7

Programming languages

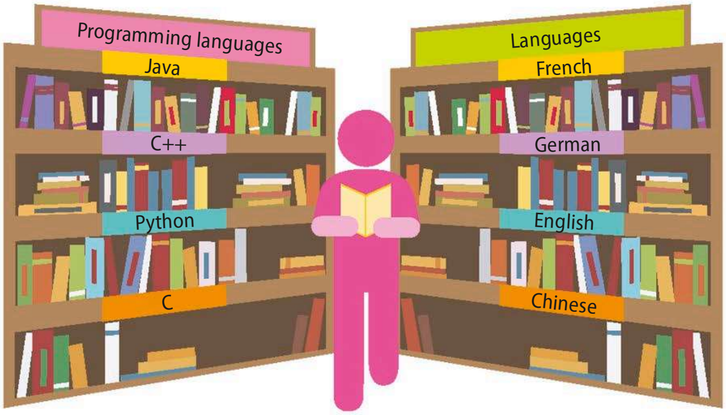
What do programming languages do?

SEE ALSO
❮ **76–77** Algorithms
❮ **108–109** Program structures
❮ **110–111** Translation
Types of programming language **120–121** ❯

Programming languages were developed to help humans communicate with computers. The fundamental challenge is translating instructions humans can understand into ones computers can.

Programming languages

A programming language is a formalized set of words and symbols that allows people to give instructions to computers. Just like human languages, each programming language has its own vocabulary and grammar. Translating an algorithm written in English into a programming language enables a computer to understand and carry out the instructions.



▷ **Multilingual**
It's possible to translate text into many different human languages. Similarly, computer instructions can be written in many different programming languages.

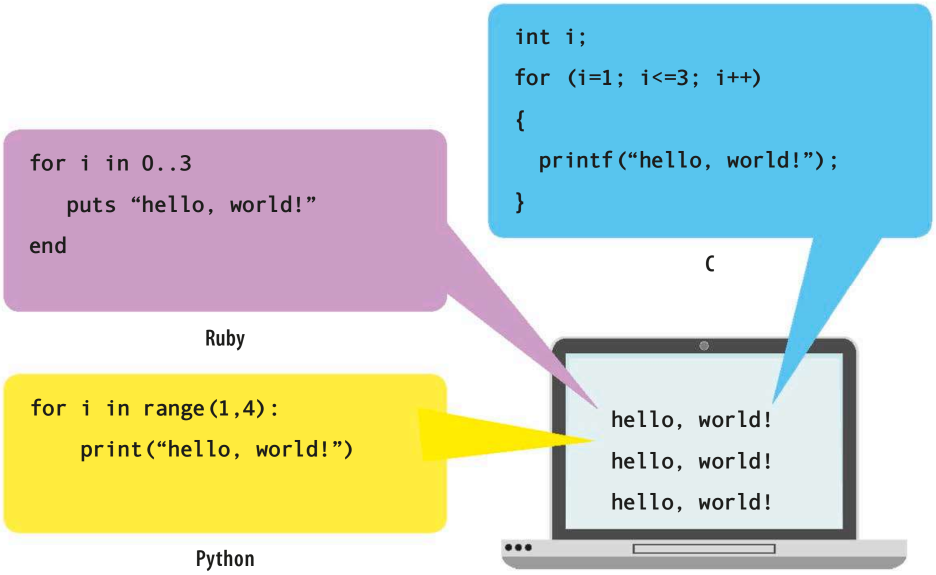
IN DEPTH

Translation

There is a variety of ways to translate programming languages into the binary code (sometimes called machine code) a computer understands. Some languages, like C and C++, use a compiler. This produces a new file containing machine code that can then be run. Scripting languages, like Python and JavaScript, use an interpreter, which translates and runs the code in a single process. Assembly languages use an assembler that, like a compiler, produces a file containing machine code.

Common features

All programming languages have certain underlying features. These are: making decisions, repeating instructions, and storing values in named containers. The words used for these features vary from language to language, but they're all doing essentially the same thing. Being familiar with these concepts in one language makes learning another language much easier.



```
int i;
for (i=1; i<=3; i++)
{
  printf(“hello, world!”);
}
```

C

```
for i in 0..3
   puts “hello, world!”
end
```

Ruby

```
for i in range(1,4):
    print(“hello, world!”)
```

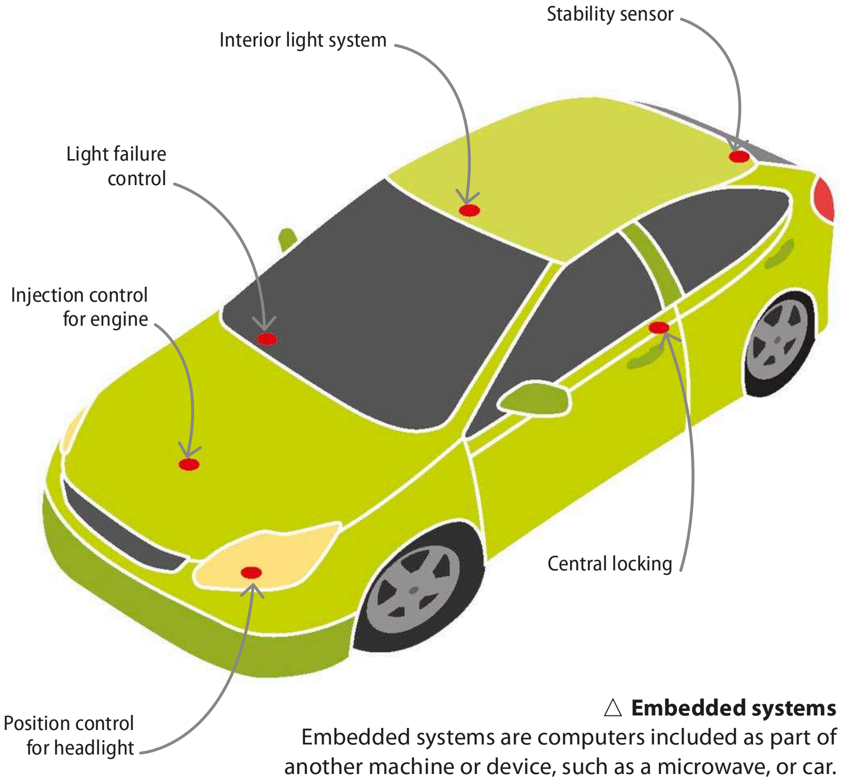
Python

```
hello, world!
hello, world!
hello, world!
```

▷ **Same outcome**
The three programs shown here look quite different, but they're all examples of a programming concept called "for loop" that is used to display a message three times.

High- and low-level languages

The term "programming languages" is usually used for high-level languages. These allow programmers to use a language closer to human language. Low-level languages work with internal hardware like registers and memory, and are tied to a specific type of computer. Programs written in a high-level language can be run on any computer with the relevant compiler or interpreter.

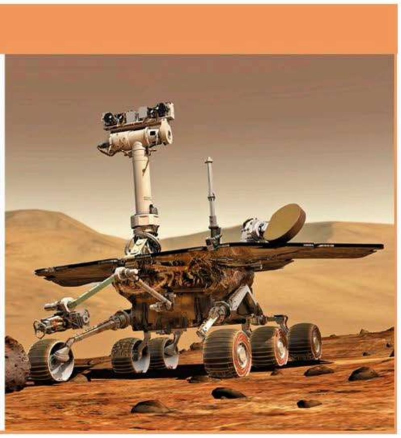


△ **Embedded systems**
Embedded systems are computers included as part of another machine or device, such as a microwave, or car. They are often programmed in C as, despite being a high-level language, it allows programmers a great deal of control over how the code runs at the level of registers.

TOP TECH

The Mars Rover

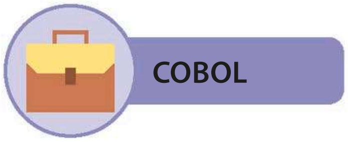
One of the most famous devices to feature an embedded system is the Mars Rover, *Curiosity*. The self-propelling robot is programmed to explore Mars and send back data. Code for the rover is written mainly in C and has been very thoroughly tested to try to ensure the rover doesn't accidentally drive into a rock and damage itself.



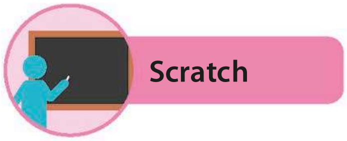
Special purpose

In the initial stages of computing, programs were written in binary code, or assembly language. Since then, programming languages have been developed as tools to meet a need or fulfil a purpose. Examples include languages that allow mathematicians and scientists to include formulas, languages used to teach people how to program, or languages that could be used to develop artificial intelligence.

△ **Scientific computing**
Fortran was designed to allow scientists to write programs that included mathematical formulas. Its name is short for "Formula Translation".



△ **Down to business**
Short for common business-oriented language, COBOL was developed to make it easier and cheaper for companies to write business-related software.



△ **Code for kids**
Scratch was created as a language that would make learning to code easy and fun for children aged between eight and 16.

△ **Thinking machines**
Lisp was based on a mathematical definition of programming languages and soon became popular with researchers studying artificial intelligence.

Types of programming language

SEE ALSO

❮ 108–109 Program structures

❮ 114–115 Software errors

C and C ++ 126–127 ❯

Python 130–131 ❯

Scratch 136–137 ❯

There are lots of different ways to group programming languages, and most languages fall into more than one group. A useful way of grouping is according to the features a language has.

Styles of programming language

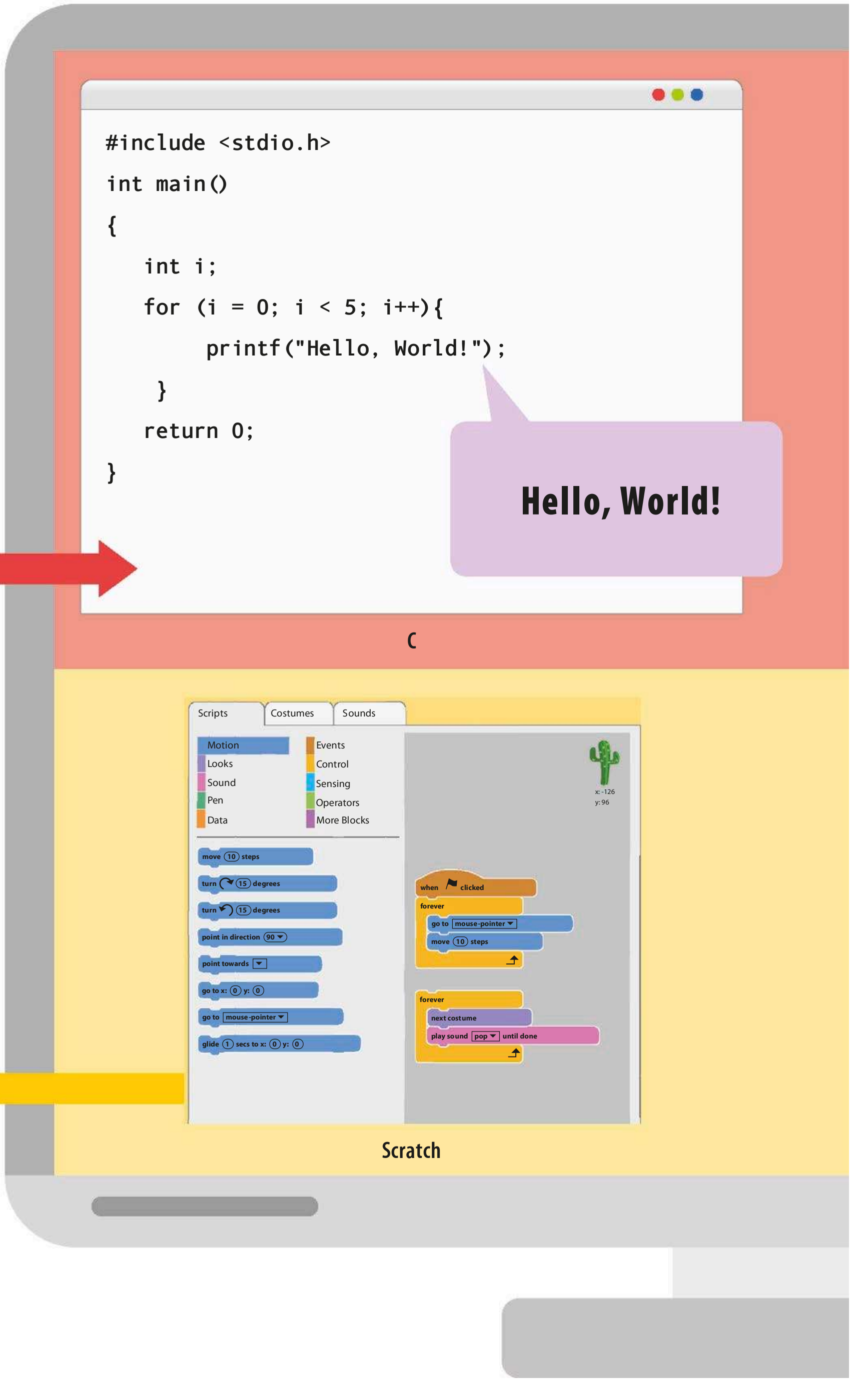
Different styles of programming are sometimes called paradigms. They represent varied ways of thinking about computation. Some styles of programming are better at solving particular problems than others. Sometimes, there's no obvious approach and programmers will simply choose the language they are most comfortable with.

Imperative

The imperative style of programming is best described as a recipe or a knitting pattern. It is a series of commands that are executed one after the other. The recipe changes the state, or condition, of the ingredients from uncooked to cooked. The state of a computer is the data stored in its memory. When it runs a program, the commands in the program change this state. Imperative languages include variables, which hold data, and control structures, such as loops and conditional branches.

Visual

The first style of programming that children encounter is often visual programming. This describes languages where the programmer fits together blocks that represent instructions. Many visual languages are designed as educational tools. They allow children, or other new programmers, to become familiar with programming concepts without needing to type in commands. This allows them to focus on solving the problem without having to worry about programming errors.



```
#include <stdio.h>
int main()
{
   int i;
   for (i = 0; i < 5; i++){
        printf("Hello, World!");
    }
   return 0;
}
```

C

Scratch

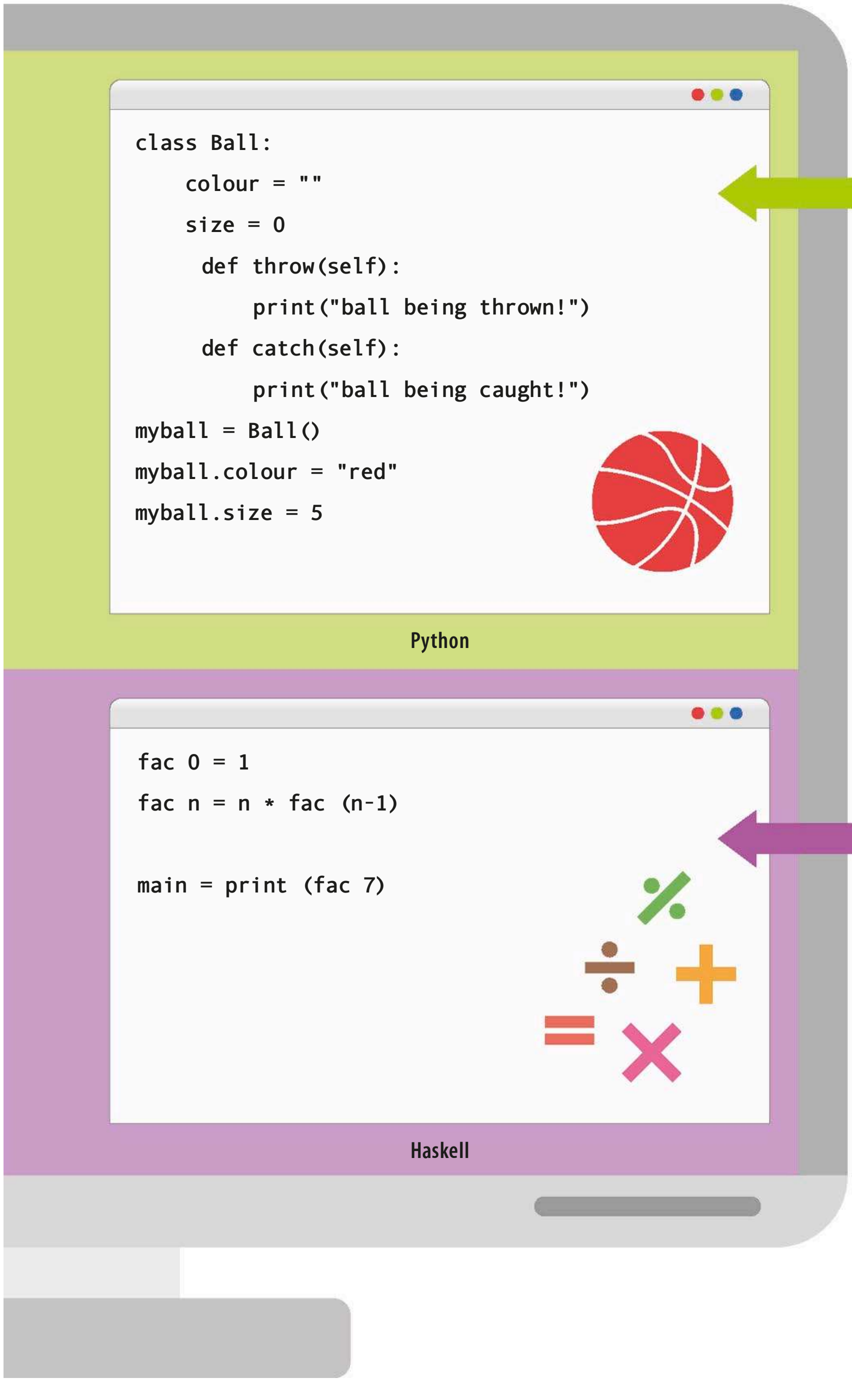
IN DEPTH

Hello, World!

Traditionally, the first program a new programmer writes is the "Hello, World!" program. This simply prints the phrase "Hello, World!" on the screen. Even experienced programmers learning a new language often start off in the same way, and it's also a good first check to see if a newly installed system is working properly. The tradition was introduced in the book *The C Programming Language*, published in 1978.

Object-oriented

This style of programming includes the concept of objects that model real-world things. An object usually has fields (containing data) and methods, (containing code) that represent behaviours. So, a ball object might have the fields colour and size, since these are characteristics of a ball, and the method bounce, since this is what a ball does. Objects are instances of classes, definitions of what a particular object would look like. This means the object ball is an instance of the class ball, similar to any real ball being an instance of the idea of a ball.



```
class Ball:
    colour = ""
    size = 0
     def throw(self):
         print("ball being thrown!")
     def catch(self):
         print("ball being caught!")
myball = Ball()
myball.colour = "red"
myball.size = 5
```

Python

```
fac 0 = 1
fac n = n * fac (n-1)

main = print (fac 7)
```

Haskell

Functional

Functional languages define a program as a series of mathematical functions. A functional language is described as pure if it doesn't affect the computer's state, or impure if it does. One major feature of functional languages is that they don't use loops to repeat operations. Instead, they use a recursive function, which calls itself as part of its own definition. Another notable feature is pattern matching, where a function decides what to do by looking at the value it's been given and seeing which of the several patterns it matches.

REAL WORLD

Natural-language programming

There are programming languages where the code looks like normal text or natural language. However, these aren't serious languages, as even tiny calculations take a large amount of code, and they're usually created just for fun. These include Shakespeare, where programs look like very confused Shakespearean plays, and Chef, where each program is written as a cooking recipe.

Language breakthroughs

SEE ALSO
❮ 110–111 Translation
❮ 114–115 Software errors
❮ 118–119 What do programming languages do?

For many tasks, high-level languages are better than machine code or assembly language. Two early programming languages, Fortran and BASIC, helped convince people of this.

Fortran

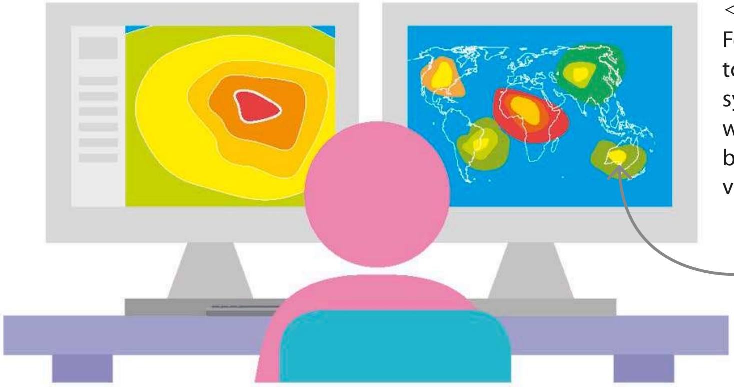
Short for “Formula Translation”, Fortran was developed in 1957 by a team at IBM, led by American computer scientist John Backus (1924–2007). Unlike earlier compilers, Fortran’s compiler produced machine code that ran almost as fast as hand-written code. Early Fortran programs were transformed a line at a time into patterns of holes on punched cards.

▷ **Selling points**
Fortran’s main selling point was that it made writing programs much easier because its syntax was much closer to English when compared with assembly languages.

```
C AREA OF A TRIANGLE - HERON’S FORMULA
C INPUT - CARD READER UNIT 5, INTEGER INPUT
C OUTPUT -
      READ(5,501) A,B,C
  501 FORMAT(3I5)
      IF(A.EQ.0 .OR. B.EQ.0 .OR. C.EQ.0) STOP 1
      S = (A + B + C) / 2.0
      AREA = SQRT( S * (S - A) * (S - B) * (S - C) )
      WRITE(6,601) A,B,C,AREA
  601 FORMAT(4H A= ,I5,5H  B= ,I5,5H  C= ,I5,8H  AREA= ,F10.2,
     $13H SQUARE UNITS)
      STOP
      END
```

What is it for?

Fortran is mainly used for writing programs involving scientific and mathematical problems. It was the first language to have in-built support for mathematical concepts such as complex numbers, used in many areas of physics. Fortran has been used for systems investigating nuclear physics, quantum mechanics, and the operation of aeroplanes and wind turbines.

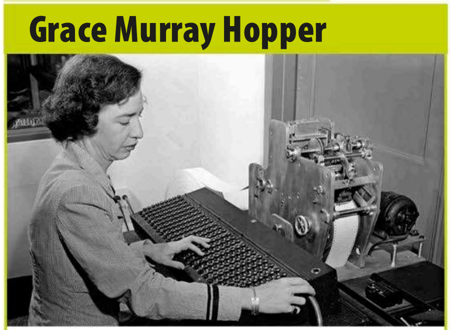


◁ **Scientific systems**
Fortran is still in use today. Many scientific systems use code that was written decades ago but has proved to be very reliable over time.

Fortran is also used for weather prediction systems.

BIOGRAPHY

Grace Murray Hopper



American mathematician and Rear Admiral in the US Navy, Grace Hopper (1906–1992) was involved in developing COBOL, a programming language for businesses. She developed one of the first compilers, and her idea of making programming languages more like English helped spread computer usage.

BASIC

BASIC (Beginner's All-purpose Symbolic Instruction Code) was developed at Dartmouth College in the USA in 1964. Maths professors John G Kemeny (1926–1992) and Thomas E Kurtz (b. 1928) wanted a simple language that they could use to teach programming. They also developed a system where programmers could run their code immediately after entering it at a terminal. Before this, students' programs would be queued and run hours later.

> "Everybody… should **learn** how to **program a computer**, because it **teaches** you **how to think.**"
>
> **Steve Jobs (1955–2011), American co-founder of Apple**

Students on a variety of courses used BASIC.

▷ **BASIC for all**
BASIC was designed to be easy to learn for everyone, not just mathematicians. As a result, writing course-related BASIC programs became part of the syllabus for many students at Dartmouth University, regardless if they were studying to be engineers, doctors, or to work in the arts.

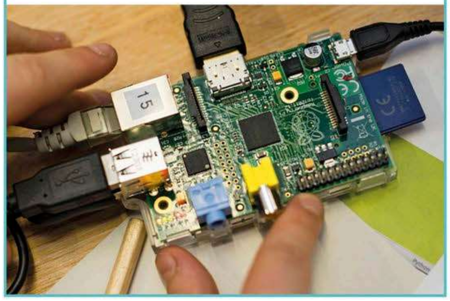
Home computers

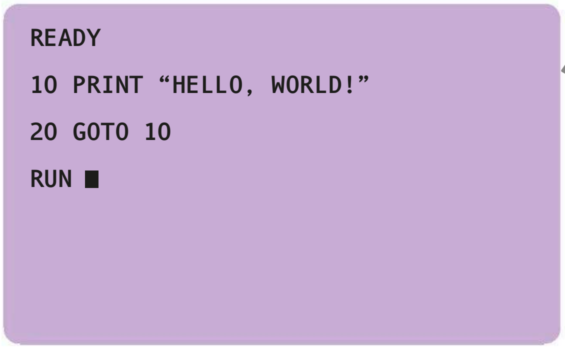
BASIC's popularity really took off in the 1970s and 1980s, when home computers first became available. Most machines came with a version of BASIC, which became many people's introduction to programming. The syntax of the language was straightforward and easy to learn, and allowed people to write software to help them in their businesses, or as a hobby. It gave people the power to "hack their own machine", enabling them to write the software they wanted rather than being restricted to what already existed.

REAL WORLD

Raspberry Pi

Since the 1990s, computers have become increasingly user friendly, to the point that this has discouraged people from experimenting with programming. English inventor Eben Upton (b. 1978) developed the Raspberry Pi in 2012, in an effort to reverse this trend. A very low-cost, simple computer, the Raspberry Pi comes with Python and Scratch as standard, and can be used for all kinds of projects.





```
READY
10 PRINT "HELLO, WORLD!"
20 GOTO 10
RUN ■
```

This program will keep printing "HELLO, WORLD!" until it is stopped.

◁ **BBC BASIC**
In 1981, BBC Micro was launched. It contained a version of BASIC that was used by schoolchildren all across the UK to learn how to code.

Application programming interface

SEE ALSO

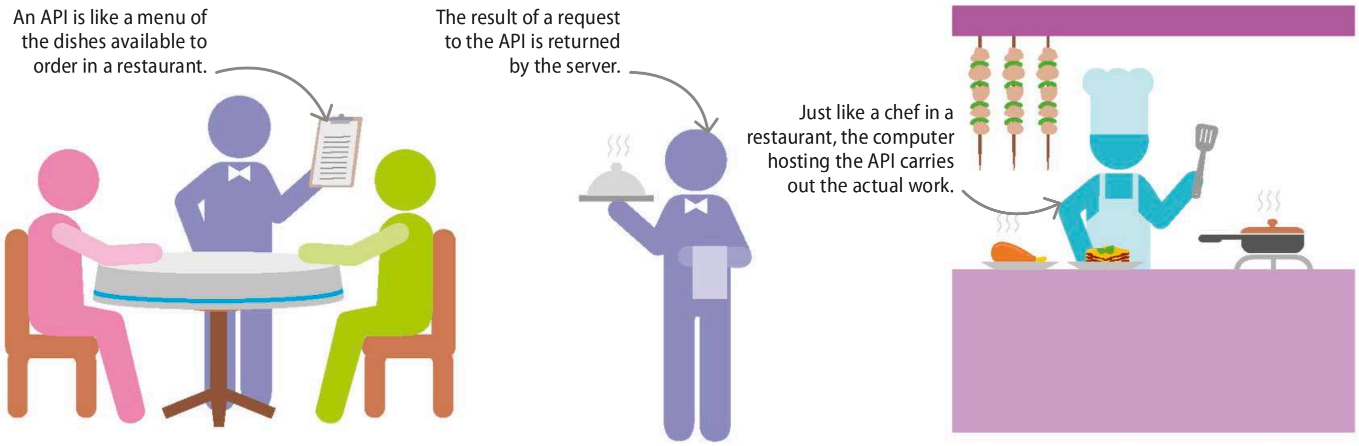
❮ 72–73 Abstraction	
Python	130–131 ❯
Cloud computing	152–153 ❯
The Internet of Things	226–227 ❯

Websites often feature embedded functions such as maps or social media feeds. They do not create these themselves, but use an application programming interface (API).

What is an API?

An interface describes the way a program interacts with another. The other system can be a user, through a user interface (UI), or another program, through an API. APIs make it easier for programmers to use functions and objects from other programs in their code. When an API function is requested, the computer hosting the API executes the function's code and sends the result back to the program requesting it.

▽ **Abstraction**
An API is an abstraction of the program it represents. Just as a menu only lists the names of dishes and not their recipes, an API only shows the features that can be used by other programs. All details of the program's construction are hidden.



An API is like a menu of the dishes available to order in a restaurant.

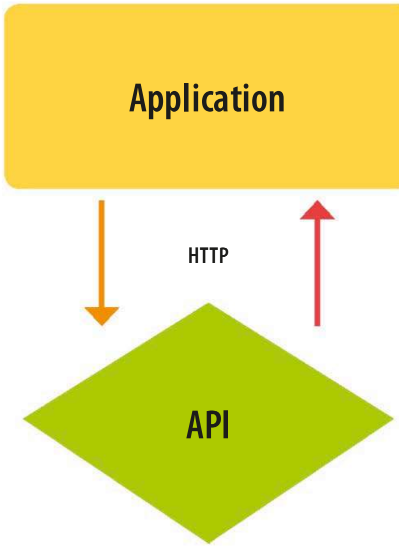
The result of a request to the API is returned by the server.

Just like a chef in a restaurant, the computer hosting the API carries out the actual work.

Which languages?

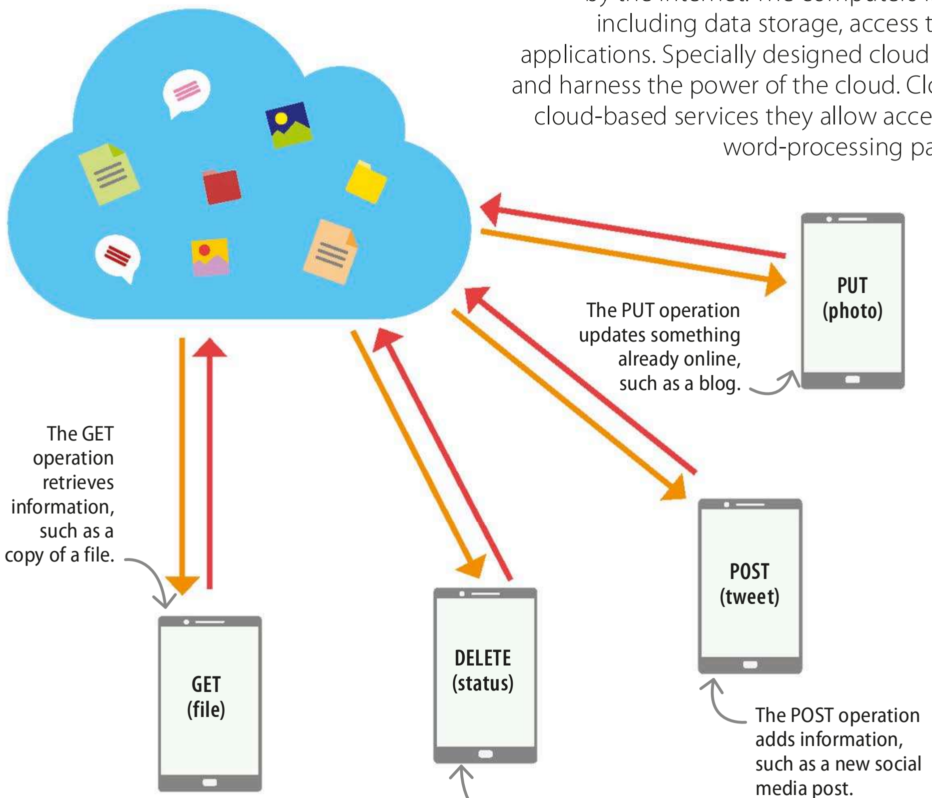
APIs are written in a variety of programming languages, including PHP, Python, Ruby, and Java. Programs that are written in a different language from the one an API is written in can still use its functions. Requests to an API make use of the Hypertext Transfer Protocol (HTTP) used to transfer information across the world wide web. The API returns the result to the calling program in a standard format.

▷ **Helper libraries**
Many APIs provide helper libraries for different programming languages, which makes it easier to call them using another language. For example, a program written in Python can use an API's Python helper library.



Cloud APIs

The cloud is the network of computers across the world connected by the internet. The computers in the cloud provide a wide variety of services, including data storage, access to very powerful computers, and data analysis applications. Specially designed cloud APIs help programmers access these services and harness the power of the cloud. Cloud APIs are grouped according to the sort of cloud-based services they allow access to. These services include software, such as word-processing packages, and hardware, such as storage space.



The PUT operation updates something already online, such as a blog.

The GET operation retrieves information, such as a copy of a file.

The POST operation adds information, such as a new social media post.

The DELETE operation removes information – for example, taking a photo off a social media profile.

Representational state transfer

The cloud computers providing services are known as servers. Other devices, known as clients, make requests for these services using cloud APIs. Most cloud APIs are created using a format called representational state transfer (REST). This means each function performs one of the four standard web operations on data: GET, PUT, POST, or DELETE.

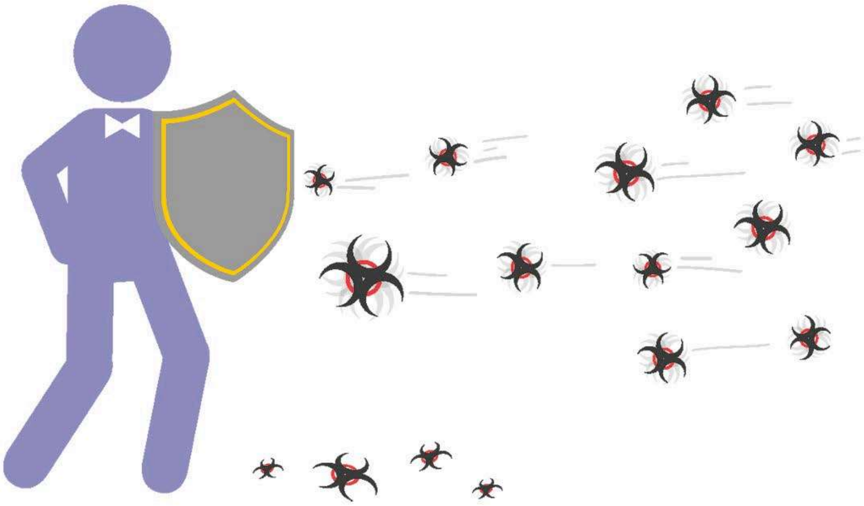
REAL WORLD

Light up tweets

The Twitter API has been used to find out how the world is feeling. A programmer wrote code that monitored the predominant emotions mentioned in tweets from across the world. Anybody can use this code to make their own LED lights glow in a different colour for each emotion.

API security and the Internet of Things

The Internet of Things is the term used for objects in the physical world that are connected to the internet. These items all need APIs that allow programmers to interact with them; for example, by controlling an item or retrieving data created by it. These APIs could potentially all be vulnerable to attack from hackers, giving them access to items in people's homes and cars. To prevent this, APIs have to include a security system, restricting access to those who have a legitimate purpose.



C and C++

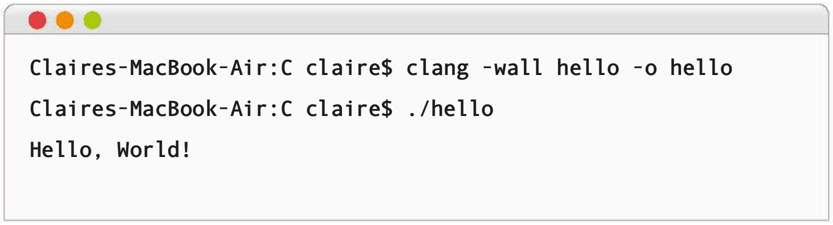
The oddly named C and C++ are two of the most popular programming languages in existence. They have been used to create a huge amount of software we use today.

SEE ALSO
❮ **54–55** Build-your-own computers
❮ **110–111** Translation
❮ **118–119** What do programming languages do?
❮ **120–121** Types of programming language

The C programming language

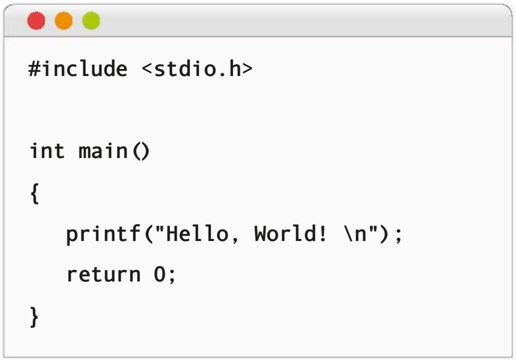
American computer scientist Dennis Ritchie (1941–2011), a programmer at Bell Labs in the USA, released C in 1978. He developed the language while working on the Unix operating system. Unix was coded in assembly language, which tied it to a particular type of computer. This meant that the number of customers willing to buy it were limited. Ritchie created C so that a new version of UNIX could be made that could run on any machine.

▽ IDE
An integrated development environment (IDE) lets programmers write, compile, and run code using a single program with a graphical interface. IDEs make writing large software systems a more manageable process.



```
Claires-MacBook-Air:C claire$ clang -wall hello -o hello
Claires-MacBook-Air:C claire$ ./hello
Hello, World!
```

Command line



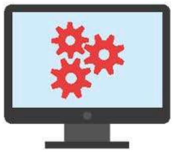
```
#include <stdio.h>


int main()
{
   printf("Hello, World! \n");
   return 0;
}
```

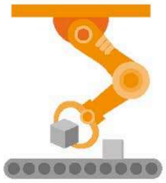
IDE

How does it work?

C is an imperative programming language and doesn't allow object-oriented or functional styles of programming. C's syntax, using curly braces { } to enclose blocks of code, has influenced many other languages. It's a high-level language that doesn't abstract away from the internal structure of the computer. This means programmers can directly access areas in a computer's memory.

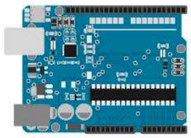


Computer operating system



Robotics

NASA's core flight system



Arduino microprocessor

△ What is it used for?
C's combination of high- and low-level features make it popular for writing operating systems, particularly for the most essential parts. Given its flexibility, it is used in a huge variety of applications.

LINGO

Behind the names

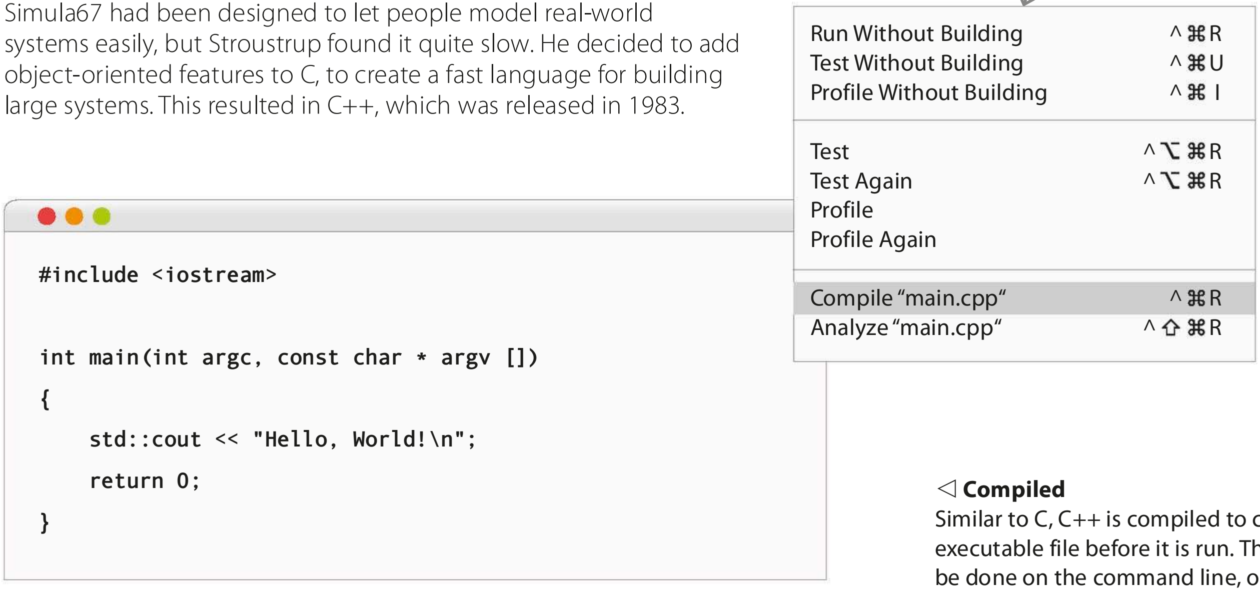
The language Ritchie originally tried to re-implement Unix in was called B, short for BCPL (Basic Computer Programming Language). C was simply the next letter in the alphabet. Putting "++" after a variable in C tells the computer to add one to it, so 1++ = 2. The name C++ reflects the fact that the new language is C, but with additions.

This code sets a variable called "age" to 20, and then by asking for "age++" the number 21 is displayed.

```
int age = 20;
printf("Age is: %d", age);
age++;
printf("Age is now: %d", age);
```

The C++ programming language

In 1979, Danish programmer Bjarne Stroustrup (b. 1950) started working at Bell Labs. He had previously worked using Simula67, considered to be the first object-oriented programming language. Simula67 had been designed to let people model real-world systems easily, but Stroustrup found it quite slow. He decided to add object-oriented features to C, to create a fast language for building large systems. This resulted in C++, which was released in 1983.



```
#include <iostream>

int main(int argc, const char * argv [])
{
    std::cout << "Hello, World!\n";
    return 0;
}
```

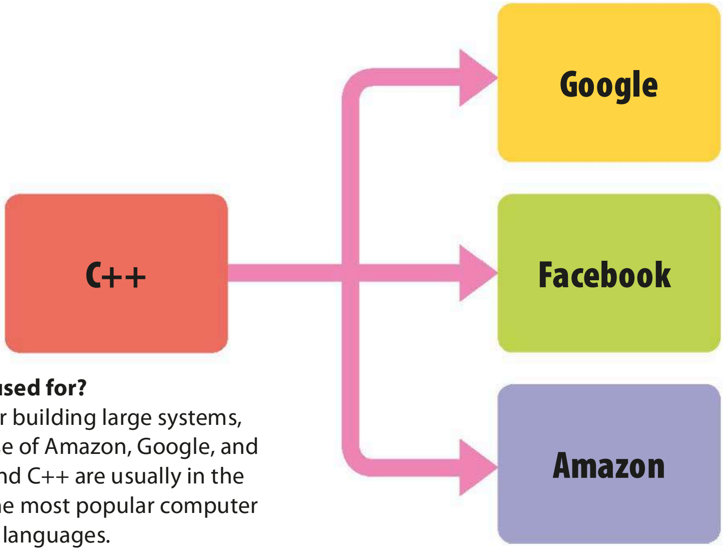
Compiling program for Xcode IDE

C++

◁ **Compiled**
Similar to C, C++ is compiled to create an executable file before it is run. This can be done on the command line, or through an IDE, such as Visual Studio or Xcode.

How does it work?

C++ looks very similar to C. It also allows programmers to access the computer's hardware in the same way. However, unlike C, it includes features that allow programmers to abstract away from the hardware of the computer without slowing down their code. For instance, data structures are ways of organizing data in a program. C++ includes built-in data structures, whereas C programmers have to code these themselves.

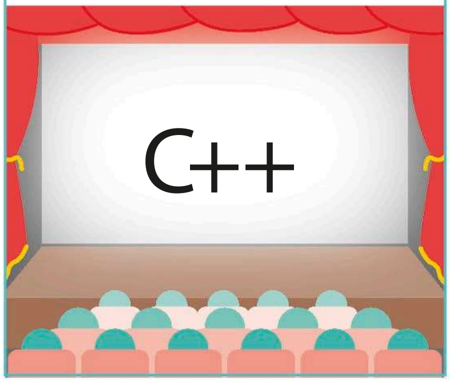


▷ **What is it used for?**
C++ is used for building large systems, including those of Amazon, Google, and Facebook. C and C++ are usually in the top three of the most popular computer programming languages.

REAL WORLD

C++ at the movies

Autodesk's Maya animation tool is written in C++. Maya has been used to create visual effects for many popular films, including *Star Wars Episode I*, *Spider-Man*, *Lord of the Rings*, and several *Harry Potter* movies. It's possible for programmers to write their own plug-ins in C++ to add functionality to Maya.



Java

SEE ALSO

❮ 126–127 C and C++

Python 130–131 ❯

Scratch 136–137 ❯

The Internet of Things 226–227 ❯

Java was developed in 1995 to make it easier to write code for the range of computers available at the time. It is still a major player today.

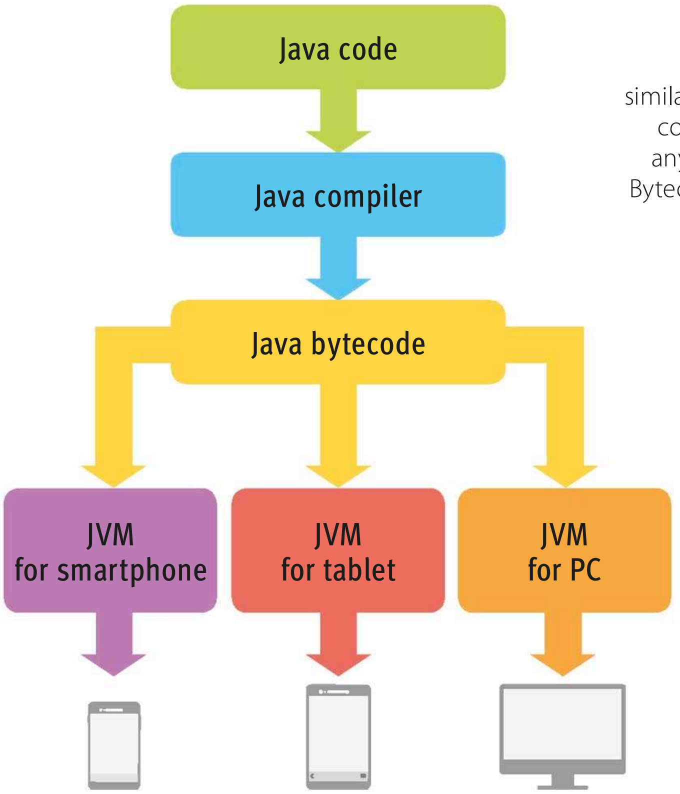
Background

The Java programming language was developed by Canadian computer scientist James Gosling (b. 1955) for the American computer company Sun Microsystem's Java platform – a collection of software designed to allow programmers to develop a variety of systems. These ranged from tiny applications hosted on smartcards for personal banking, to large systems designed for use by many people across an organization. Web browsers soon included the ability to run small self-contained applications, called Java applets, which increased Java's popularity.

▷ **Language of gadgets**
The team behind Java wanted to design a language for programming the increasing number of electronic gadgets available, such as personal digital assistants (handheld personal computers) and webcams.

How does it work?

Java is an object-oriented language. Its syntax was designed to be similar to that of C and C++, but it doesn't include many of their low-level commands. A Java program is designed to behave in the same way on any machine. To enable this, a Java program is compiled into bytecode. Bytecode is machine code for the Java Virtual Machine (JVM), a simulated computer running on the user's real computer.



◁ **Java Virtual Machine**
The Java Virtual Machine is an abstraction that lets programmers write code without worrying about how it will work on a variety of different computers.

REAL WORLD

Bytecode verification

Users can download files containing bytecode and run them on the JVM on their computer, which could allow malicious people to send out bytecode that could cause harm to computers. To avoid this, each bytecode file is examined by the JVM's bytecode verifier, which checks it doesn't perform specific undesirable actions, for example, accessing data to which it shouldn't have access to.

What is Java used for?

Java is used in many systems that people use today, including microblogging social media sites, film-streaming sites, and lots of Android phone apps. Many large banks and airlines use Java to code their systems as it enables them to create and subsequently enlarge systems that carry out large numbers of database operations. Java is possibly the world's most popular programming language.

```
public class Countdown {
    public static void main(String[] args) {
        int count = 10;
        while(count > 0){
            System.out.println(count + "\n");
            count--;
        }
        System.out.print("LIFT OFF! \n");
    }
}
```

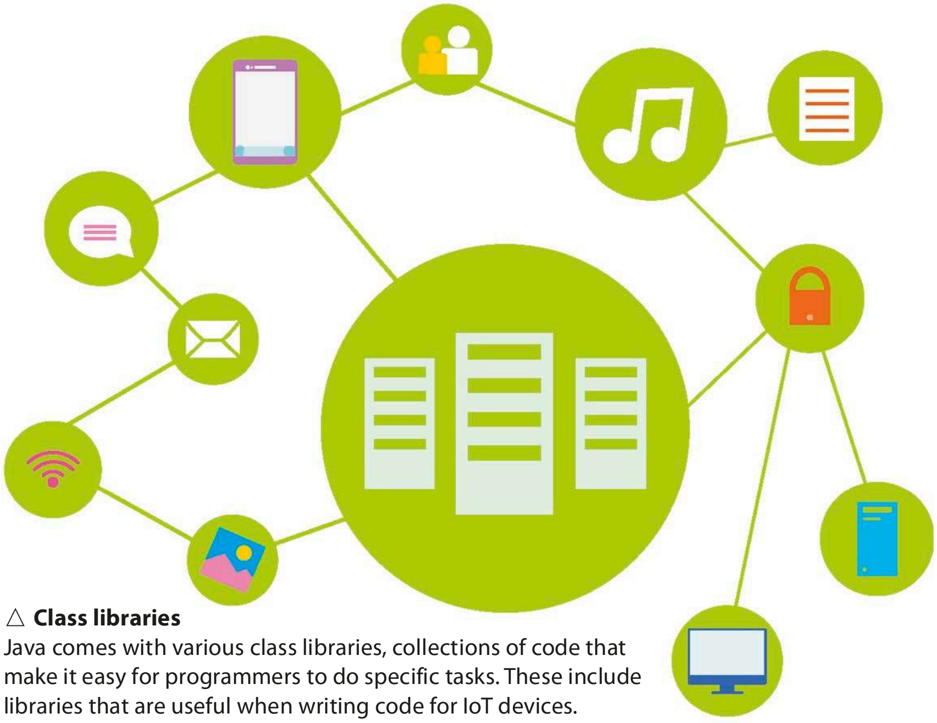
This code prints a countdown from 10 to 0.

When the count reaches 0, "LIFT OFF!" is printed.

▷ **Who can use it?**
Java is very powerful and opens up many possibilities, but it could be confusing for a new coder who may be better off learning Scratch or Python to begin with.

Internet of Things

The Internet of Things (IoT) is the name for the increasing network of objects in the physical world able to connect to the internet. These can include smart appliances such as refrigerators, sensors on farm animals to monitor their health, or thermostats in forests to detect fires. Java has many advantages when it comes to programming these devices, as there is already a version of Java designed for programming small embedded and mobile systems.



△ **Class libraries**
Java comes with various class libraries, collections of code that make it easy for programmers to do specific tasks. These include libraries that are useful when writing code for IoT devices. Programmers using Java have also created libraries for specific devices, many of which are available for other coders to use.

REAL WORLD

Minecraft

The original version of the popular game Minecraft was written in Java. Users were able to write "mods" (short for "modifications") that changed the behaviour of the game world. This was done either by editing the Java source code of the game, or uploading their own Java code. Microsoft recently bought the game and is moving towards a version in C++, but is currently still supporting the Java version.

Python

SEE ALSO
❮ 118–119 What do programming languages do?
❮ 120–121 Types of programming language
Scratch 136–137❯

Released in the 1990s, Python is one of the most popular computer programming languages in the world. It takes a bit longer to learn than Scratch, but can be used to build just about anything.

Python is **named after** the popular **British comedy** series ***Monty Python's Flying Circus***.

Why Python?

Created by Dutch programmer Guido van Rossum, Python is a text-based programming language. It is extremely versatile and can be used to make many different types of program, such as apps, games, and websites. Python is a great language for getting started with computer programming, and is used by many schools and universities to teach coding. Here are its most important features:

Simple and easy to learn

A simple and minimalistic language, Python is extremely beginner-friendly. The code is written in a combination of words, numbers, and punctuation. Its easy syntax allows beginners to focus on learning programming concepts without having to worry about too many details.

Free and open source

Python is an example of a FLOSS (free/libre and open source software), which means that it can be freely distributed, its source code can be read and changed, and its code can be used in new programs. The Python community even encourages people to contribute code, documentation, and resources.

Portable

Python is extremely flexible and can run on a wide variety of hardware platforms and operating systems. Programming languages with these qualities are called “portable”. From Windows to Mac, Linux, PlayStation, and more, Python works everywhere. Its interface looks the same and the programs behave the same way on each platform.

Embeddable

Embeddable with C or C++ encoding, Python allows its users to improve their code with scripting functions. The code can be inserted into an application to provide a programmable interface. It can also be used as a scripting language for building large applications.

Extensive library

Python's greatest strength is its standard library, which supports many common programming tasks, such as connecting to web servers, reading and modifying files, and searching text with regular expressions. It also contains built-in modules that make it easier and quicker to build programs.

Great support

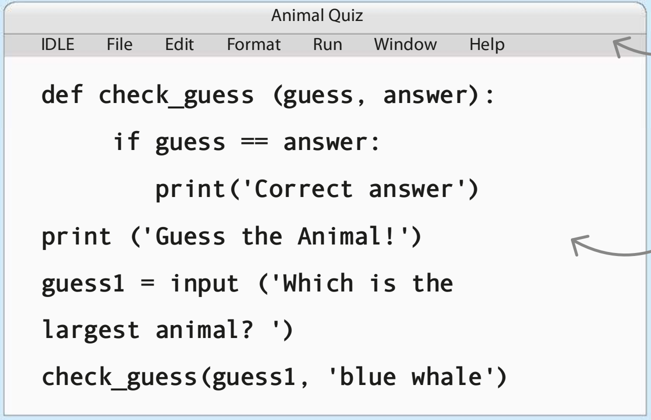
Python provides comprehensive and well-written documentation to its users. It has a guide to getting started, a reference section to explain what things mean, and a lot of example code. Its active support community makes sure Python projects have detailed and easy-to-understand technical documentation.

Working in IDLE

IDLE is a free application that is installed with Python. Designed for beginners, it includes a basic text editor that allows the user to write and edit Python code. It has two different windows – the editor window, which can be used to write and save programs, and the shell window, which runs Python instructions immediately. The shell window gives an immediate response, which makes it ideal for testing and exploring.

The name of the file is shown here.

Editor window

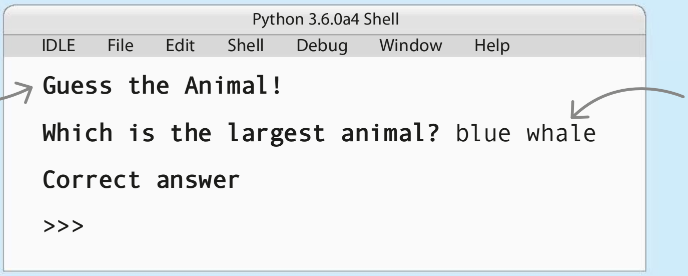


```
def check_guess (guess, answer):
     if guess == answer:
        print('Correct answer')
print ('Guess the Animal!')
guess1 = input ('Which is the
largest animal? ')
check_guess(guess1, 'blue whale')
```

Programs are saved and run from the menu bar.

The code is entered in the editor window.

The output for the program appears in the shell window.



```
Guess the Animal!
Which is the largest animal? blue whale
Correct answer
>>>
```

Users type the answer in the shell window

Shell window

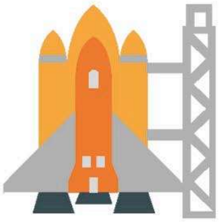
▷ **Testing the code**
IDLE works in three easy steps: write the code, save it, and then run it. This program will ask the user a question and will then check to see if the answer is correct.

Python in action

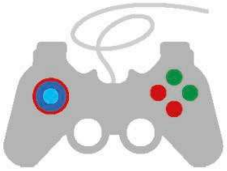
A general-purpose programming language, Python has various applications in the fields of business, medicine, science, and media. It is used to test microchips, power apps, build video games, and write real-world programs.

Business
Python's special libraries and easily readable syntax make it a suitable coding language for customizing larger applications. It can be used by banks to keep track of transactions, and by stores to set prices for their products.

Web development
Widely used on the internet, Python is often used as a support language by software developers and for build control and testing.



Space
Software engineers have used Python to create tools for NASA's Mission Control Centre. These tools help the crew prepare for and monitor the progress of each mission.



Game development
Python has various modules, libraries, and platforms that support computer game development. PySoy is a 3D game engine that supports Python, and PyGame provides functionality and a library for game development.

Scientific computing
Python is used for scientific computing, and it even has some libraries dedicated to specific areas of science. It can also be used to program robots to perform tricky operations.

Ruby

Ruby is a text-based language that offers a great progression when moving on from Scratch. Primarily designed to be programmer-friendly, it has a syntax that's close to English.

SEE ALSO	
❮ 120–121 Types of programming language	
Scratch	136–137 ❯
HTML	162–163 ❯
Cascading Style Sheets	164–165 ❯
Using JavaScript	166–167 ❯

Background

Ruby was released in 1995 by Japanese computer scientist Yukihiro Matsumoto, who wanted to design a simple and general-purpose scripting language. Matsumoto wanted his language to implement all the features necessary to write in an object-oriented style. Ruby was designed to make it easier for programmers to do tasks, rather than making it easier for computers to run code quickly.

```
[irb(main):007:0> puts "Hello, World!"
Hello, World!
=> nil
irb(main):008:0>
```

```
[irb(main):014:0> apples = 3
=> 3
[irb(main):015:0> oranges = 4
=> 4
[irb(main):016:0> fruit = apples + oranges
=> 7
irb(main):018:0>
```

The prompt appears at the start of each line.

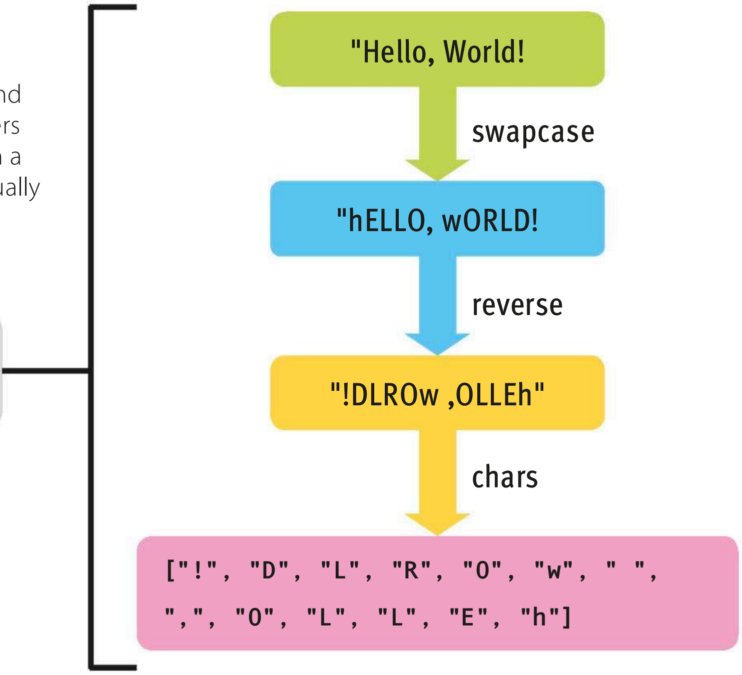
Result of the command just executed

▷ **Interactive Ruby**
Ruby has an interactive interpreter, called the Interactive Ruby Shell or IRB. It allows programmers to type individual commands that can be executed immediately.

How does it work?

Ruby has a very different approach from most languages. Almost everything in it is an object, even items like numbers and characters. These objects have methods that allow programmers to do things with them. In Ruby, it's also possible to program in a mostly imperative style. This allows new programmers to gradually get to grips with the object-oriented style.

```
"Hello, World!".swapcase.reverse.chars
```


"Hello, World!
swapcase
"hELLO, wORLD!
reverse
"!DLROw ,OLLEh"
chars
["!", "D", "L", "R", "O", "w", " ",
",", "O", "L", "L", "E", "h"]

▷ **Chaining**
Ruby aims to be concise. One example of this is chaining, which applies several methods to an object. The method names are separated by dots and applied from left to right.

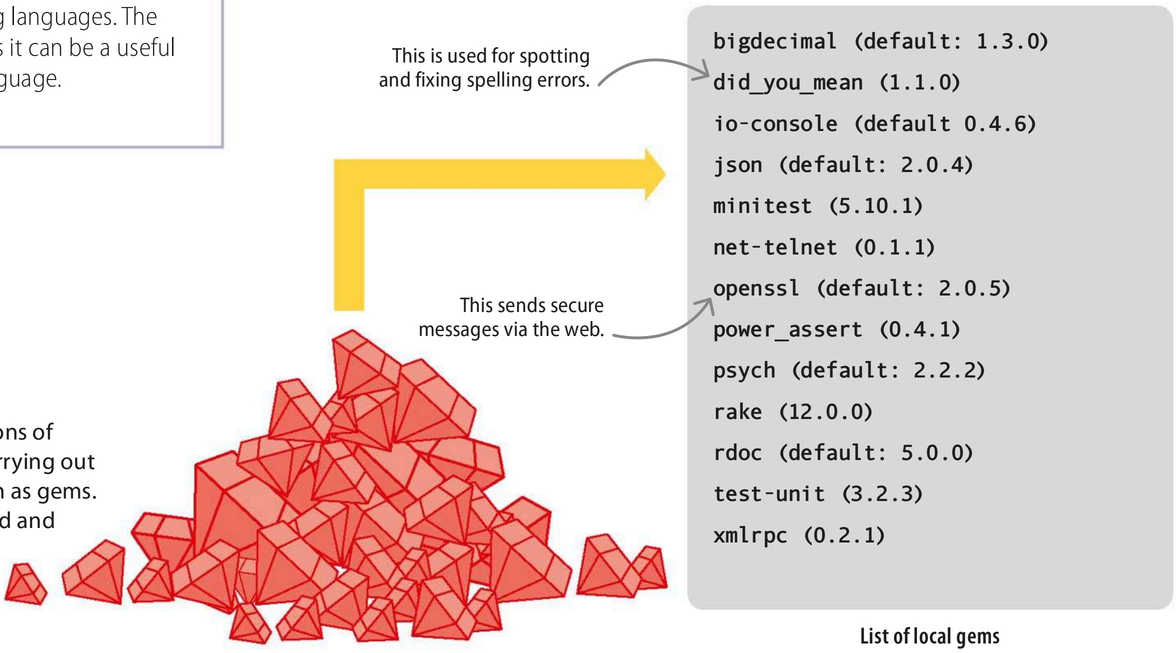
IN DEPTH

REPL

The Ruby interactive interpreter is an example of a REPL, a Read-Evaluate-Print Loop. This is a program that takes one command at a time, runs it, and prints the result. REPLs are a common feature of interpreted languages, which are often referred to as scripting languages. The quick feedback means it can be a useful tool for learning a language.

What is Ruby used for?

Ruby was made popular by "Ruby on Rails", a framework for making websites. It allows programmers to create websites that are connected to a database. The Rails framework simplifies retrieving and displaying data from the database, and allows users to input data through the website. Rails combines Ruby with the languages of web programming: HTML, CSS, and JavaScript.



This is used for spotting and fixing spelling errors.

This sends secure messages via the web.

```
bigdecimal (default: 1.3.0)
did_you_mean (1.1.0)
io-console (default 0.4.6)
json (default: 2.0.4)
minitest (5.10.1)
net-telnet (0.1.1)
openssl (default: 2.0.5)
power_assert (0.4.1)
psych (default: 2.2.2)
rake (12.0.0)
rdoc (default: 5.0.0)
test-unit (3.2.3)
xmlrpc (0.2.1)
```

List of local gems

▷ **RubyGems**
Ruby's libraries, collections of ready-made code for carrying out specific tasks, are known as gems. They can be downloaded and installed using RubyGems, Ruby's built-in tool for gem management.

Why use Ruby?

Ruby was designed to reflect how people think about problems, rather than how computers think about them. All high-level languages abstract away from the way computation is done by a computer's hardware. Ruby is particularly focused on this approach, making it easy for users to program in a variety of styles.

Advantages	Disadvantages
Commands are often closer to a human language than other languages.	Used in fewer areas in the programming world than other languages.
Thanks to Ruby on Rails, Ruby is one of the fastest-growing programming languages.	Ruby code runs slower than compiled languages like Java or C.
Ruby is still being actively developed, keeping pace with new technologies.	There are fewer libraries for coding in areas other than the web.

REAL WORLD

Sonic Pi

A free program, Sonic Pi was built using Ruby. It turns a computer into a musical instrument, which can be played by typing code. Most of its commands are specially created to allow users to do musical tasks, but it uses many of Ruby's basic features as well.

```
use_synth :piano
8.times do
  play :c4
  sleep 0.5
  sample :drum_cowbell
  sleep 0.5
end
```

JavaScript

JavaScript lets programmers create user-friendly interactive webpages. It also allows them to add animations, or change a website's layout when viewed on smartphones.

SEE ALSO	
❮ 128–129 Java	
Malware	156–157 ❯
HTML	162–163 ❯
Cascading Style Sheets	164–165 ❯

Background

In the early 1990s, users couldn't interact with webpages beyond reading them. Websites were often created by amateur programmers who were interested in the new technology, or by designers whose background was in art. JavaScript was designed to enable these users to add interactive elements to their pages. The name "JavaScript" was largely a marketing strategy. Java was very popular and JavaScript's creators hoped the association would be advantageous.

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

Java

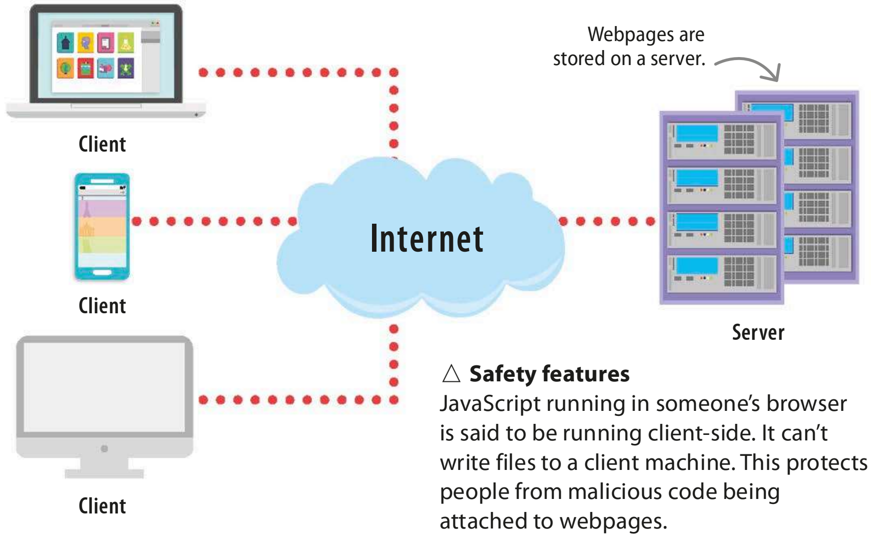
```
function helloworld() {
    alert("Hello, World !");
}
```

JavaScript

▷ **Different from Java**
Though you might expect them to be related, Java and JavaScript are different languages, seen here in their "Hello, World!" code.

How does it work?

A JavaScript program is usually called a script. It's associated with a particular webpage and runs whenever someone loads the page in a browser. JavaScript is interpreted, not compiled, similar to Python. It's predominantly an object-oriented language, but looks quite similar to C as it uses curly brackets and semicolons.

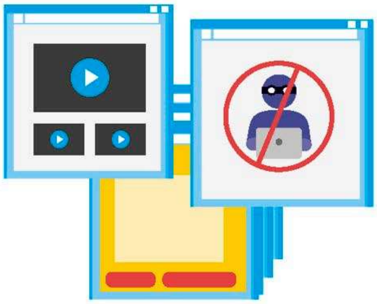


△ **Safety features**
JavaScript running in someone's browser is said to be running client-side. It can't write files to a client machine. This protects people from malicious code being attached to webpages.

REAL WORLD

Pop-ups

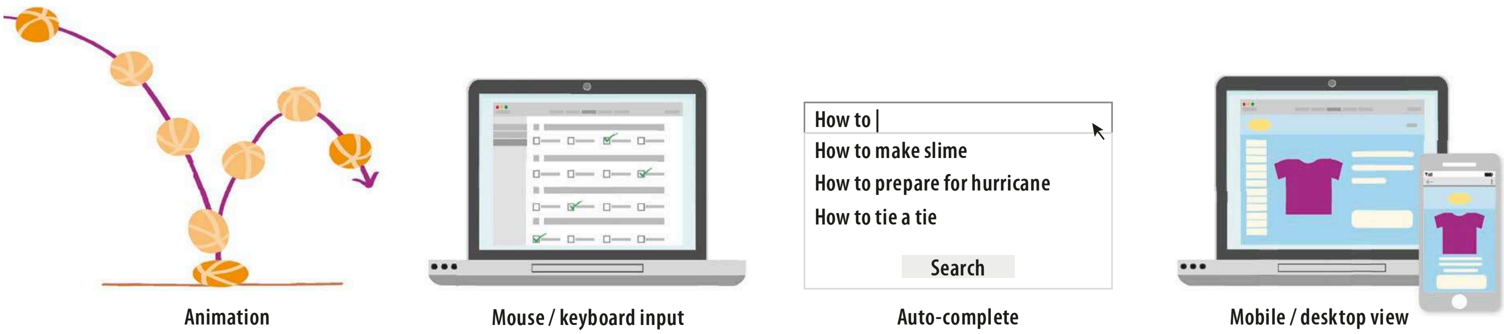
One of JavaScript's least popular applications may be pop-up dialogue boxes. Pop-ups block the browser window and require the user to interact with them. They range from being a mild annoyance that interrupts the browsing experience to redirecting users towards malware and other online scams.



Growth of JavaScript

Professional developers looked down on JavaScript at first, largely because it was designed for and mainly used by amateur programmers. Also, unlike Java applications, JavaScript programs couldn't move data to and from a database on a server. This changed with the introduction of AJAX (Asynchronous JavaScript And XML), a collection of web technologies, including JavaScript, that allowed webpages to connect to a server. Professionals now consider JavaScript to be a useful language, and lots of code libraries have been written to make a variety of tasks easier.

▽ **Using the console**
JavaScript programs are extremely versatile. They allow programmers to create code that will help with multiple aspects of websites, such as animation, user input, auto-complete technology, and enabling the smooth flow of user interface from desktop to mobile websites.



Animation

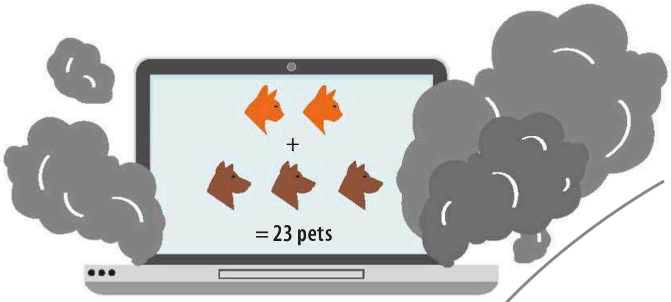
Mouse / keyboard input

Auto-complete

Mobile / desktop view

Why JavaScript?

For programmers new to web development, JavaScript is a great place to start. Using just a text editor and a web browser, programmers can add an array of interactivity and animations to their sites, from the very simple to the quite complex. It's also possible to create apps, known as web apps, using only JavaScript, HTML, and CSS. These apps run on any mobile phone with a browser. There are also websites that help new programmers by allowing them to see the real-time effects of changing their JavaScript, HTML, or CSS scripts.



The "+" symbol can be used to add two numbers or two lines of code. Here, it has added cats and dogs together as a line, and come up with 23 instead of 5, as expected.

```
function pets() {
  var cats = 2;
  var dogs = "3";
  console.log("Number of pets:" + (cats + dogs));
}
```

This code stores numerical values for cats and dogs and then asks to add them together.

△ **Drawbacks**
A variable's type describes whether it's a number, character, or something else. JavaScript doesn't have strict rules about types, which can sometimes lead to unexpected results.

REAL WORLD

JavaScript games

Many browser-based games, including the original version of 2048 – a popular number-puzzle game – are written in JavaScript. Several free JavaScript game engines are also available, which allow programmers to easily create browser-based games. There are even games where the player has to write some JavaScript to complete each level.

A game of 2048

Scratch

Scratch is the first programming language learned by many children. A visual language, it doesn't require users to type code. Instead, programs are made using coloured blocks that represent commands.

SEE ALSO

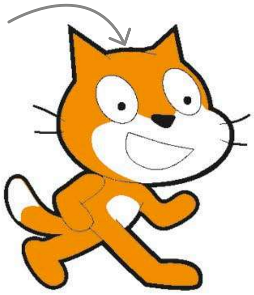
❮ 36–37 Peripheral devices

❮ 118–119 What do programming languages do?

❮ 120–121 Types of programming language

Background

Scratch was created by the Lifelong Kindergarten group at the Massachusetts Institute of Technology (MIT) in the USA. They wanted to make it easier for children aged 8–16 to learn how to code. It emphasizes the creative potential of code, allowing children to create interactive stories, games, art, and more.



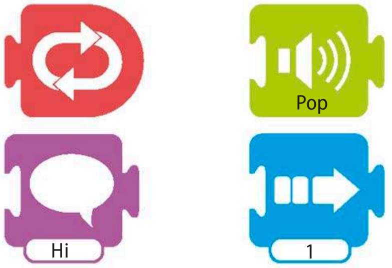
Scratch cat is Scratch's mascot and the default character in projects.

▷ **Scratch worldwide**
Designed to be fun as well as educational, Scratch has a worldwide community of users who share their creations with each other.

TOP TECH

ScratchJr

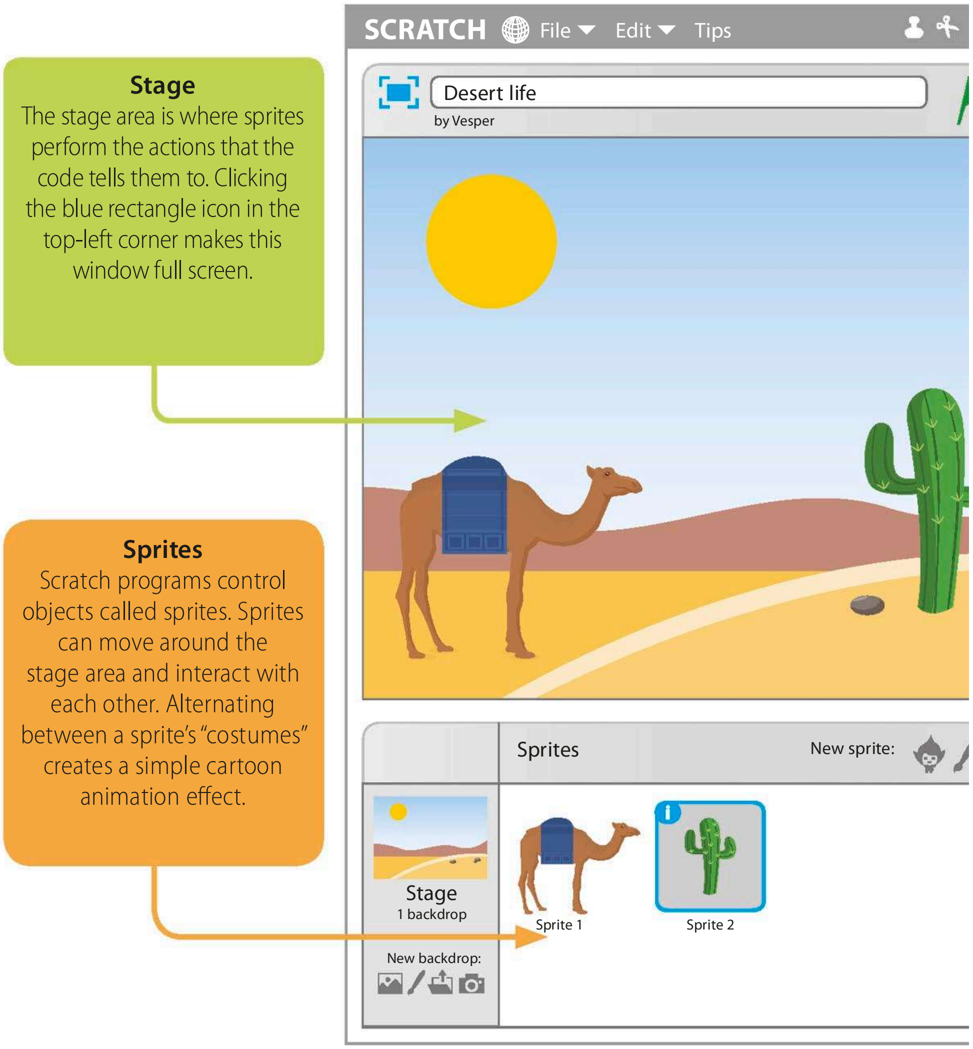
This is a simplified version of Scratch for 5–7 year olds. It allows users to make animations and interactive stories by clicking together coloured blocks representing commands. Blocks mainly feature symbols rather than text and there are fewer than in the standard version. It's available as an app for tablets, rather than a desktop program.



How does it work?

To "write" a program in Scratch, the user drags together coloured blocks that represent instructions. The instructions control images and sounds on the area of the window called the "stage". Scratch doesn't require users to have previous programming skills. The main abilities needed are basic reading, numeracy, and sufficient skill with a computer mouse to drag blocks to the desired locations.

▽ **Scratch 2.0 screen**
A program in Scratch is called a project, and its window is split into several areas, each with its own features.



Stage
The stage area is where sprites perform the actions that the code tells them to. Clicking the blue rectangle icon in the top-left corner makes this window full screen.

Sprites
Scratch programs control objects called sprites. Sprites can move around the stage area and interact with each other. Alternating between a sprite's "costumes" creates a simple cartoon animation effect.

Using Scratch

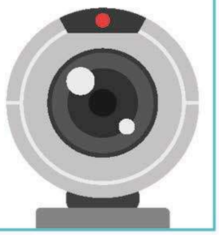
Scratch is an excellent language for new programmers of all ages. It allows them to grasp the basic concepts of programming without the frustration of errors caused by mistyping commands. Scratch's ethos of "remixing" other users' shared code to create new projects also encourages exploration.

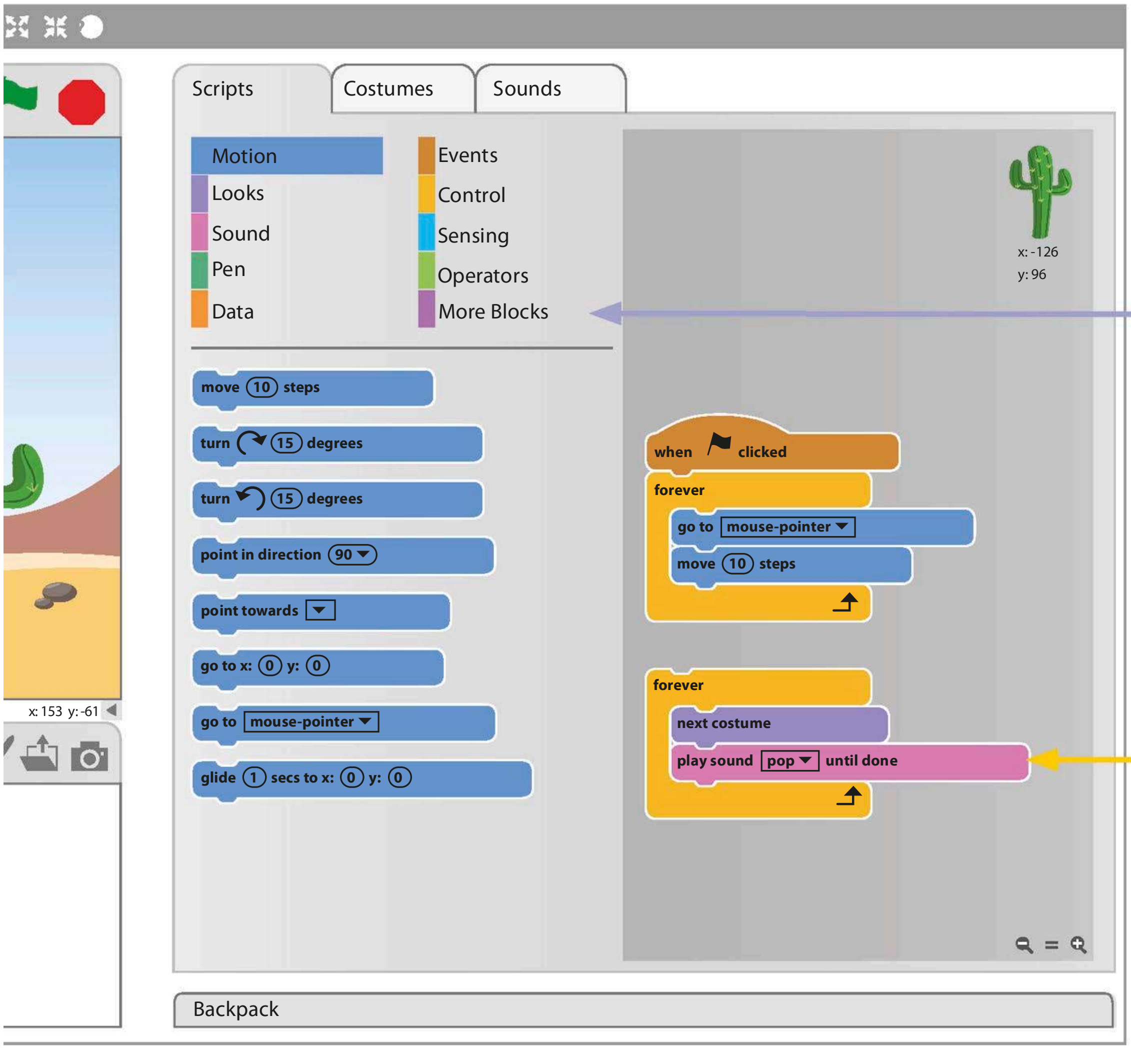
Advantages	Disadvantages
Gives immediate and appealing feedback in the form of animations and sounds	Only supports a limited range of computational concepts
Ability to see and modify other users' shared code is a great aid to learning	Not suitable for more advanced programming, as it doesn't include functions.
Doesn't require typing skills or memorization of commands	Restricts users' ability to write programs that integrate with other systems

REAL WORLD

Controlling code

Scratch allows users to control their programs by moving their body, either by using their computer's webcam or with devices like the Kinect games controller or the Leap motion sensor. This opens up possibilities, such as creating a game that users play without touching a keyboard or another type of controller. They can even create a new musical instrument that's played by dancing.





```
when ⚑ clicked
forever
    go to mouse-pointer
    move 10 steps

forever
    next costume
    play sound pop until done
```

Extending Scratch

There are several extensions to Scratch that allow it to be used beyond its original scope. These include blocks to control motors, LEDs, and more from the Raspberry Pi and ScratchX, a suite of experimental blocks that lets users control robots and other devices.

Blocks

The instruction blocks fit together to make a script that controls a sprite. Blocks doing similar tasks are all one colour; for example, "Sound" blocks are pink, and are labelled with what it does.

Kodu

Kodu is a programming language within a game. It allows players to create their own 3D games on Microsoft's Xbox 360 game console and Windows PCs.

SEE ALSO

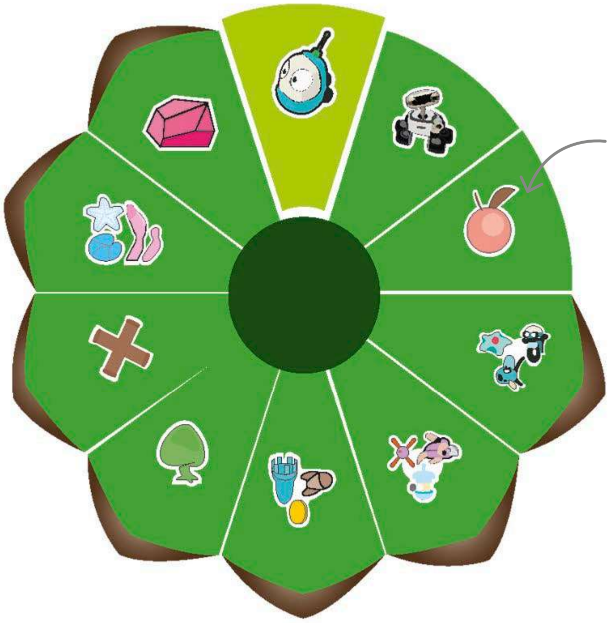
❮ **62–63** Gaming consoles

❮ **118–119** What do programming languages do?

❮ **120–121** Types of programming language

Background

First released in 2009, Kodu is an application for Microsoft's Xbox 360 game console. As children in the 1980s, its developers enjoyed modifying video games by altering their code. Consequently, they felt that modern games deprived children of such opportunities, and decided to make a language that could be used to create a modern 3D video game world. Kodu aims to get kids to see coding as a creative tool for expressing their ideas.

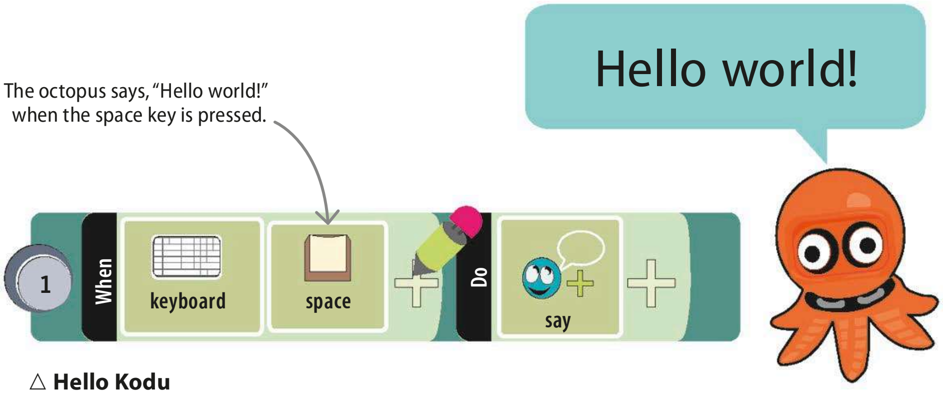


This is one of Kodu's circular visual menus.

▷ **What type of programming?**
Kodu is a visual programming language. It's also object-oriented, as each character or item in the game world is an object with features that can be recognized or changed, and actions that can be done by or to them.

How does it work?

In Kodu, users write programs by creating new rules for the game world. The rules are made up of icons that represent items, actions, and properties, such as colour. The rules determine how the characters in the world react to various situations, and each is in the form, When: <condition> Do: <action>. Icons are selected from circular visual menus using a mouse or game controller.



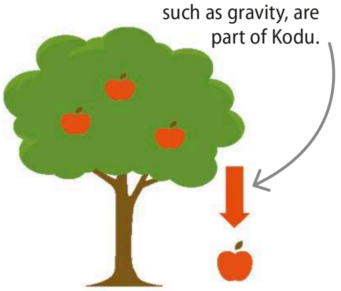
The octopus says, "Hello world!" when the space key is pressed.

△ **Hello Kodu**
Kodu has a "say" command where users can type text that is displayed in speech or thought bubbles next to a character.

IN DEPTH

Built-in physics

Most game programmers have to write a lot of code to create the laws of physics that govern their game world. These laws usually mirror those in the real world. Kodu's developers decided that they would provide a game world with working physics, so that kids using Kodu can work on their own ideas without worrying about technicalities.

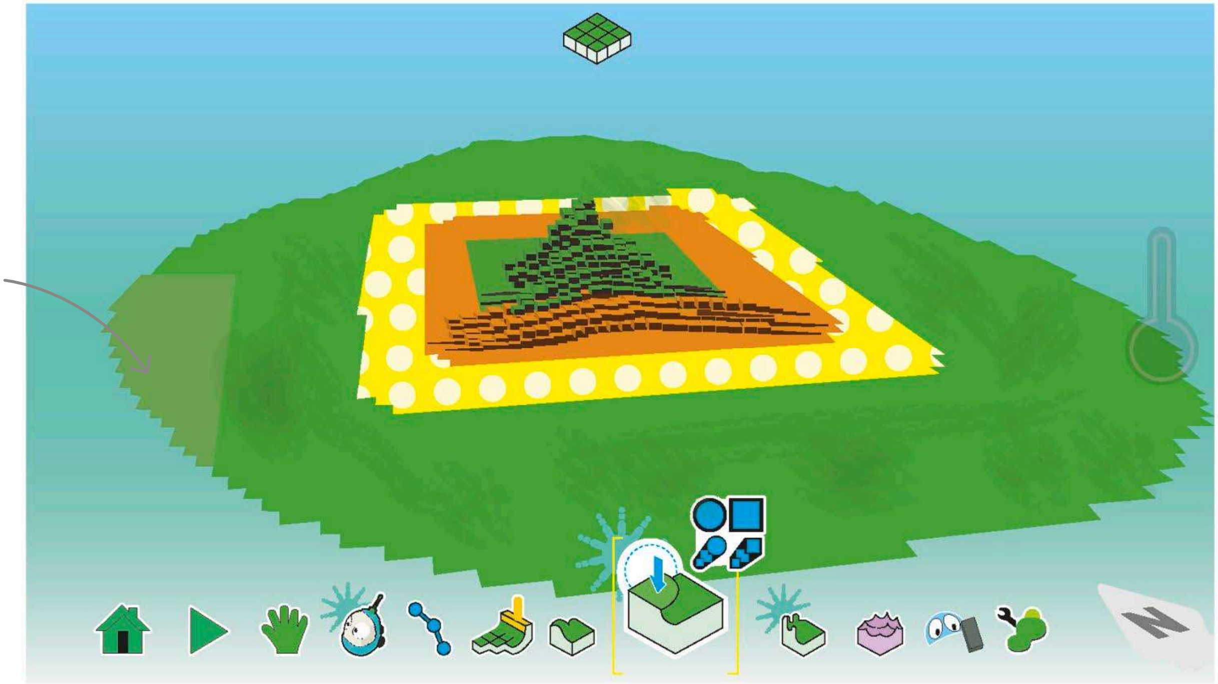


Aspects of physics, such as gravity, are part of Kodu.

Switching pages

Kodu's "switch page" feature allows programmers to make characters behave differently at different points in the game. For instance, at first, bumping a starfish might make it move away – as per a rule on Page 1 of its program. Another rule on Page 1 might make the program switch to Page 2 after 20 seconds. The rule on Page 2 might then tell the starfish to shoot purple missiles when it's bumped.

▽ **Terrain**
Kodu allows users to create the terrain of their world by painting in different types of terrain blocks. It's also possible to add features such as water, hills, and walls.

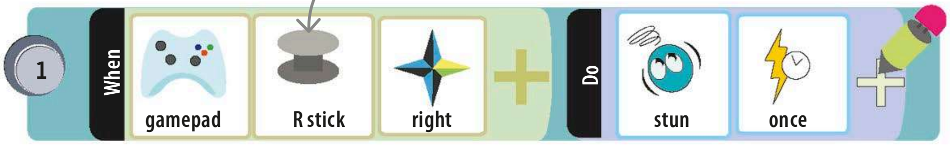


This game world has various different types of terrain.

Why Kodu?

Kodu may be easier for younger children as it's more symbol based than Scratch. It allows children to develop computational thinking skills while creating games. Kodu puts the emphasis on inspiring creativity and making ideas come to life, while enabling kids to build up complex and detailed games. As with Scratch, users can share their games with the Kodu community.

This program wouldn't work on a computer without a joypad.



△ **Limitations**
Being a language within a game, the range of programming that can be done with Kodu is more limited than with other languages. The available options and commands are purely related to gaming.

REAL WORLD

Kodu Kinect

It's possible to control the 3D world of Kodu using body movements or speech. Microsoft's Kinect controller Software Development Kit (SDK) allows programmers to write code that connects the Kinect's input to Kodu. This means that players can control characters with their voice, or make a character jump onscreen by jumping in real life. It does involve a reasonable amount of programming experience, so kids may require assistance from an adult.

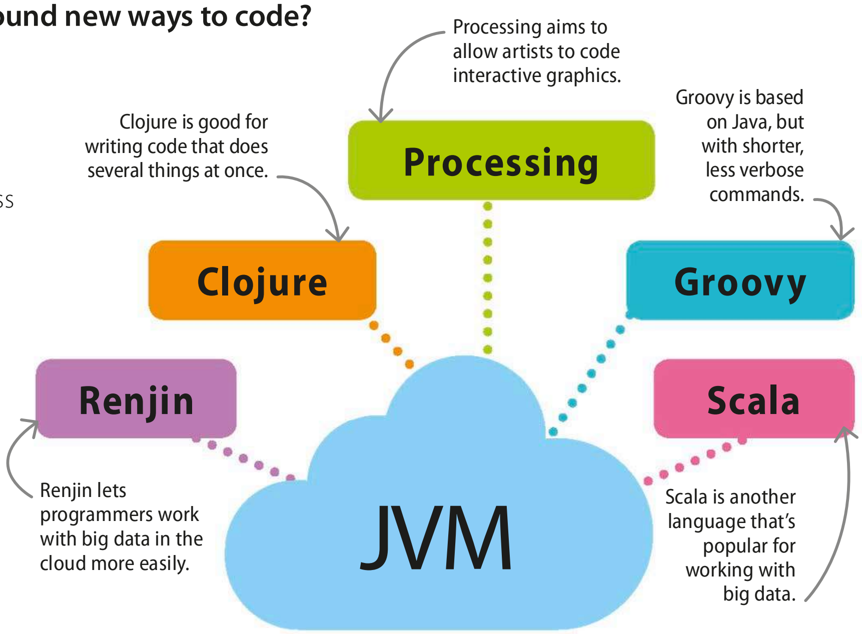
Future languages

SEE ALSO
❮ 80–81 Bits and digitization
❮ 112–113 Assemblers, interpreters, and compilers
❮ 120–121 Types of programming language

Computer-programming languages have come a long way in a relatively short time. In five years' time, will programmers still be using the languages of today or will they have found new ways to code?

Rising stars

A number of programming languages are rapidly increasing in popularity. R is designed for statistical programming and is useful for programs that process a lot of data. Go is very readable and good for networking. It's used by many large organizations. Haskell is a functional language that encourages better programming practices. Rust is based on C, but also includes elements of Haskell. TypeScript is a version of JavaScript with stricter rules, which results in safer code.



Processing aims to allow artists to code interactive graphics.

Clojure is good for writing code that does several things at once.

Groovy is based on Java, but with shorter, less verbose commands.

Renjin lets programmers work with big data in the cloud more easily.

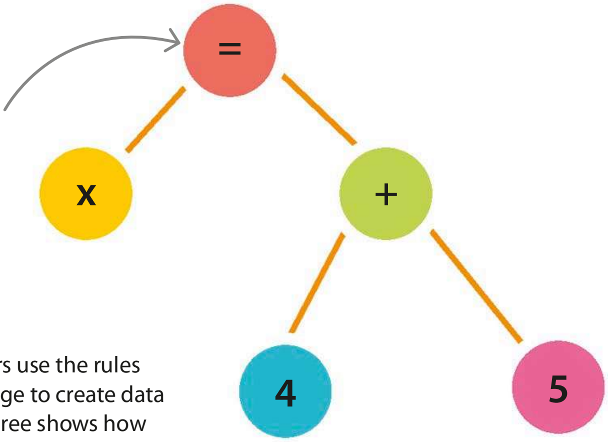
Scala is another language that's popular for working with big data.

▷ **Java Virtual Machine**
Several emerging languages can be run on any computer that has a Java Virtual Machine (JVM) implementation. This is an advantage for the Internet of Things as the JVM can take inputs from devices that run different programming languages.

Creating a language

A number of things have to be considered when creating a new programming language, such as the styles of programming it allows, the existing language it will be written in, and whether the language will be compiled or interpreted. The next step is to create a grammar for the language. This is a set of rules defining how programs can be constructed. Once the grammar is defined, it can be used to write a compiler or interpreter for the language.

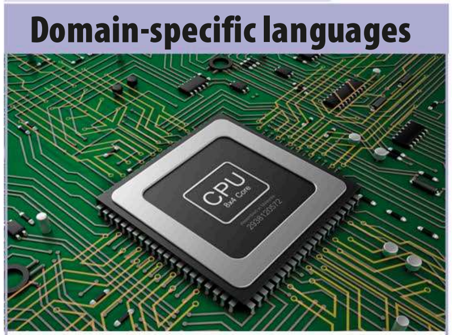
This tree represents the statement x = 4 + 5.



▷ **Trees**
Compilers and interpreters use the rules of a programming language to create data structures called trees. A tree shows how parts of a statement are connected.

IN DEPTH

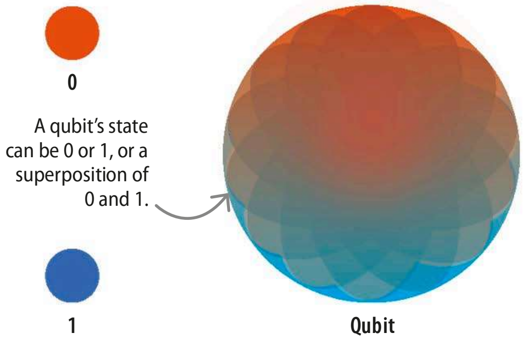
Domain-specific languages



Programmers occasionally create domain-specific languages that are designed for writing programs to solve problems in one specialized area. Some examples include Verilog, used by hardware and computer chip designers, Logo, an early educational language that allowed children to move a turtle-shaped robot around the screen, and SQL, a language used for working with databases.

Future programming languages

What sort of languages would be needed to program entirely new types of computer? Quantum computers use the principles of quantum physics to do calculations, which would take an impossibly long time using normal computers. They're currently in the early stages of development, but Quantum Computing Language (QCL), based on C, has already been created for them.


0
A qubit's state can be 0 or 1, or a superposition of 0 and 1.
1
Qubit

◁ **Qubit**
A bit in a quantum computer is known as a quantum bit, or qubit. It can be in one of three possible states: 0, 1, or a state where it is both 0 and 1 at the same time. The last state is known as a quantum superposition.

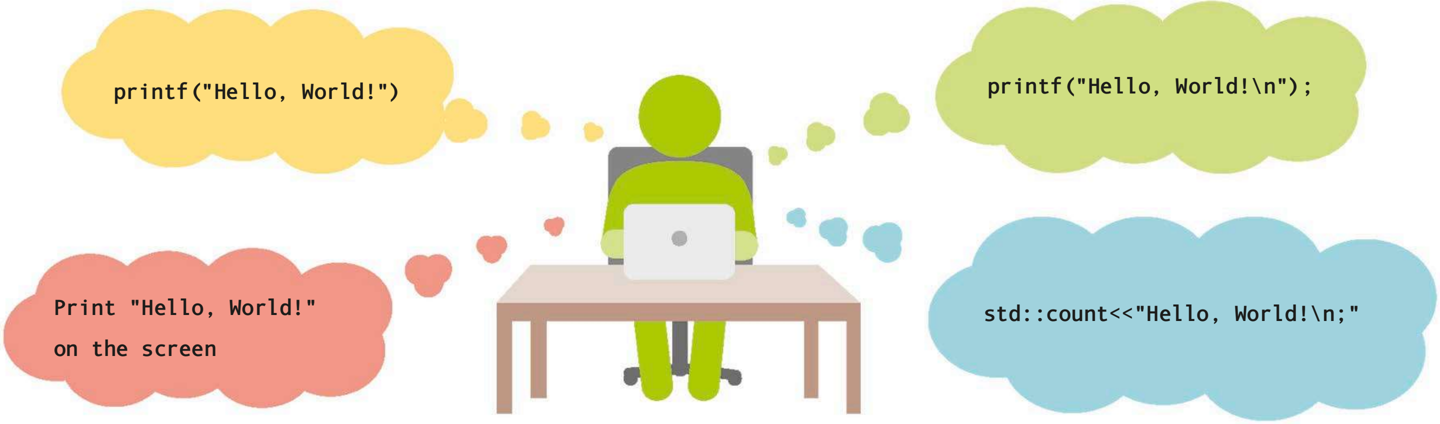
REAL WORLD

Molecular computing

Biological engineers at Massachusetts Institute of Technology (MIT), USA, recently developed a programming language that enabled them to construct biochemical circuits made from DNA. These circuits are placed in biological cells, enabling the cells to react to their environment in specific ways.

A universal language?

It's unlikely that programming languages will move towards a situation where there is one language used for everything. Just like with physical tools, each language is specifically designed to have particular strengths. Machine learning – the ability for computers to learn new things without being specifically programmed – is likely to have an effect on programming in the future, as programmers will make use of tools that have this ability.



```
printf("Hello, World!")
```

```
printf("Hello, World!\n");
```

```
Print "Hello, World!"
on the screen
```

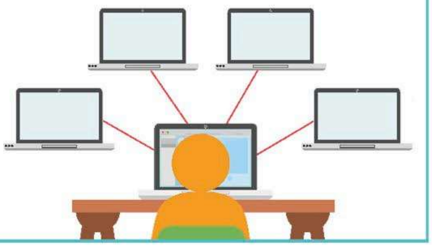
```
std::count<<"Hello, World!\n;"
```

△ **Transferable skills**
Programming involves taking an algorithm and expressing it using instructions a computer can understand. It's a skill that is transferable from one programming language to another. The traditional "Hello, World!" greeting can be written in different programming languages, but have the same result.

REAL WORLD

Machine learning languages

While machine learning may reduce the need for traditional programming skills, the machine learning systems themselves will have to be written by programmers. The most popular languages used to create learning systems include Python, the programming language R, and Java.



8