

ÍNDICE

PREFACIO	IX
CAPÍTULO 1. INTRODUCCIÓN	1
Historia.....	2
Python.....	3
Instalar Python.....	4
Preparar el entorno de desarrollo.....	6
Conociendo el IDE	10
Hola mundo	13
CAPÍTULO 2. VARIABLES, TIPOS DE DATOS Y SINTAXIS.....	19
Variables	19
Tipado	20
Tipos de datos.....	21
Números	21
String.....	23
Boolean	25
Sintaxis	26
Tuplas.....	32

Listas	36
Matrices	46
Diccionarios.....	49
CAPÍTULO 3. OPERADORES EN PYTHON.....	55
Operadores matemáticos en Python	55
Operadores de asignación.....	56
Operadores relacionales	58
Operadores lógicos.	59
CAPÍTULO 4. CONDICIONALES	63
IF	63
IF anidados	67
elif	68
CAPÍTULO 5. BUCLES	71
FOR.....	72
While.....	76
Break, continue, pass	78
CAPÍTULO 6. FUNCIONES.....	83
Introducción.....	83
Funciones con parámetros.....	88
Retorno de valores en funciones	92
Parámetros con valor por defecto	95
Parámetros con nombre	98
Funciones con diferentes cantidad de parámetros.....	100
Palabras reservadas en Python	104

CAPÍTULO 7. PROGRAMACIÓN ORIENTADA A OBJETOS	107
Clases	109
Atributos	110
Métodos.....	111
Crear una clase en Python	113
Variables de instancia en Python	116
Variables de clase.....	119
Método <code>__init__</code>	120
Métodos de instancia.....	121
Métodos de clase	122
Métodos estáticos.....	124
Método <code>__new__</code>	126
Crear un objeto	129
Método Destructor	133
Sobrecarga de métodos	136
Herencia.....	142
Herencia simple	143
Herencia multiple.....	148
Super	151
Polimorfismo.....	153
CAPÍTULO 8. OTROS CONCEPTOS POO O RELACIONADOS.....	161
Clases abstractas.....	161
Interfaces	171
Modificadores de acceso o visibilidad.....	173
name mangling	177

Self	179
Instrucción import.....	180
Sobrescribir métodos	182
CAPÍTULO 9. ITERATORS, ITERABLE, GENERATORS E ITERACIÓN Y OBJETOS ...	187
Iterators	188
Iterable.....	189
Generators	189
iteración e índices	191
Secuencias.....	195
CAPÍTULO 10. DECORATORS.....	199
CAPÍTULO 11. EXPRESIONES REGULARES.....	205
Metacaracteres iteradores.....	207
Metacaracteres delimitadores	208
Utilizar expresiones regulares	208
Sustituir	214
CAPÍTULO 12. EXCEPCIONES.....	219
Try,except	221
Mecanismos para el manejo de excepción	222
Else	226
Finally	226
CAPÍTULO 13. ZEN DE PYTHON	229
ÍNDICE ANALÍTICO	237

PREFACIO

Python es un lenguaje de programación que regularmente despierta la sensación de ser muy “natural” o “simple”, esto se debe a su principal característica, que es la sintaxis sencilla, la facilidad de las reglas con las que cuenta el lenguaje lo hacen muy comprensible al momento de aprenderlo.

Otro de los elementos que hace que Python sea ya un lenguaje reconocido en el desarrollo de software es la gran cantidad de propósitos que este puede llegar a tener, no tiene que ver únicamente con que soporte distintos paradigmas de programación como el estructurado, orientado a objetos, funcional y otros. Se puede utilizar con objetivos tan distintos como realizar un sitio web, automatización de tareas, análisis de datos, microcontroladores y podemos decir que en casi cualquier propósito que pensemos, y aplicarlo en la programación en general.

Python es un lenguaje completo que solo basta con entender su forma trabajar para que el usuario decida cómo implementar el lenguaje ya que en casi cualquier área del desarrollo de software se puede implementar.

Python como lenguaje es moderno e innovador, pero lo sorprendente es que este lenguaje está en el rango de edad de lenguajes como Java, PHP y otros a los que se les ve antiguos y es ahí lo sorprendente de Python, que sin importar la fecha en la que se aprenda nos asombran su diseño y sus capacidades de uso.

Este libro no es un manual de usuario para el lenguaje o la documentación de Python en español, busca compartir con el lector una estructura y un orden para conocer este lenguaje de la mejor forma, además de que iremos resolviendo las

principales dudas que aparecen al empezar a programar y estamos aprendiendo la forma particular de Python de hacer las cosas.

Acerca del autor

El autor de este libro es un apasionado de la tecnología y la docencia. Comenzó en el mundo de la programación con el lenguaje Basic a los 13 años de edad. Colaborador habitual de comunidades en español sobre temas como: desarrollo de videojuegos, programación orientada a objetos, desarrollo web y dispositivos móviles.

José Dimas Luján Castillo nació en 1986, tiene el grado de Maestría en Tecnologías de Información. En la docencia ha colaborado con más de 14 universidades a nivel presencial en Latinoamérica en los niveles de Licenciatura y Maestría. En la educación en línea es colaborador de las plataformas más importantes a nivel mundial con más de 90 cursos en línea en la actualidad; además de ser conferenciante habitual de eventos tecnológicos apoyando siempre la adopción de nuevas tecnologías.

Para que el lector pueda consultar y contactar con el autor, puede localizarlo en redes sociales con el alias `josedlujan`, twitter, Facebook, entre otras. En su web: www.josedlujan.com o por correo electrónico a josedlujan@gmail.com.

Agradecimientos

Este libro es uno de los motivos por el que comencé a escribir libros y documentación técnica de programación, era mi forma de agradecerle a Python por ser un lenguaje que me orientó y enseñó mucho de la programación. Python es un lenguaje que descubrí para varios proyectos y no tenía idea de lo útil que podía ser, repito que esta es mi forma de agradecerse.

Este libro se lo quiero dedicar con mucho cariño y de una manera muy especial a mi esposa Noemí y padres José y Fabiola que siempre me han apoyado durante todo este largo viaje dentro de la tecnología.

1 INTRODUCCIÓN

Durante mucho tiempo Python fue visto como un lenguaje de programación para academias, las matemáticas y la investigación. Con la aparición de “startups” en donde se necesita un tipo de tecnología distinta para el desarrollo de software, más flexible que otros lenguajes de programación de manera general, comienza Python a tomar un nuevo aire como uno de los lenguajes preferidos para la industria del desarrollo de software.

Python ya existía, pero no había dado un salto tan notorio a la industria de software ya que como mencionamos en el párrafo anterior se había mantenido en las universidades, con esto Python empezó a adquirir popularidad y en muchos casos se le consideró como un lenguaje nuevo, aunque ya tenía más de 20 años.

No solamente es su sintaxis lo primero que nos salta a la vista, es que no se colocan los puntos y coma al final de cada línea como en la mayoría de los lenguajes de programación; otra diferencia notable son los espacios que debemos de dejar para ayudarle al intérprete a identificar cuándo acaba o inicia un bloque de código, Python cuenta con una filosofía y aunque podemos decir lo mismo de muchos lenguajes, Python lo tiene muy claro y hasta está redactado como el “Zen de Python” que veremos al final de este libro.

Estas y otras razones son las que no podemos ignorar para mencionar que Python sí es un lenguaje de programación diferente al resto, no solamente en forma sino en

su filosofía. Tanto así que contamos con unos “mandamientos” que debemos seguir si queremos aprovecharlo y sacar el máximo rendimiento.

El camino de Python ha sido largo, pero sobre todo sorpresivo en los últimos años, ya que se adapta al tiempo para por ejemplo ahora ser de los más implementados en temas como Machine Learning, análisis de datos, seguridad informática y otras áreas.

Algo es seguro, si una persona aprende Python, se le abre el camino a distintas áreas de la tecnología en donde irá muy bien acompañado de un lenguaje como este.

Sobre todo podemos decir que Python tiene una curva de aprendizaje muy corta, de hecho se comenta en el mundo académico que Python podría ser nombrado como un lenguaje por defecto para aprender la programación, esto es debido a lo sencillo que es entender su sintaxis y su forma de trabajar los datos, por eso en muchos cursos o carreras universitarias aparece en los primeros semestres en donde se busca facilitar los primeros pasos de cualquier programador con este lenguaje.

Historia

Python es un lenguaje de programación que se inició a principios de los años noventa, su creador es un holandés llamado Guido Van Rossum que se dedicaba a la investigación. Guido Van Rossum simplemente opto como muchos otros creadores de lenguajes de programación por utilizar uno diferente al que ya le habían sugerido usar, aunque en este caso el que él tenía en mente era uno propio. Entonces se decide a crear un lenguaje por los detalles que había observado en un lenguaje de programación que se llama ABC.

Podemos decir que Python comenzó a ver la luz en el año 1991 pero en realidad solo lo primero que se publicó fue la versión 0.9, en ese momento aún faltaban cosas, pero ya se podía sentir la esencia de Python. Hasta el año 1994 es donde en realidad vemos la versión 1.0.

Podemos decir que Python cuenta con 2 versiones de forma general, la 3.x y la 2.x, definiendo X como un número cualquiera como por ejemplo: 2.6, 2.7, etc. Lo mismo pasaría con la versión 3.X. En este caso hay varias justificaciones para esto.

Regularmente el desarrollador novato se pregunta: ¿Para qué tener dos versiones o más de un mismo lenguaje?

En realidad es que la industria funciona de manera diferente a la realidad. Aún existen proyectos muy buenos e interesantes que simplemente no se han migrado a las últimas versiones y por lo tanto dejar de dar el soporte significaría un gran problema para proyectos de este tipo y por lo tanto podría ser un autosabotaje al lenguaje, es por eso que muchos lenguajes no solamente dan soporte a la versión más nueva del mismo, siguen considerando las versiones anteriores mientras esto no genera un problema que ponga en riesgo el crecimiento o avance del lenguaje.

Python

En el momento de escribir este libro, Python se encuentra en la versión 3.7.2, aunque como mencionamos se siguen considerando las versiones 2.x, en algunos casos vamos a ir haciendo comparaciones para evitar futuros problemas o complicaciones durante el proceso de aprender el lenguaje.

Python es un lenguaje interpretado, esto significa que necesita un intérprete que va a leer línea por línea el código que escribimos y ejecutará las instrucciones, así que a diferencia de otros lenguajes nos podemos olvidar de los tiempos de compilación. Además es un lenguaje de alto nivel, es decir, que las instrucciones que tenemos son más simples y que con una instrucción podemos evitar escribir 20 pequeñas instrucciones para realizar la misma tarea.

Podemos sumarle a sus características que es un lenguaje multiplataforma, esto también es una gran fortaleza ya que el 95% de los casos bastará con escribir el código una vez y podemos utilizarlo en cualquier máquina independiente del sistema operativo, esto es algo muy bueno ya que nos permite aprovechar el código, así que si tienes Mac OS, Linux o Windows no es un obstáculo para escribir el código, de hecho algunas versiones de Linux y Mac OS ya vienen con Python por defecto.

En la cuestión de licencias no vamos a entrar en detalle ya que Python ha realizado cambios en su historia respecto a este tema, pero al lector lo que le importa es el tema del costo o pago que tendría que realizar, en este caso decimos de manera muy clara que cualquiera puede trabajar con Python para proyectos personales, proyectos empresariales o investigaciones, no es necesario pagar absolutamente nada.

El nombre del lenguaje es otro de los principales temas cuando estamos comenzando, aunque hoy en día el logo es muy claro que es sobre una pitón, es decir, el nombre del lenguaje pero en español, la realidad es otra. El nombre es un homenaje a la serie de televisión Monty Python, esta serie no es tan popular en América Latina como lo fue en otras partes del mundo, así que el lenguaje de programación tiene este nombre en honor a este grupo de cómicos.

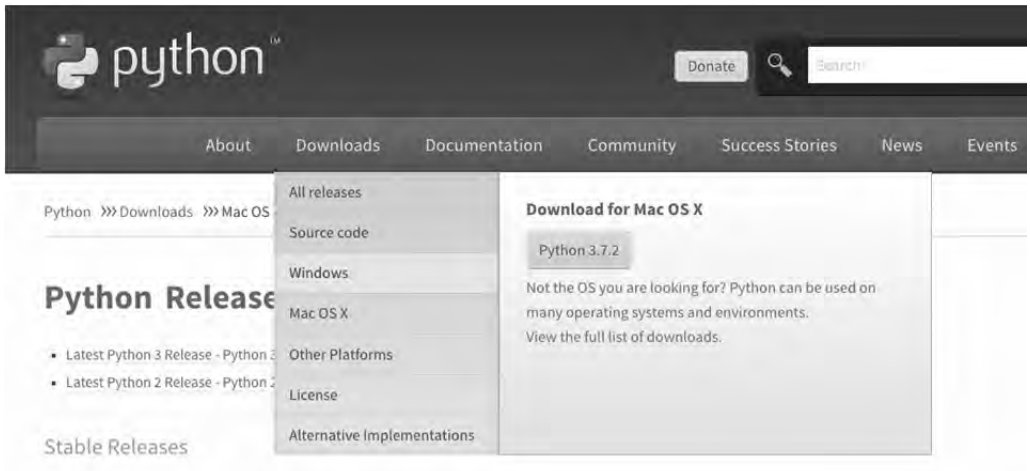
Instalar Python

Como ya se mencionó anteriormente Python ya viene por defecto en los sistemas operativos Mac OS y Linux, pero vamos a ver cómo obtenerlo de manera general en el caso de que el lector sea un usuario Windows o le interese tener la última versión de Python que es con la que vamos a trabajar.

Primero vamos a entrar a la web oficial de Python que es:

www.python.org

Ahora es solamente seleccionar el sistema operativo y la versión de Python que queremos, en principio queremos destacar la última versión como en la siguiente imagen:



Lo importante en este punto es descargar las versiones estables, cualquier versión que en este caso de prelanzamiento o en fase de pruebas puede hacer que el proyecto se cierre de forma inesperada o no funcione del todo bien, por ello debemos cuidar y utilizar versiones ya consideradas como estables, en el momento de escribir este libro son dos:

Python 2.7.16

Python 3.7.2

Algo que podemos mencionar es que es posible tener más de una versión de Python, por ejemplo en mi caso yo cuento con las dos. La versión 2.7.10 y la versión 3.7.2. Esto se puede comprobar con la línea de comandos de esta forma:

```
python --version
```

```
Python 2.7.10
```

```
Python3 --version
```

```
Python 3.7.2
```

Como podemos ver en un caso el comando es solamente Python y en otro el comando incluye el número tres (python3) para hacer la referencia a qué versión de

Python queremos que lo ejecute y así cada versión nos arroja el número que corresponda.

En el caso de Linux podemos aprovechar que ya tenemos Linux por defecto en la mayoría de versiones y ejecutar:

```
sudo apt-get install python3
```

En el caso de versiones basadas en RPM utilizaríamos lo siguiente:

```
yum install python3
```

En resumen podemos decir lo siguiente:

Windows – hacemos la descarga desde la web oficial en la que se nos incluye el IDLE.

Mac OS – ya tenemos la versión 2.x dependiendo de la versión del sistema operativo, algunos cuentan con la versión 2.6 otros con 2.7. Para obtener la versión 3 vamos a ir al sitio oficial y lo instalamos como cualquier aplicación.

Linux – vamos a utilizar la versión 2.x que viene por defecto en nuestra computadora y ejecutamos uno de los dos comandos que ya compartimos dependiendo de si es una versión basada en formato RPM o no.

¡Felicidades! Ahora ya tenemos la última versión de Python en nuestro equipo.

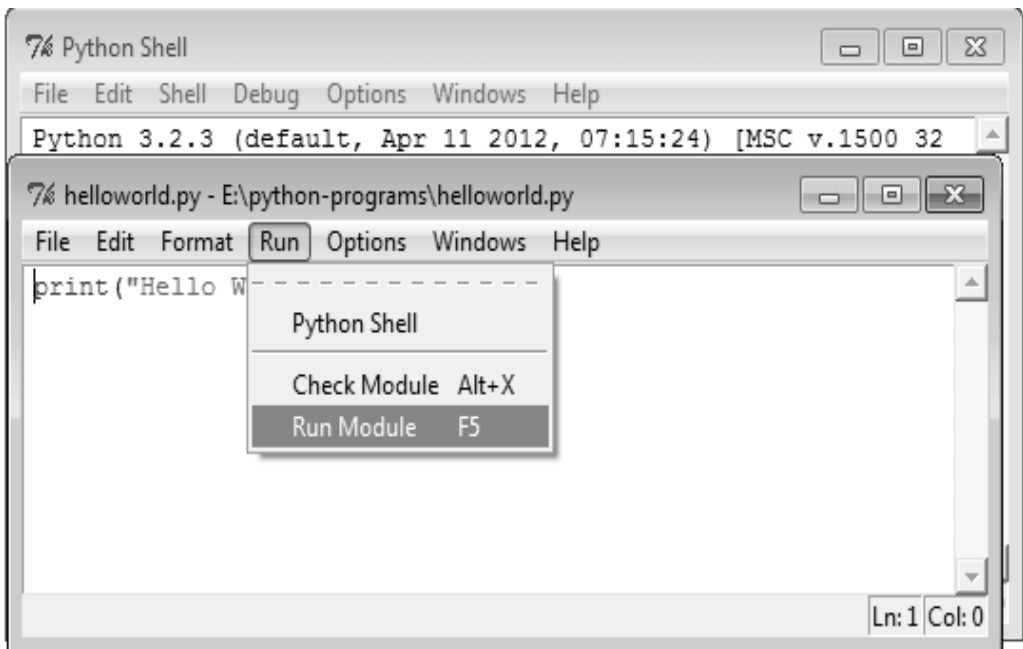
Preparar el entorno de desarrollo

Por defecto cuando descargamos Python desde la web se nos incluye en la mayoría de las versiones el IDLE, este es un entorno de desarrollo integrado que ya viene preparado para trabajar con Python, probablemente no sea la herramienta más sofisticada o avanzada para el desarrollo, pero en cuestión de Python es una herramienta que viene perfectamente integrada para trabajar con el lenguaje, aunque es en algunas ocasiones duramente criticada por su sencillez y en la que

algunas tareas complejas se le van de las manos o porque simplemente no se pueden ejecutar, la realidad es que para algunos desarrolladores novatos les ha hecho mucho más fácil dar sus primeros pasos gracias a lo limpio de la interfaz y lo práctico.

Mencionamos anteriormente que podemos ejecutar Python desde la línea de comandos como ya lo hicimos al verificar su versión, pero en realidad eso no es muy práctico para nuestro caso ya que vamos a ir generando programas de más de una línea y queremos que el lector se sienta cómodo con esto, queremos que pueda leer a detalle el código y que pueda cambiar una variable, valor o cierto elemento para ver qué es lo que está sucediendo.

El IDLE se ve como el bloc de notas y como ya se comentó viene por defecto con la instalación de Python, así que solamente tenemos que ver los nuevos elementos instalados después de la instalación de Python y listo, tendríamos una ventana parecida a esta:



Vamos a compartir una opción que se puede considerar como más avanzada y que es muy fácil de manejar, esta herramienta se conoce como PyCharm.

PyCharm es un IDE, un IDE o también conocido como un Entorno de Desarrollo Integrado son herramientas avanzadas que no solamente son una herramienta para escribir en un lenguaje de programación, estas herramientas dependiendo de las tecnologías que queramos usar ya vienen con otras herramientas que sirven para ser una solución completa.

Por ejemplo, cuando queremos utilizar algo con Java, existen casos en los que distintas versiones del IDE vienen con elementos ya prefabricados, por ejemplo si vas a crear un sitio web Java debemos también de poder escribir, HTML, CSS y en algunos casos hasta necesitamos activar un servidor. En otro caso si queremos utilizar Java para una aplicación de escritorio puede ver ya con algunas librerías o elementos prefabricados, por ejemplo, para usar JavaFX o bases de datos en Java, así que es común que busquemos un IDE dependiendo de nuestras necesidades.

También podemos mencionar que muchos IDE hoy en día cuentan con tiendas como las tiendas de aplicaciones de Android y iOS para descargar lo que se conoce como paquetes o Plugins. Estos son paquetes que agregan funcionalidades al IDE para por ejemplo hacer depuración de algún error o buscar por ejemplo una configuración nueva para combinar tecnologías. En realidad que estos plugins o módulos son cada vez más importantes en el desarrollo y que hacen crecer el repertorio de herramientas.

PyCharm es una herramienta desarrollada por la empresa JetBrains, esta compañía es conocida hoy en día por ser una compañía de gran importancia en el desarrollo de software ya que no solo tiene esta herramienta, sino que cuenta con una gran lista:

- IntelliJ IDEA
- Rider
- PyCham

- PhpStorm
- Clion
- GoLand
- Otras...

Estas son solamente algunas y de las más populares en el mercado, PyCharm como vemos, podemos decir que ya cuenta con un prestigio en el mundo del desarrollo del software.

El sitio para entrar es:

www.jetbrains.com/pycharm/download

En el momento de entrar a la web de PyCharm nos vamos a encontrar con que por defecto detectará el sistema operativo que estamos usando, así que no nos debemos de preocupar por esto, ya que PyCharm no está solamente para un sistema operativo, entonces podemos descargar la versión correspondiente para el sistema operativo que tengamos de manera sencilla.

En el momento de descargar nos vamos a encontrar siempre con 2 versiones:

- Professional
- Community

El tema de la versión tiene que ver con las versiones de pago y gratuita, en este caso la versión “Professional” es la versión de pago y que viene con más opciones habilitadas para utilizar, en el caso de la versión “Community” es la versión gratuita y viene con menos opciones aunque para trabajar en el día a día funciona bastante bien y nos sirve para tomar el curso de manera perfecta.

Vamos a compartir las diferencias entre las versiones, la versión “Professional” viene con estas: