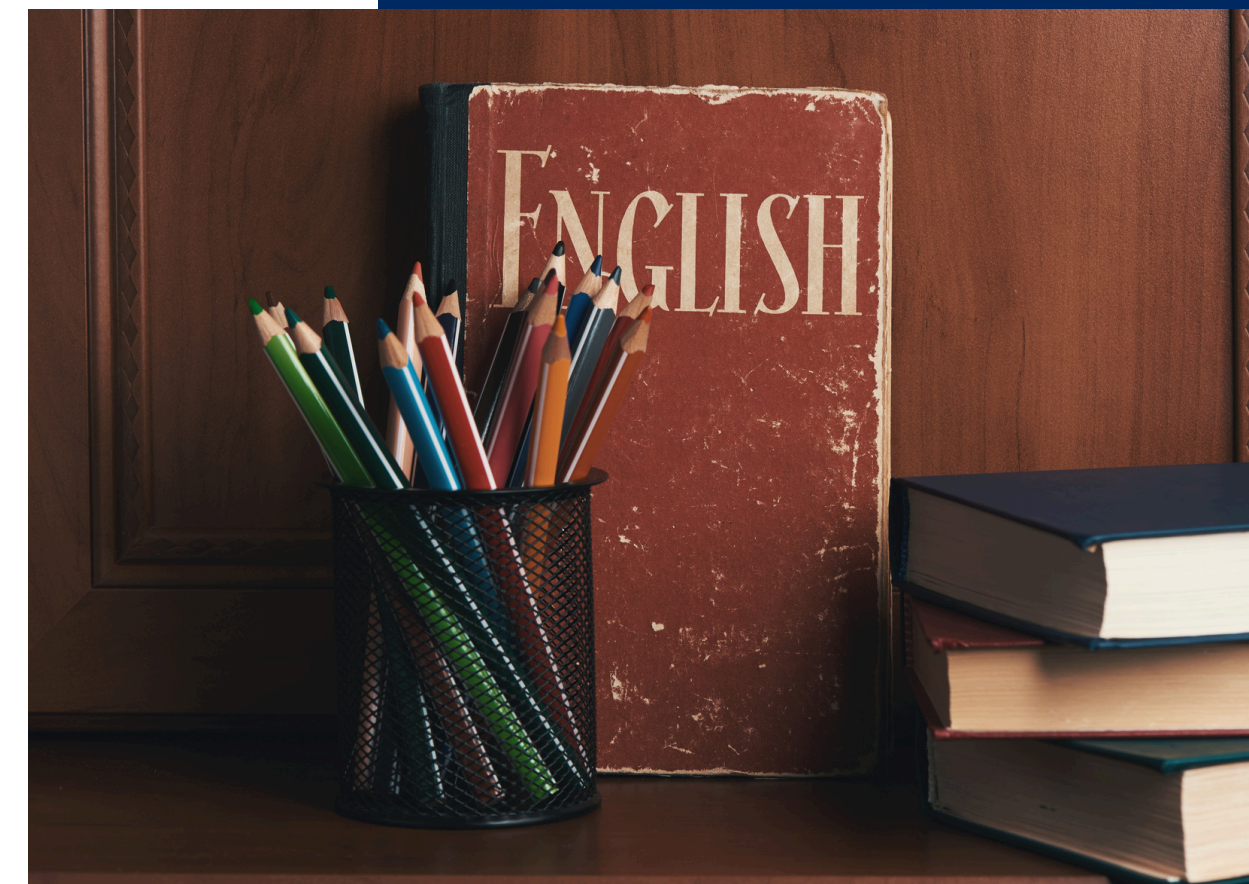


3.3 Natural language (programming)

- Natural Language Programming (NLP) is structured like human language sentences, e.g. English.
- It is not to be mixed up with natural language interfacing or voice control.

Sample:

- Add 4 to the routine's parameter size.
- Subtract the length from the local's offset.
- Multiply the type's scale by the base type's scale.
- Calculate the length of the field's type.
- Round the address up to the nearest multiple of 4096.



4. Web programming language

Markup language

- HTML
- CSS

Front-end language

- JavaScript

Back-end language

- JavaScript (node.js)
- PHP, SQL
- Java, Python, Ruby, etc



* Some languages may be associated with multiple types.





4.1 Markup language

- Use text encoding to control the formatting and structure of content on a page or document.
- HTML and CSS are examples.
- Not technically considered as programming languages because by themselves they cannot receive input, process the data, and display the result dynamically.

Code

```
<table style='background: #F0F0F0; padding: 1px;
border: thin solid #DDDDDD; margin-left: 1em'>
<tr> <td>Name:</td><td>this is the name</td>
</tr>
<tr> <td>Given Name:</td><td>this is the
givenname</td>
</tr>
<tr> <td>Family name:</td><td>this is the
family_name</td>
</tr>
<tr> <td>Home page:</td><td><a
href='http://somewhere.com'>http://somewhere.com</a
></td>
```

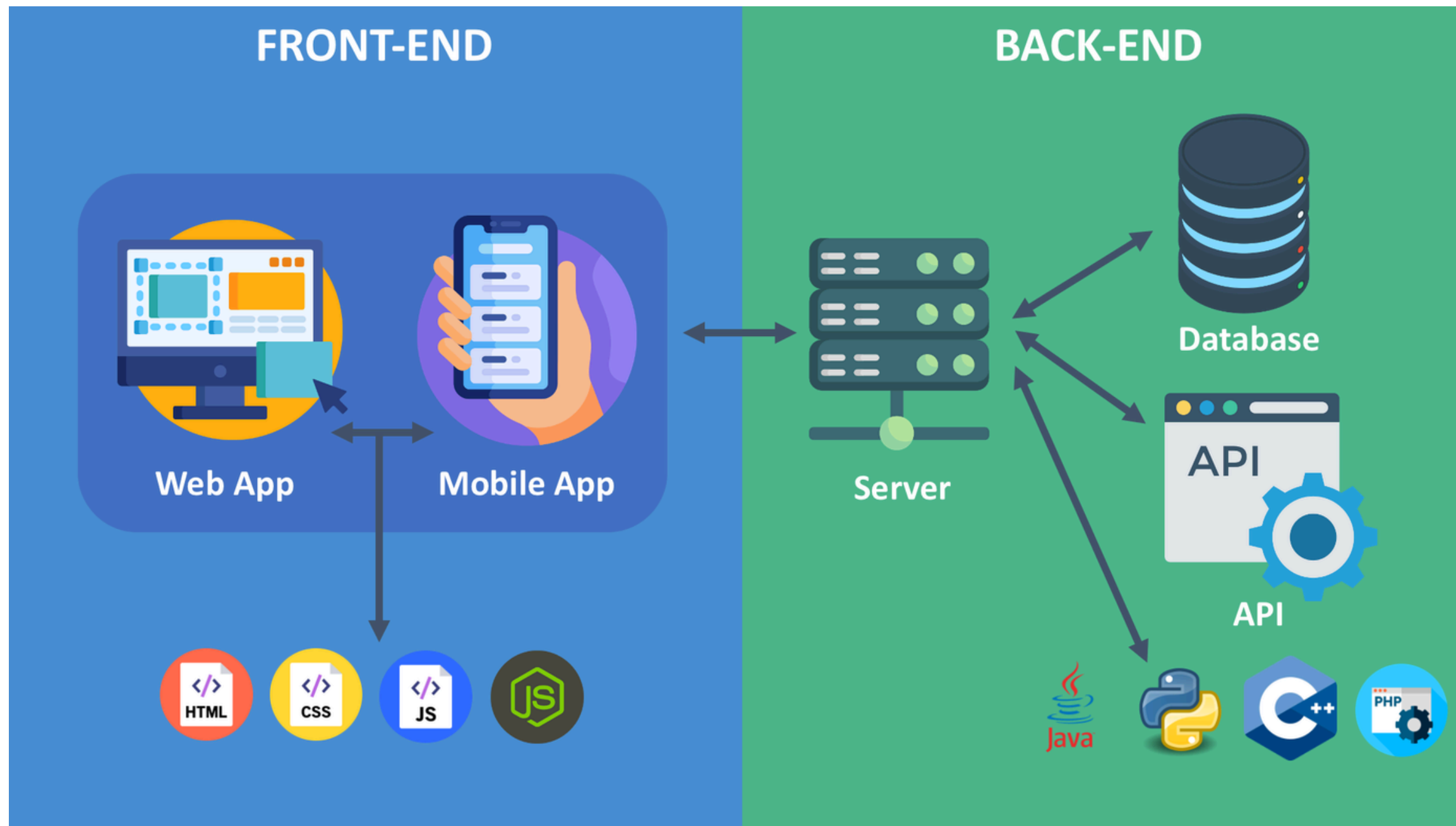
Rendering

Name:	this is the name
Given Name:	this is the givenname
Family name:	this is the family_name
Home page:	http://somewhere.com
Depiction:	
Knows:	list of known people



4.2 Front-end and back-end language

Often used in terms of web applications.





Front-end language

- Concerned with the "user" aspect of the software.
- Handles text, colors, buttons, images, and navigation.
- Examples are HTML, CSS, and JavaScript (JavaScript is typically considered as the only front-end language)

Back-end language

- Handles the server-side management of the software, including database, storage, and communication.
- Users do not directly experience.
- Examples are Node.js, PHP, Java, and Python.

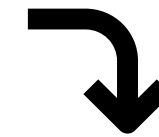




4.3 SQL

- Used to communicate with and manipulate databases
- For relational database management systems (RDBMS), where data are in a tabular format with rows and columns.

```
SELECT * FROM Customers WHERE Country='Mexico';
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

SQL is simple and straightforward.

You can easily guess what the above query does.

5. Compiled and interpreted language

Compiled language

C, C++, C#, Rust, Go, Swift, etc.



Interpreted language

JavaScript, PHP, Ruby, Perl, etc.



Others

Java, Python, R, etc.



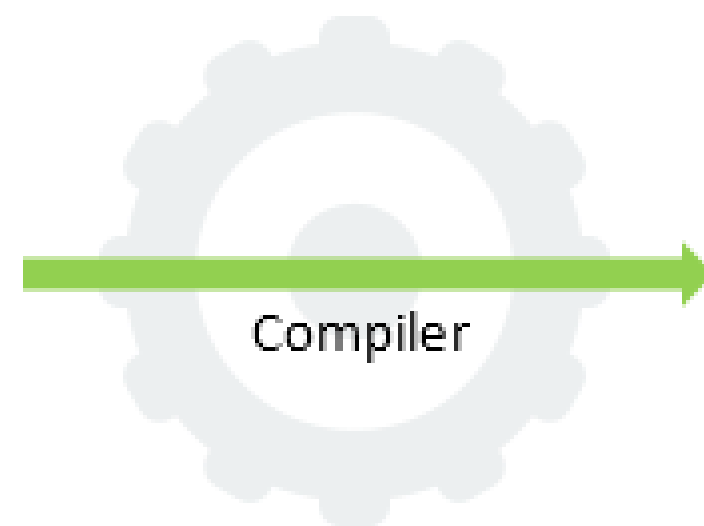


Compiled language

- Source code is translated into machine code before execution.
- The compiled program runs independently of the source code.
- Offer better performance but modification can be difficult if the source code is not available.
- Examples are C, C++, and Rust.

```
#include  
<stdio.h>  
int main()  
{  
    printf("Hello")  
;  
    return 0;  
}
```

Source code



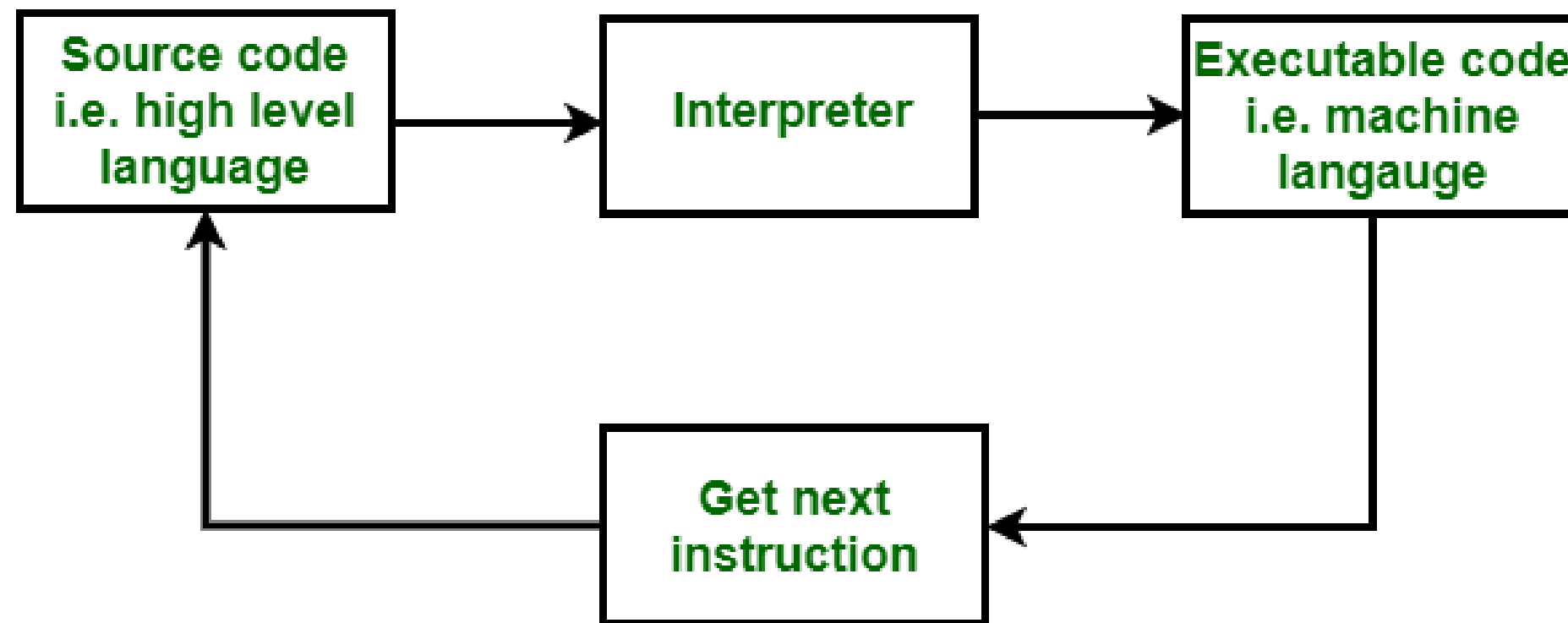
```
100010101010101  
000100101010111  
011111100110000  
001011001101010  
010111011100011  
011111001111000  
000110011110101  
010010010101000
```

Executable code



Interpreted language

- An interpreter reads and executes the source code line by line at run time with no separate compilation step.
- The source code is required to run the program.
- Offer greater flexibility, allow modifications at runtime.
- Have slightly lower performance.
- Examples are Python, JavaScript, and Ruby.



6. Compiled language

- C
- C++
- C#
- Rust
- Go
- Swift
- ...



6.1 C

- One of the oldest programming languages
- Known for its efficiency and versatility
- Commonly used in system programming, embedded systems, and high-performance computing

```
#include <stdio.h>
int main(void) {
    printf("Hello World!\n");
    return 0;
}
```

C is a compiled language. Typically, you need to install a C compiler (gcc) to compile the code (.c) into an executable program (.exe).



6.2 C++

- An extension of the C programming language
- With features like (OOP) and templates
- A wide range of applications, including game development, system software, and large-scale applications.

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!";
    return 0;
}
```



6.3 C#

- Developed by Microsoft as part of its .NET initiative
- More object-oriented
- Widely used for building desktop, web, and mobile apps on the Microsoft platform

```
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```



6.4 Rust

- Systems programming language
- Emphasizes safety, performance, and concurrency
- Designed to prevent common programming errors such as null pointer dereferencing and buffer overflows

```
fn main() {  
    println!("Hello World!");  
}
```



6.5 Go (Golang)

- Developed by Google
- Known for its simplicity, efficiency, and concurrency.
- Go is commonly used for web development, networking, and distributed systems.

```
package main
import ("fmt")

func main() {
    fmt.Println("Hello World!")
}
```



6.6 Swift

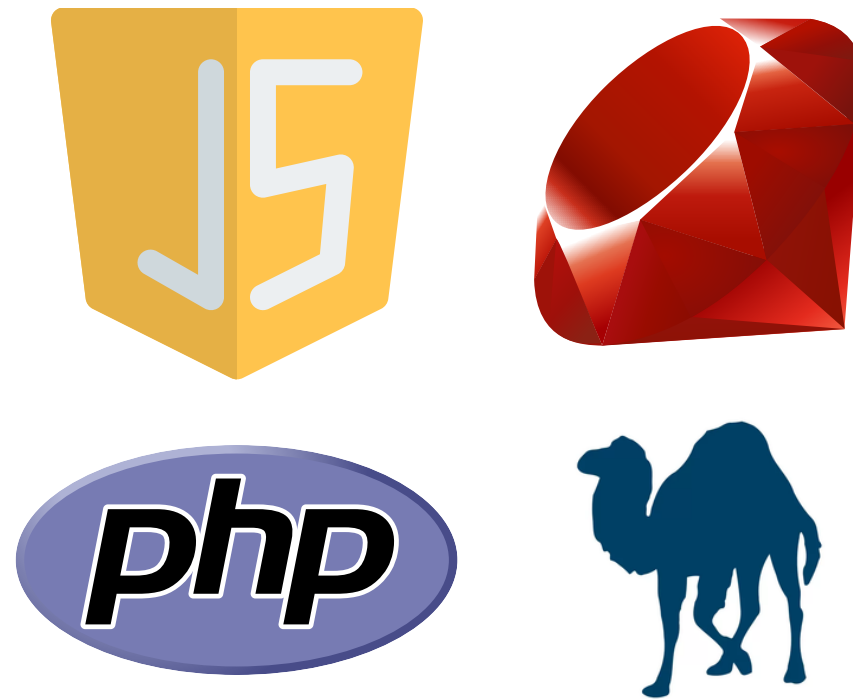
- Developed by Apple for building iOS, macOS, watchOS, and tvOS applications
- Designed to be fast, safe, and expressive, with modern features like optionals, closures, and generics.

```
print("Hello World!")
```



7. Interpreted language

- JavaScript
- PHP
- Ruby
- Perl
- ...



7.1 JavaScript

- Primarily used for client-side web development.
- Add interactivity and dynamic behavior to web pages
- Also used for server-side development (Node.js) and mobile app development (React Native)
- Modern websites are typically created under a JavaScript framework like React (by Meta) and Angular (by Google)

```
document.write('Hello, World!');
```

JavaScript (JS) is not similar to or related to Java, another programming language.



7.2 PHP

- Server-side language designed for web development
- Widely used for building dynamic websites and web apps
- Handling form data, interacting with databases, and generating HTML content.

```
<?php  
echo "Hello World!";  
?>
```



7.3 Ruby

- Commonly used for web development with frameworks like Ruby on Rails
- Also used for automation and scripting
- Known for its simplicity and productivity

```
puts "Hello World!";
```



7.4 Perl

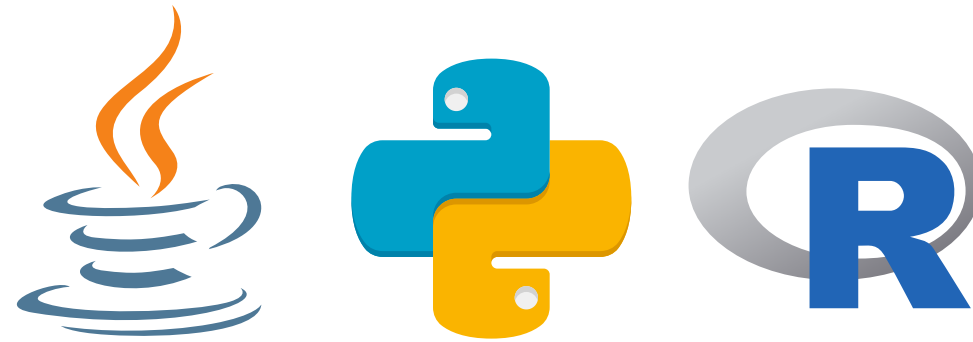
- Commonly used for system administration, web development, and network programming
- Known for its powerful text processing and extensive library

```
print "Hello World!\n";
```



8 Others

- Java
- Python
- R



8.1 Java

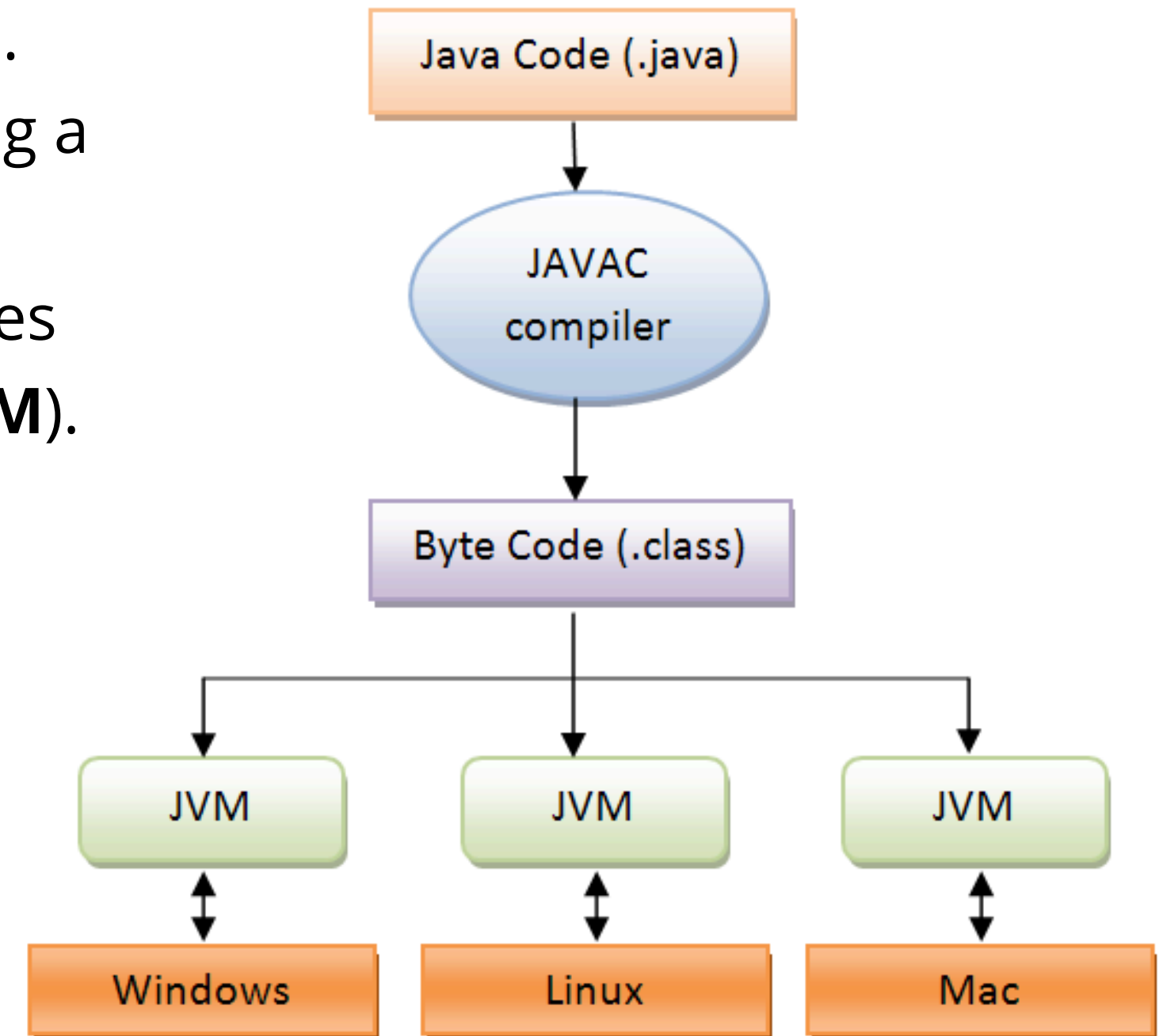
- Java is an object-oriented general-purpose programming language developed by Sun Microsystems in the early 1990s.
- Primarily used for web applications.
- Java is platform independent, making it suitable for apps running on different platforms.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```



Java is considered as both interpreted and compiled.

- Source code is first compiled into **bytecode** using a Java compiler (**JAVAC**).
- Bytecode runs on the target device which requires a Java interpreter called Java Virtual Machine (**JVM**).



8.2 Python

- Known for its simplicity, readability, and versatility.
- Recommended for beginners due to its user-friendly interface and straightforward syntax.
- Extensively used in scientific computing, data analysis, AI, and automation.
- Also used in web development under frameworks like Django and Flask.

```
print("Hello World!")
```



Python is considered as both interpreted and compiled.

- Source code is first converted into bytecode.
- Bytecode is then executed by the Python virtual machine on the target device.

