**Python Essentials 1:  
Module 1**

**Introduction to Python and computer programming**

Introducción a Python y la programación de computadoras

In this module, you will learn about:

*En este módulo, aprenderá sobre:*

* the fundamentals of computer programming, i.e., how the computer works, how the program is executed, how the programming language is defined and constructed;

*• los fundamentos de la programación informática, es decir, cómo funciona la computadora, cómo se ejecuta el programa, cómo se define y construye el lenguaje de programación;*

* the difference between compilation and interpretation;

*• la diferencia entre compilación e interpretación;*

* what Python is, how it is positioned among other programming languages, and what distinguishes the different versions of Python.

*• qué es Python, cómo se posiciona entre otros lenguajes de programación y qué distingue a las diferentes versiones de Python.*

**How does a computer program work?**

*¿Cómo funciona un programa de computadora?*

This course aims to show you what the Python language is and what it is used for. Let's start from the absolute basics.

*Este curso tiene como objetivo mostrarte qué es el lenguaje Python y para qué se utiliza. Comencemos desde lo básico absoluto.*

A program makes a computer usable. Without a program, a computer, even the most powerful one, is nothing more than an object. Similarly, without a player, a piano is nothing more than a wooden box.

*Un programa hace que una computadora sea usable. Sin un programa, una computadora, incluso la más poderosa, no es más que un objeto. Del mismo modo, sin un jugador, un piano no es más que una caja de madera.*

Computers are able to perform very complex tasks, but this ability is not innate. A computer's nature is quite different.

*Las computadoras pueden realizar tareas muy complejas, pero esta habilidad no es innata. La naturaleza de una computadora es bastante diferente.*

It can execute only extremely simple operations. For example, a computer cannot understand the value of a complicated mathematical function by itself, although this isn't beyond the realms of possibility in the near future.

*Solo puede ejecutar operaciones extremadamente simples. Por ejemplo, una computadora no puede comprender el valor de una función matemática complicada por sí misma, aunque esto no está fuera de los límites de la posibilidad en el futuro cercano.*

Contemporary computers can only evaluate the results of very fundamental operations, like adding or dividing, but they can do it very fast, and can repeat these actions virtually any number of times.

*Las computadoras contemporáneas solo pueden evaluar los resultados de operaciones muy fundamentales, como sumar o dividir, pero pueden hacerlo muy rápido y pueden repetir estas acciones prácticamente cualquier cantidad de veces.*

Imagine that you want to know the average speed you've reached during a long journey. You know the distance, you know the time, you need the speed.

*Imagina que quieres saber la velocidad media que has alcanzado durante un largo viaje. Conoces la distancia, conoces el tiempo, necesitas la velocidad.*

Naturally, the computer will be able to compute this, but the computer is not aware of such things as distance, speed, or time. Therefore, it is necessary to instruct the computer to:

*Naturalmente, la computadora podrá calcular esto, pero la computadora no es consciente de cosas como la distancia, la velocidad o el tiempo. Por lo tanto, es necesario instruir a la computadora para que:*

* accept a number representing the distance;
* accept a number representing the travel time;
* divide the former value by the latter and store the result in the memory;
* display the result (representing the average speed) in a readable format.
* *• aceptar un número que represente la distancia;*
* *• aceptar un número que represente el tiempo de viaje;*
* *• dividir el valor anterior por el segundo y almacenar el resultado en la memoria;*
* *• mostrar el resultado (que representa la velocidad promedio) en un formato legible.*

These four simple actions form a **program**. Of course, these examples are not formalized, and they are very far from what the computer can understand, but they are good enough to be translated into a language the computer can accept.

*Estas cuatro acciones simples forman un programa. Por supuesto, estos ejemplos no están formalizados y están muy lejos de lo que la computadora puede entender, pero son lo suficientemente buenos para ser traducidos a un lenguaje que la computadora pueda aceptar.*

*El lenguaje es la palabra clave.*

**Language** is the keyword.

# Natural languages vs. programming languages

*Lenguajes naturales vs lenguajes de programación*

A language is a means (and a tool) for expressing and recording thoughts. There are many languages all around us. Some of them require neither speaking nor writing, such as body language; it's possible to express your deepest feelings very precisely without saying a word.

*Un lenguaje es un medio (y una herramienta) para expresar y registrar pensamientos. Hay muchos idiomas a nuestro alrededor. Algunos de ellos no requieren ni hablar ni escribir, como el lenguaje corporal; es posible expresar tus sentimientos más profundos con mucha precisión sin decir una palabra.*

Another language you use each day is your mother tongue, which you use to manifest your will and to ponder reality. Computers have their own language, too, called **machine**language, which is very rudimentary.

*Otro idioma que usas cada día es tu lengua materna, que usas para manifestar tu voluntad y para ponderar la realidad. Las computadoras también tienen su propio lenguaje, llamado lenguaje de máquina, que es muy rudimentario.*

A computer, even the most technically sophisticated, is devoid of even a trace of intelligence. You could say that it is like a well-trained dog - it responds only to a predetermined set of known commands.

*Una computadora, incluso la más sofisticada técnicamente, está desprovista incluso de un rastro de inteligencia. Se podría decir que es como un perro bien entrenado: responde solo a un conjunto predeterminado de comandos conocidos.*

The commands it recognizes are very simple. We can imagine that the computer responds to orders like "take that number, divide by another and save the result".

*Los comandos que reconoce son muy simples. Podemos imaginar que la computadora responde a órdenes como "toma ese número, divide por otro y guarda el resultado".*

A complete set of known commands is called an **instruction list**, sometimes abbreviated to **IL**. Different types of computers may vary depending on the size of their ILs, and the instructions could be completely different in different models.

*Un conjunto completo de comandos conocidos se denomina lista de instrucciones, a veces abreviada como IL. Los diferentes tipos de computadoras pueden variar según el tamaño de sus IL, y las instrucciones pueden ser completamente diferentes en diferentes modelos.*

Note: machine languages are developed by humans.

*Nota: los lenguajes de máquina son desarrollados por humanos.*

No computer is currently capable of creating a new language. However, that may change soon. Just as people use a number of very different languages, machines have many different languages, too. The difference, though, is that human languages developed naturally.

*Actualmente, ninguna computadora es capaz de crear un nuevo idioma. Sin embargo, eso puede cambiar pronto. Así como las personas usan varios lenguajes muy diferentes, las máquinas también tienen muchos lenguajes diferentes. Sin embargo, la diferencia es que los lenguajes humanos se desarrollaron naturalmente.*

Moreover, they are still evolving, and new words are created every day as old words disappear. These languages are called **natural languages**.

*Además, todavía están evolucionando y cada día se crean nuevas palabras a medida que desaparecen las viejas. Estos lenguajes se llaman lenguajes naturales.*

**What makes a language?**

We can say that each language (machine or natural, it doesn't matter) consists of the following elements:

*Podemos decir que cada lenguaje (máquina o natural, no importa) consta de los siguientes elementos:*

* an **alphabet**: a set of symbols used to build words of a certain language (e.g., the Latin alphabet for English, the Cyrillic alphabet for Russian, Kanji for Japanese, and so on)

*• un alfabeto: un conjunto de símbolos que se utilizan para construir palabras de un determinado idioma (por ejemplo, el alfabeto latino para el inglés, el alfabeto cirílico para el ruso, el kanji para el japonés, etc.)*

* a **lexis**: (aka a dictionary) a set of words the language offers its users (e.g., the word "computer" comes from the English language dictionary, while "cmoptrue" doesn't; the word "chat" is present both in English and French dictionaries, but their meanings are different)
* *• un léxico: (también conocido como diccionario) un conjunto de palabras que el idioma ofrece a sus usuarios (p. ej., la palabra "computadora" proviene del diccionario del idioma inglés, mientras que "cmoptrue" no; la palabra "chat" está presente en los diccionarios de inglés y francés, pero sus significados son diferentes)*
* a **syntax**: a set of rules (formal or informal, written or felt intuitively) used to determine if a certain string of words forms a valid sentence (e.g., "I am a python" is a syntactically correct phrase, while "I a python am" isn't)
* *• una sintaxis: un conjunto de reglas (formales o informales, escritas o sentidas intuitivamente) que se utilizan para determinar si una determinada cadena de palabras forma una oración válida (p. ej., "Soy una pitón" es una frase sintácticamente correcta, mientras que "Soy una pitón" no lo es)*
* **semantics**: a set of rules determining if a certain phrase makes sense (e.g., "I ate a doughnut" makes sense, but "A doughnut ate me" doesn't)
* *• semántica: un conjunto de reglas que determinan si una determinada frase tiene sentido (por ejemplo, "Me comí un donut" tiene sentido, pero "Me comió un donut" no).*

The IL is, in fact, **the alphabet of a machine language**. This is the simplest and most primary set of symbols we can use to give commands to a computer. It's the computer's mother tongue.

*El IL es, de hecho, el alfabeto de un lenguaje de máquina. Este es el conjunto de símbolos más simple y primario que podemos usar para dar comandos a una computadora. Es la lengua materna de la computadora.*

Unfortunately, this mother tongue is a far cry from a human mother tongue. We both (computers and humans) need something else, a common language for computers and humans, or a bridge between the two different worlds.

*Desafortunadamente, esta lengua materna está muy lejos de ser una lengua materna humana. Ambos (computadoras y humanos) necesitamos algo más, un lenguaje común para computadoras y humanos, o un puente entre los dos mundos diferentes.*

We need a language in which humans can write their programs and a language that computers may use to execute the programs, one that is far more complex than machine language and yet far simpler than natural language.

*Necesitamos un lenguaje en el que los humanos puedan escribir sus programas y un lenguaje que las computadoras puedan usar para ejecutar los programas, uno que sea mucho más complejo que el lenguaje de máquina y, sin embargo, mucho más simple que el lenguaje natural.*

Such languages are often called high-level programming languages. They are at least somewhat similar to natural ones in that they use symbols, words and conventions readable to humans. These languages enable humans to express commands to computers that are much more complex than those offered by ILs.

*Dichos lenguajes a menudo se denominan lenguajes de programación de alto nivel. Son al menos algo similares a los naturales en que usan símbolos, palabras y convenciones legibles para los humanos. Estos lenguajes permiten a los humanos expresar comandos a las computadoras que son mucho más complejos que los que ofrecen las IL.*

A program written in a high-level programming language is called a **source code** (in contrast to the machine code executed by computers). Similarly, the file containing the source code is called the **source file**.

*Un programa escrito en un lenguaje de programación de alto nivel se denomina código fuente (en contraste con el código de máquina ejecutado por las computadoras). De manera similar, el archivo que contiene el código fuente se denomina archivo fuente.*

**Compilation vs. interpretation**

*Compilación versus interpretación*

Computer programming is the act of composing the selected programming language's elements in the order that will cause the desired effect. The effect could be different in every specific case – it's up to the programmer's imagination, knowledge and experience.

*La programación de computadoras es el acto de componer los elementos del lenguaje de programación seleccionado en el orden que causará el efecto deseado. El efecto podría ser diferente en cada caso específico: depende de la imaginación, el conocimiento y la experiencia del programador.*

Of course, such a composition has to be correct in many senses:

*Por supuesto, tal composición tiene que ser correcta en muchos sentidos:*

* **alphabetically** – a program needs to be written in a recognizable script, such as Roman, Cyrillic, etc.
* *alfabéticamente: un programa debe estar escrito en un alfabeto reconocible, como romano, cirílico, etc.*
* **lexically** – each programming language has its dictionary and you need to master it; thankfully, it's much simpler and smaller than the dictionary of any natural language;
* *léxicamente: cada lenguaje de programación tiene su diccionario y debes dominarlo; afortunadamente, es mucho más simple y pequeño que el diccionario de cualquier idioma natural;*
* **syntactically** – each language has its rules and they must be obeyed;
* *sintácticamente – cada idioma tiene sus reglas y deben ser obedecidas;*
* **semantically** – the program has to make sense.
* *semánticamente, el programa tiene que tener sentido.*

Unfortunately, a programmer can also make mistakes with each of the above four senses. Each of them can cause the program to become completely useless.

*Desafortunadamente, un programador también puede cometer errores con cada uno de los cuatro sentidos anteriores. Cada uno de ellos puede hacer que el programa se vuelva completamente inútil.*

Let's assume that you've successfully written a program. How do we persuade the computer to execute it? You have to render your program into machine language. Luckily, the translation can be done by a computer itself, making the whole process fast and efficient.

*Supongamos que ha escrito con éxito un programa. ¿Cómo persuadimos a la computadora para que lo ejecute? Tienes que convertir tu programa en lenguaje de máquina. Afortunadamente, la traducción puede ser realizada por una computadora, lo que hace que todo el proceso sea rápido y eficiente.*

There are two different ways of **transforming a program from a high-level programming language into machine language**:

*Hay dos formas diferentes de transformar un programa de un lenguaje de programación de alto nivel a un lenguaje de máquina:*

**COMPILATION** - the source program is translated once (however, this act must be repeated each time you modify the source code) by getting a file (e.g., an .exe file if the code is intended to be run under MS Windows) containing the machine code; now you can distribute the file worldwide; the program that performs this translation is called a compiler or translator;

*COMPILACIÓN: el programa fuente se traduce una vez (sin embargo, este acto debe repetirse cada vez que modifique el código fuente) al obtener un archivo (por ejemplo, un archivo .exe si el código está destinado a ejecutarse en MS Windows) que contiene el código de la máquina; ahora puede distribuir el archivo en todo el mundo; el programa que realiza esta traducción se llama compilador o traductor;*

**INTERPRETATION** - you (or any user of the code) can translate the source program each time it has to be run; the program performing this kind of transformation is called an interpreter, as it interprets the code every time it is intended to be executed; it also means that you cannot just distribute the source code as-is, because the end-user also needs the interpreter to execute it.

*INTERPRETACIÓN: usted (o cualquier usuario del código) puede traducir el programa fuente cada vez que se debe ejecutar; el programa que realiza este tipo de transformación se denomina intérprete, ya que interpreta el código cada vez que se pretende ejecutarlo; también significa que no puede simplemente distribuir el código fuente tal cual, porque el usuario final también necesita el intérprete para ejecutarlo.*

Due to some very fundamental reasons, a particular high-level programming language is designed to fall into one of these two categories.

*Debido a algunas razones muy fundamentales, un lenguaje de programación de alto nivel en particular está diseñado para caer en una de estas dos categorías.*

There are very few languages that can be both compiled and interpreted. Usually, a programming language is projected with this factor in its constructors' minds - will it be compiled or interpreted?

*Hay muy pocos lenguajes que puedan compilarse e interpretarse. Por lo general, un lenguaje de programación se proyecta con este factor en la mente de sus constructores: ¿será compilado o interpretado?*

# What does the interpreter actually do?

Let's assume once more that you have written a program. Now, it exists as a **computer file**: a computer program is actually a piece of text, so the source code is usually placed in **text files**.

*Supongamos una vez más que ha escrito un programa. Ahora, existe como un archivo de computadora: un programa de computadora es en realidad un fragmento de texto, por lo que el código fuente generalmente se coloca en archivos de texto.*

Note: it has to be **pure text**, without any decorations like different fonts, colors, embedded images or other media. Now you have to invoke the interpreter and let it read your source file.

*Nota: tiene que ser texto puro, sin decoraciones como diferentes fuentes, colores, imágenes incrustadas u otros medios. Ahora debe invocar al intérprete y dejar que lea su archivo fuente.*

The interpreter reads the source code in a way that is common in Western culture: from top to bottom and from left to right. There are some exceptions - they'll be covered later in the course.

*El intérprete lee el código fuente de la forma habitual en la cultura occidental: de arriba a abajo y de izquierda a derecha. Hay algunas excepciones; se tratarán más adelante en el curso.*

First of all, the interpreter checks if all subsequent lines are correct (using the four aspects covered earlier).

*En primer lugar, el intérprete comprueba si todas las líneas posteriores son correctas (utilizando los cuatro aspectos tratados anteriormente).*

If the interpreter finds an error, it finishes its work immediately. The only result in this case is an **error message**.

*Si el intérprete encuentra un error, finaliza su trabajo inmediatamente. El único resultado en este caso es un mensaje de error.*

The interpreter will inform you where the error is located and what caused it. However, these messages may be misleading, as the interpreter isn't able to follow your exact intentions, and may detect errors at some distance from their real causes.

*El intérprete le informará dónde se encuentra el error y qué lo causó. Sin embargo, estos mensajes pueden ser engañosos, ya que el intérprete no puede seguir sus intenciones exactas y puede detectar errores a cierta distancia de sus causas reales.*

For example, if you try to use an entity of an unknown name, it will cause an error, but the error will be discovered in the place where it tries to use the entity, not where the new entity's name was introduced.

*Por ejemplo, si intenta usar una entidad con un nombre desconocido, provocará un error, pero el error se descubrirá en el lugar donde intenta usar la entidad, no donde se introdujo el nombre de la nueva entidad.*

In other words, the actual reason is usually located a little earlier in the code, for example, in the place where you had to inform the interpreter that you were going to use the entity of the name.

*En otras palabras, el motivo real suele estar ubicado un poco antes en el código, por ejemplo, en el lugar donde tenías que informar al intérprete que ibas a utilizar la entidad del nombre.*

If the line looks good, the interpreter tries to execute it (note: each line is usually executed separately, so the trio "read-check-execute" can be repeated many times - more times than the actual number of lines in the source file, as some parts of the code may be executed more than once).

*Si la línea se ve bien, el intérprete intenta ejecutarla (nota: cada línea generalmente se ejecuta por separado, por lo que el trío "leer-verificar-ejecutar" se puede repetir muchas veces, más veces que el número real de líneas en el archivo fuente, ya que algunas partes del código pueden ejecutarse más de una vez).*

It is also possible that a significant part of the code may be executed successfully before the interpreter finds an error. This is normal behavior in this execution model.

*También es posible que una parte significativa del código se ejecute con éxito antes de que el intérprete encuentre un error. Este es un comportamiento normal en este modelo de ejecución.*

You may ask now: which is better? The "compiling" model or the "interpreting" model? There is no obvious answer. If there had been, one of these models would have ceased to exist a long time ago. Both of them have their advantages and their disadvantages.

*Usted puede preguntarse ahora: ¿cuál es mejor? ¿El modelo de "compilación" o el modelo de "interpretación"? No hay una respuesta obvia. De haberlo existido, uno de estos modelos habría dejado de existir hace mucho tiempo. Ambos tienen sus ventajas y sus desventajas.*

# Compilation vs. interpretation - advantages and disadvantages

*Compilación versus interpretación - ventajas y desventajas*

|  | **COMPILATION** | **INTERPRETATION** |
| --- | --- | --- |
| **ADVANTAGES** | * the execution of the translated code is usually faster;   *la ejecución del código traducido suele ser más rápida;*   * only the user has to have the compiler - the end-user may use the code without it;   *solo el usuario debe tener el compilador; el usuario final puede usar el código sin él;*   * the translated code is stored using machine language - as it is very hard to understand it, your own inventions and programming tricks are likely to remain your secret.   *el código traducido se almacena usando lenguaje de máquina; como es muy difícil de entender, es probable que sus propios inventos y trucos de programación sigan siendo su secreto.* | * you can run the code as soon as you complete it - there are no additional phases of translation; * the code is stored using programming language, not machine language - this means that it can be run on computers using different machine languages; you don't compile your code separately for each different architecture. |
| **DISADVANTAGES** | * the compilation itself may be a very time-consuming process - you may not be able to run your code immediately after making an amendment;   *la compilación en sí puede ser un proceso que consume mucho tiempo; es posible que no pueda ejecutar su código inmediatamente después de realizar una modificación;*   * you have to have as many compilers as hardware platforms you want your code to be run on. | * don't expect interpretation to ramp up your code to high speed - your code will share the computer's power with the interpreter, so it can't be really fast; * both you and the end user have to have the interpreter to run your code. |

**What does this all mean for you?** *¿Qué significa todo esto para ti?*

* Python is an **interpreted language**. This means that it inherits all the described advantages and disadvantages. Of course, it adds some of its unique features to both sets.
* *Python es un lenguaje interpretado. Esto significa que hereda todas las ventajas y desventajas descritas. Por supuesto, agrega algunas de sus características únicas a ambos conjuntos.*
* If you want to program in Python, you'll need the **Python interpreter**. You won't be able to run your code without it. Fortunately, **Python is free**. This is one of its most important advantages.
* *Si desea programar en Python, necesitará el intérprete de Python. No podrá ejecutar su código sin él. Afortunadamente, Python es gratis. Esta es una de sus ventajas más importantes.*

Due to historical reasons, languages designed to be utilized in the interpretation manner are often called **scripting languages**, while the source programs encoded using them are called **scripts**.

*Debido a razones históricas, los lenguajes diseñados para ser utilizados en la interpretación a menudo se denominan lenguajes de secuencias de comandos, mientras que los programas fuente codificados con ellos se denominan secuencias de comandos.*

**What is Python?**

*¿Qué es Python?*

Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming.

*Python es un lenguaje de programación de alto nivel ampliamente utilizado, interpretado, orientado a objetos y con semántica dinámica, utilizado para la programación de propósito general.*

And while you may know the python as a large snake, the name of the Python programming language comes from an old BBC television comedy sketch series called **Monty Python's Flying Circus**.

*Y si bien es posible que conozca a la pitón como una gran serpiente, el nombre del lenguaje de programación Python proviene de una antigua serie de comedia de televisión de la BBC llamada Monty Python's Flying Circus.*

At the height of its success, the Monty Python team were performing their sketches to live audiences across the world, including at the Hollywood Bowl.

*En el apogeo de su éxito, el equipo de Monty Python estaba realizando sus bocetos para audiencias en vivo en todo el mundo, incluso en el Hollywood Bowl.*

Since Monty Python is considered one of the two fundamental nutrients to a programmer (the other being pizza), Python's creator named the language in honor of the TV show.

*Dado que Monty Python se considera uno de los dos nutrientes fundamentales para un programador (el otro es la pizza), el creador de Python nombró el lenguaje en honor al programa de televisión.*

**Who created Python?**

*¿Quién creó Python?*

One of the amazing features of Python is the fact that it is actually one person's work. Usually, new programming languages are developed and published by large companies employing lots of professionals, and due to copyright rules, it is very hard to name any of the people involved in the project. Python is an exception.

*Una de las características sorprendentes de Python es el hecho de que en realidad es el trabajo de una sola persona. Por lo general, los nuevos lenguajes de programación son desarrollados y publicados por grandes empresas que emplean a muchos profesionales y, debido a las normas de derechos de autor, es muy difícil nombrar a alguna de las personas involucradas en el proyecto. Python es una excepción.*

There are not many languages whose authors are known by name. Python was created by [**Guido van Rossum**](https://en.wikipedia.org/wiki/Guido_van_Rossum), born in 1956 in Haarlem, the Netherlands. Of course, Guido van Rossum did not develop and evolve all the Python components himself.

*No hay muchas lenguas cuyos autores sean conocidos por su nombre. Python fue creado por Guido van Rossum, nacido en 1956 en Haarlem, Países Bajos. Por supuesto, Guido van Rossum no desarrolló ni evolucionó todos los componentes de Python por sí mismo.*

The speed with which Python has spread around the world is a result of the continuous work of thousands (very often anonymous) programmers, testers, users (many of them aren't IT specialists) and enthusiasts, but it must be said that the very first idea (the seed from which Python sprouted) came to one head - Guido's.

*La velocidad con la que Python se ha extendido por todo el mundo es el resultado del trabajo continuo de miles (muchas veces anónimos) programadores, evaluadores, usuarios (muchos de ellos no son especialistas en TI) y entusiastas, pero hay que decir que la primera idea (la semilla de la que brotó Python) surgió en una cabeza: la de Guido.*

# A hobby programming project

*Un proyecto de programación aficionado*

The circumstances in which Python was created are a bit puzzling. According to Guido van Rossum:

*Las circunstancias en las que se creó Python son un poco desconcertantes. Según Guido van Rossum:*

In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (...) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).Guido van Rossum

*En diciembre de 1989, estaba buscando un proyecto de programación de "pasatiempo" que me mantuviera ocupado durante la semana de Navidad. Mi oficina (...) estaría cerrada, pero tenía una computadora en casa y no mucho más en mis manos. Decidí escribir un intérprete para el nuevo lenguaje de secuencias de comandos en el que había estado pensando últimamente: un descendiente de ABC que atraería a los piratas informáticos de Unix/C. Elegí Python como título provisional para el proyecto, ya que estaba de un humor ligeramente irreverente (y era un gran admirador de Flying Circus de Monty Python).*

*guido van rossum*

# Python goals

*Objetivos de Python*

In 1999, Guido van Rossum defined his goals for Python:

*En 1999, Guido van Rossum definió sus objetivos para Python:*

* an **easy and intuitive** language just as powerful as those of the major competitors;
* **open source**, so anyone can contribute to its development;
* code that is as **understandable** as plain English;
* **suitable for everyday tasks**, allowing for short development times.
* *un lenguaje fácil e intuitivo tan potente como el de los grandes competidores;*
* *código abierto, para que cualquiera pueda contribuir a su desarrollo;*
* *código que es tan comprensible como el inglés simple;*
* *adecuado para las tareas cotidianas, lo que permite tiempos de desarrollo cortos.*

About 20 years later, it is clear that all these intentions have been fulfilled. Some sources say that Python is the most popular programming language in the world, while others claim it's the second or the third.

*Unos 20 años después, está claro que todas estas intenciones se han cumplido. Algunas fuentes dicen que Python es el lenguaje de programación más popular del mundo, mientras que otras afirman que es el segundo o el tercero.*

Either way, it still occupies a high rank in the top ten of the [PYPL PopularitY of Programming Language](http://pypl.github.io/PYPL.html) and the [TIOBE Programming Community Index](https://www.tiobe.com/tiobe-index/).

*De cualquier manera, todavía ocupa un alto rango entre los diez primeros de la Popularidad del lenguaje de programación PYPL y el Índice de comunidad de programación TIOBE.*

Python isn't a young language anymore. It is **mature and trustworthy**. It's not a one-hit wonder. It's a bright star in the programming firmament, and time spent learning Python is a very good investment.

*Python ya no es un lenguaje joven. Es maduro y digno de confianza. No es una maravilla de un solo golpe. Es una estrella brillante en el firmamento de la programación, y el tiempo dedicado a aprender Python es una muy buena inversión.*

# What makes Python special?

*¿Qué hace que Python sea especial?*

How does it happen that programmers, young and old, experienced and novice, want to use it? How did it happen that large companies adopted Python and implemented their flagship products using it?

*¿Cómo es que los programadores, jóvenes y viejos, experimentados y novatos, quieren usarlo? ¿Cómo sucedió que las grandes empresas adoptaron Python e implementaron sus productos estrella usándolo?*

There are many reasons – we've listed some of them already, but let's enumerate them again in a more practical manner:

*Hay muchas razones; ya hemos enumerado algunas de ellas, pero vamos a enumerarlas nuevamente de una manera más práctica:*

* it's **easy to learn** – the time needed to learn Python is shorter than for many other languages; this means that it's possible to start the actual programming faster;
* *es fácil de aprender: el tiempo necesario para aprender Python es más corto que para muchos otros idiomas; esto significa que es posible iniciar la programación real más rápido;*
* it's **easy to teach** – the teaching workload is smaller than that needed by other languages; this means that the teacher can put more emphasis on general (language-independent) programming techniques, not wasting energy on exotic tricks, strange exceptions and incomprehensible rules;
* *es fácil de enseñar: la carga de trabajo docente es menor que la que necesitan otros idiomas; esto significa que el profesor puede poner más énfasis en técnicas de programación generales (independientes del lenguaje), sin gastar energía en trucos exóticos, excepciones extrañas y reglas incomprensibles;*
* it's **easy to use** for writing new software – it's often possible to write code faster when using Python;
* *es fácil de usar para escribir software nuevo; a menudo es posible escribir código más rápido cuando se usa Python;*
* it's **easy to understand** - it's also often easier to understand someone else's code faster if it is written in Python;
* *es fácil de entender; a menudo también es más fácil entender el código de otra persona más rápido si está escrito en Python;*
* it's **easy to obtain, install and deploy** – Python is free, open and multiplatform; not all languages can boast that.
* *es fácil de obtener, instalar y desplegar: Python es gratuito, abierto y multiplataforma; no todos los idiomas pueden presumir de eso.*

Of course, Python has its drawbacks, too:

*Por supuesto, Python también tiene sus inconvenientes:*

* it's not a speed demon – Python does not deliver exceptional performance;
* *no es un demonio de la velocidad: Python no ofrece un rendimiento excepcional;*
* *Tiene muchos seguidores, pero hay muchos que prefieren otros lenguajes y ni siquiera consideran Python para sus proyectos*
* in some cases it may be resistant to some simpler testing techniques – this may mean that
* *en algunos casos, puede ser resistente a algunas técnicas de prueba más simples; esto puede significar que la depuración del código de Python puede ser más difícil que con otros lenguajes; afortunadamente, cometer errores también es más difícil en Python.*
* debugging Python code can be more difficult than with other languages; fortunately, making mistakes is also harder in Python.

# Python rivals?

*¿Rivales de Python?*

Python has two direct competitors, with comparable properties and predispositions. These are:

*Python tiene dos competidores directos, con propiedades y predisposiciones comparables. Estos son:*

* **Perl** – a scripting language originally authored by Larry Wall;
* *• Perl: un lenguaje de secuencias de comandos creado originalmente por Larry Wall;*
* **Ruby** – a scripting language originally authored by Yukihiro Matsumoto.
* *• Ruby: un lenguaje de secuencias de comandos creado originalmente por Yukihiro Matsumoto.*

The former is more traditional and more conservative than Python, and resembles some of the old languages derived from the classic C programming language.

*El primero es más tradicional y más conservador que Python, y se parece a algunos de los lenguajes antiguos derivados del lenguaje de programación C clásico.*

In contrast, the latter is more innovative and more full of fresh ideas than Python. Python itself lies somewhere between these two creations.

*Por el contrario, este último es más innovador y está más lleno de ideas frescas que Python. Python mismo se encuentra en algún lugar entre estas dos creaciones.*

The Internet is full of forums with infinite discussions on the superiority of one of these three over the others, should you wish to learn more about each of them.

*Internet está lleno de foros con infinitas discusiones sobre la superioridad de uno de estos tres sobre los demás, si deseas conocer más sobre cada uno de ellos.*

# Where can we see Python in action?

*¿Dónde podemos ver a Python en acción?*

We see it every day and almost everywhere. It's used extensively to implement complex **Internet services** like search engines, cloud storage and tools, social media and so on. Whenever you use any of these services, you are actually very close to Python, although you wouldn't know it.

*Lo vemos todos los días y en casi todas partes. Se utiliza ampliamente para implementar servicios de Internet complejos como motores de búsqueda, herramientas y almacenamiento en la nube, redes sociales, etc. Cada vez que usa cualquiera de estos servicios, en realidad está muy cerca de Python, aunque no lo sepa.*

Many **developing tools** are implemented in Python. More and more **everyday-use applications** are being written in Python. Lots of **scientists** have abandoned expensive proprietary tools and switched to Python. Lots of IT project **testers** have started using Python to carry out repeatable test procedures. The list is long.

*Muchas herramientas de desarrollo se implementan en Python. Cada vez más aplicaciones de uso diario se escriben en Python. Muchos científicos han abandonado costosas herramientas propietarias y se han cambiado a Python. Muchos probadores de proyectos de TI han comenzado a usar Python para llevar a cabo procedimientos de prueba repetibles. La lista es larga.*

# Why not Python?

*¿Por qué no Pitón?*

Despite Python's growing popularity, there are still some niches where Python is absent, or is rarely seen:

*A pesar de la creciente popularidad de Python, todavía hay algunos nichos en los que Python está ausente o rara vez se ve:*

* **low-level programming** (sometimes called "close to metal" programming): if you want to implement an extremely effective driver or graphical engine, you wouldn't use Python;
* *• programación de bajo nivel (a veces llamada programación "cercana al metal"): si desea implementar un controlador o motor gráfico extremadamente efectivo, no usaría Python;*
* **applications for mobile devices**: although this territory is still waiting to be conquered by Python, it will most likely happen someday.
* *• aplicaciones para dispositivos móviles: aunque este territorio aún está esperando a ser conquistado por Python, lo más probable es que algún día suceda.*

# There is more than one Python

# *Hay más de un Python*

There are two main kinds of Python, called Python 2 and Python 3.

*Hay dos tipos principales de Python, llamados Python 2 y Python 3.*

Python 2 is an older version of the original Python. Its development has since been intentionally stalled, although that doesn't mean that there are no updates to it. On the contrary, the updates are issued on a regular basis, but they are not intended to modify the language in any significant way. They rather fix any freshly discovered bugs and security holes. Python 2's development path has reached a dead end already, but Python 2 itself is still very much alive.

*Python 2 es una versión anterior del Python original. Desde entonces, su desarrollo se ha detenido intencionalmente, aunque eso no significa que no haya actualizaciones. Por el contrario, las actualizaciones se publican periódicamente, pero no tienen por objeto modificar el lenguaje de forma significativa. Prefieren arreglar cualquier error recién descubierto y agujeros de seguridad. La ruta de desarrollo de Python 2 ya ha llegado a un callejón sin salida, pero Python 2 en sí mismo todavía está muy vivo.*

**Python 3 is the newer (or to be more precise, the current) version of the language. It's going through its own evolutionary path, creating its own standards and habits.**

*Python 3 es la versión más nueva (o para ser más precisos, la actual) del lenguaje. Está pasando por su propio camino evolutivo, creando sus propios estándares y hábitos.*

These two versions of Python aren't compatible with each other. Python 2 scripts won't run in a Python 3 environment and vice versa, so if you want the old Python 2 code to be run by a Python 3 interpreter, the only possible solution is to rewrite it, not from scratch, of course, as large parts of the code may remain untouched, but you do have to revise all the code to find all possible incompatibilities. Unfortunately, this process cannot be fully automatized.

*Estas dos versiones de Python no son compatibles entre sí. Los scripts de Python 2 no se ejecutarán en un entorno de Python 3 y viceversa, por lo que si desea que un intérprete de Python 3 ejecute el antiguo código de Python 2, la única solución posible es reescribirlo, no desde cero, por supuesto, ya que gran parte del código puede permanecer intacto, pero debe revisar todo el código para encontrar todas las incompatibilidades posibles. Desafortunadamente, este proceso no se puede automatizar por completo.*

It's too hard, too time-consuming, too expensive, and too risky to migrate an old Python 2 application to a new platform, and it's even possible that rewriting the code will introduce new bugs into it. It's easier, and more sensible, to leave these systems alone and to improve the existing interpreter, instead of trying to work inside the already functioning source code.

*Es demasiado difícil, demasiado lento, demasiado costoso y demasiado arriesgado migrar una aplicación antigua de Python 2 a una nueva plataforma, e incluso es posible que reescribir el código introduzca nuevos errores. Es más fácil, y más sensato, dejar estos sistemas solos y mejorar el intérprete existente, en lugar de intentar trabajar dentro del código fuente que ya funciona.*

Python 3 isn't just a better version of Python 2 – it is a completely different language, although it's very similar to its predecessor. When you look at them from a distance, they appear to be the same, but when you look closely, though, you notice a lot of differences.

*Python 3 no es solo una mejor versión de Python 2, es un lenguaje completamente diferente, aunque es muy similar a su predecesor. Cuando los miras desde la distancia, parecen ser iguales, pero cuando los miras de cerca, sin embargo, notas muchas diferencias.*

If you're modifying an old existing Python solution, then it's highly likely that it was coded in Python 2. This is the reason why Python 2 is still in use. There are too many existing Python 2 applications to discard it altogether.

*Si está modificando una antigua solución de Python existente, es muy probable que se haya codificado en Python 2. Esta es la razón por la que Python 2 todavía está en uso. Hay demasiadas aplicaciones Python 2 existentes para descartarlas por completo.*

**NOTE**

If you're going to start a new Python project, **you should use Python 3, and this is the version of Python that will be used during this course.**

*NOTA*

*Si va a comenzar un nuevo proyecto de Python, debe usar Python 3, y esta es la versión de Python que se usará durante este curso.*

It is important to remember that there may be smaller or bigger differences between subsequent Python 3 releases (e.g., Python 3.6 introduced ordered dictionary keys by default under the CPython implementation) – the good news, though, is that all the newer versions of Python 3 are **backward compatible** with the previous versions of Python 3. Whenever meaningful and important, we will always try to highlight those differences in the course.

*Es importante recordar que puede haber diferencias más pequeñas o más grandes entre las versiones posteriores de Python 3 (p. ej., Python 3.6 introdujo claves de diccionario ordenadas de forma predeterminada en la implementación de CPython); sin embargo, la buena noticia es que todas las versiones más nuevas de Python 3 son compatibles con las versiones anteriores de Python 3. Siempre que sea significativo e importante, siempre intentaremos resaltar esas diferencias en el curso.*

All the code samples you will find during the course have been tested against Python 3.4, Python 3.6, Python 3.7, and Python 3.8.

*Todos los ejemplos de código que encontrará durante el curso han sido probados con Python 3.4, Python 3.6, Python 3.7 y Python 3.8.*

**Python aka CPython**

*Python también conocido como CPython*

In addition to Python 2 and Python 3, there is more than one version of each.

*Además de Python 2 y Python 3, hay más de una versión de cada uno.*



First of all, there are the Pythons which are maintained by the people gathered around the PSF ([Python Software Foundation](https://www.python.org/psf-landing/)), a community that aims to develop, improve, expand, and popularize Python and its environment. The PSF's president is Guido von Rossum himself, and for this reason, these Pythons are called **canonical**. They are also considered to be **reference Pythons**, as any other implementation of the language should follow all standards established by the PSF.

*En primer lugar, están los Python que mantienen las personas reunidas en torno a la PSF (Python Software Foundation), una comunidad que tiene como objetivo desarrollar, mejorar, expandir y popularizar Python y su entorno. El presidente del PSF es el mismo Guido von Rossum, y por eso, estos pitones se llaman canónicos. También se consideran Pythons de referencia, ya que cualquier otra implementación del lenguaje debe seguir todos los estándares establecidos por la PSF.*

Guido van Rossum used the "C" programming language to implement the very first version of his language and this decision is still in force. All Pythons coming from the PSF are written in the "C" language. There are many reasons for this approach. One of them (probably the most important) is that thanks to it, Python may be easily ported and migrated to all platforms with the ability to compile and run "C" language programs (virtually all platforms have this feature, which opens up many expansion opportunities for Python).

*Guido van Rossum usó el lenguaje de programación "C" para implementar la primera versión de su lenguaje y esta decisión aún está vigente. Todos los pitones que provienen del PSF están escritos en el lenguaje "C". Hay muchas razones para este enfoque. Uno de ellos (probablemente el más importante) es que gracias a él, Python puede ser portado y migrado fácilmente a todas las plataformas con la capacidad de compilar y ejecutar programas en lenguaje "C" (prácticamente todas las plataformas tienen esta función, lo que abre muchas oportunidades de expansión para Python).*

This is why the PSF implementation is often referred to as **CPython**. This is the most influential Python among all the Pythons in the world.

*Esta es la razón por la cual la implementación de PSF a menudo se denomina CPython. Este es el Python más influyente entre todos los Pythons del mundo.*

**Cython**

Another Python family member is **Cython**.

*Otro miembro de la familia Python es Cython.*



Cython is one of a possible number of solutions to the most painful of Python's traits – the lack of efficiency. Large and complex mathematical calculations may be easily coded in Python (much easier than in "C" or any other traditional language), but the resulting code execution may be extremely time-consuming.

*Cython es una de las posibles soluciones a la característica más dolorosa de Python: la falta de eficiencia. Los cálculos matemáticos grandes y complejos se pueden codificar fácilmente en Python (mucho más fácil que en "C" o cualquier otro lenguaje tradicional), pero la ejecución del código resultante puede llevar mucho tiempo.*

How are these two contradictions reconciled? One solution is to write your mathematical ideas using Python, and when you're absolutely sure that your code is correct and produces valid results, you can translate it into "C". Certainly, "C" will run much faster than pure Python.

*¿Cómo se reconcilian estas dos contradicciones? Una solución es escribir sus ideas matemáticas usando Python, y cuando esté absolutamente seguro de que su código es correcto y produce resultados válidos, puede traducirlo a "C". Ciertamente, "C" se ejecutará mucho más rápido que Python puro.*

This is what Cython is intended to do – to automatically translate the Python code (clean and clear, but not too swift) into "C" code (complicated and talkative, but agile).

*Esto es lo que pretende hacer Cython: traducir automáticamente el código de Python (limpio y claro, pero no demasiado rápido) en código "C" (complicado y comunicativo, pero ágil).*

**Jython**

Another version of Python is called **Jython**.

*Otra versión de Python se llama Jython.*

"J" is for "Java". Imagine a Python written in Java instead of C. This is useful, for example, if you develop large and complex systems written entirely in Java and want to add some Python flexibility to them. The traditional CPython may be difficult to integrate into such an environment, as C and Java live in completely different worlds and don't share many common ideas.

*"J" es para "Java". Imagine un Python escrito en Java en lugar de C. Esto es útil, por ejemplo, si desarrolla sistemas grandes y complejos escritos completamente en Java y desea agregarles algo de flexibilidad de Python. El CPython tradicional puede ser difícil de integrar en un entorno de este tipo, ya que C y Java viven en mundos completamente diferentes y no comparten muchas ideas comunes.*

Jython can communicate with existing Java infrastructure more effectively. This is why some projects find it useful and necessary.

*Jython puede comunicarse con la infraestructura Java existente de manera más efectiva. Es por esto que algunos proyectos lo encuentran útil y necesario.*

Note: the current Jython implementation follows Python 2 standards. There is no Jython conforming to Python 3, so far.

*Nota: la implementación actual de Jython sigue los estándares de Python 2. No hay Jython conforme a Python 3, hasta ahora.*



**PyPy and RPython**

Take a look at the logo below. It's a rebus. Can you solve it?



It's a logo of the **PyPy** - a Python within a Python. In other words, it represents a Python environment written in Python-like language named **RPython** (Restricted Python). It is actually a subset of Python.  
  
The source code of PyPy is not run in the interpretation manner, but is instead translated into the C programming language and then executed separately.

This is useful because if you want to test any new feature that may be (but doesn't have to be) introduced into mainstream Python implementation, it's easier to check it with PyPy than with CPython. This is why PyPy is rather a tool for people developing Python than for the rest of the users.

This doesn't make PyPy any less important or less serious than CPython, of course.

In addition, PyPy is compatible with the Python 3 language.

There are many more different Pythons in the world. You'll find them if you look, but **this course will focus on CPyth**