



Lenguaje de Programación Web II

Introducción

Lenguaje de Programación Python



Introducción al lenguaje Python

- Python es un lenguaje de programación de propósito general muy poderoso y flexible, a la vez que sencillo y fácil de aprender.
- En esta lección se busca introducir al lenguaje Python, sus características, modos de instalación, soporte comunitario, y los recursos más destacados disponibles en la Web para tomar en cuenta.

Acerca de Python

- Python es un lenguaje de programación de alto nivel, que permite procesar fácilmente todo tipo de estructuras de datos, tanto numéricos como de texto.



Información personal	
Nacimiento	31 de enero de 1956 (64 años) Países Bajos
Residencia	Belmont (California)
Nacionalidad	Holandesa
Familia	
Cónyuge	Kim Knapp
Hijos	Orlijn Michiel
Educación	
Educado en	Universidad de Ámsterdam
Información profesional	
Ocupación	Científico de la computación
Conocido por	Crear el lenguaje de programación Python
Cargos ocupados	Benevolent Dictator for Life de Python (hasta 2018)
Empleador	Dropbox
Obras notables	•Python •ABC
Distinciones	FSF Award for the Advancement of Free Software en (2001)
Sitio web	Guido van Rossum

Acerca de Python

- Este lenguaje fue creado a principios de los noventa por Guido van Rossum en los Países Bajos.



Acercade Python

- Es relativamente joven (Fortran 1957, Pascal 1970, C 1972, Modula-2 1978, Java 1991).
- Toma características de lenguajes predecesores, incluso, compatibilizando la solución de varios de ellos. Por ejemplo, habilita tres formas de imprimir el valor de una variable: desde el entorno interactivo escribiendo su nombre (como en Basic), usando la función print, con concatenación de elementos (al estilo del write de Pascal) o bien con patrones de formato (al estilo del printf de C).

Open source

- Python se desarrolla bajo una licencia de Open source o código abierto aprobada por OSI, por lo que se puede usar y distribuir libremente, incluso para uso comercial.
- La licencia de Python es administrada por *Python Software Foundation*.



Aplicaciones

- El Python Package Index (PyPI) o en español significa *Índice de paquetes de Python* alberga miles de módulos de terceros para Python.



Aplicaciones

Tanto la biblioteca estándar de Python como los módulos aportados por la comunidad permiten infinitas posibilidades.

- Desarrollo web e Internet.
- Acceso a la base de datos.
- GUIs de escritorio.
- Científico y numérico.
- Educación.
- Programación de red.
- Desarrollo de Software y Juegos.



Desarrollo Web e Internet

Python ofrece muchas opciones para el desarrollo web :

- Frameworks como Django y Pyramid .
- Microestructuras como Flask y Bottle .
- Sistemas avanzados de gestión de contenido como Plone y django CMS .

La biblioteca estándar de Python admite muchos protocolos de Internet:

- HTML y XML
- JSON
- Procesamiento de correo electrónico .
- Soporte para FTP , IMAP y otros protocolos de Internet .
- Interfaz de enchufe fácil de usar .

Científico y numérico

Python se usa ampliamente en computación científica y numérica :

- SciPy es una colección de paquetes para matemáticas, ciencias e ingeniería.
- Pandas es una biblioteca de análisis y modelado de datos.
- IPython es un potente shell interactivo que permite editar y grabar fácilmente una sesión de trabajo y admite visualizaciones y computación paralela.
- El curso de carpintería de software enseña habilidades básicas para la informática científica, ejecutando campamentos de entrenamiento y proporcionando materiales de enseñanza de acceso abierto.

Educación

- Python es un excelente lenguaje para enseñar programación, tanto a nivel introductorio como en cursos más avanzados.
- Libros como How to Think Like a Computer Scientist , Python Programming: An Introduction to Computer Science y Practical Programming .
- El Grupo de Interés Especial en Educación es un buen lugar para discutir temas de enseñanza.

GUI de escritorio

La biblioteca Tk GUI se incluye con la mayoría de las distribuciones binarias de Python.

Algunos kits de herramientas que se pueden utilizar en varias plataformas están disponibles por separado:

- wxWidgets
- Kivy , para escribir aplicaciones multitáctiles.
- Qt a través de pyqt o pyside

Los kits de herramientas específicos de la plataforma también están disponibles:

- GTK +
- Microsoft Foundation Classes a través de las extensiones win32

Desarrollo de software

- Python se usa a menudo como lenguaje de soporte para desarrolladores de software, para control y administración de compilaciones, pruebas y de muchas otras formas.
- SCons para el control de la construcción.
- Buildbot y Apache Gump para compilación y pruebas continuas automatizadas.
- Roundup o Trac para seguimiento de errores y gestión de proyectos.

Aplicaciones de negocios

- Python también se utiliza para crear sistemas ERP y de comercio electrónico:
- **Odoo** es un software de administración todo en uno que ofrece una gama de aplicaciones comerciales que forman un conjunto completo de aplicaciones de administración empresarial.
- **Tryton** es una plataforma de aplicaciones de uso general de alto nivel de tres niveles.

Comunidad

- El gran software es soportado por grandes personas. La base de usuarios es entusiasta, dedicada a fomentar el uso del lenguaje y comprometida a que sea diversa y amigable.
<https://www.python.org/community/diversity/>



Listas de correo

- Existen listas de correo de Python y grupos de noticias como recursos de la comunidad. Estos recursos están disponibles públicamente de python.org, y son usando como un canal de información o discusión sobre ideas nuevas, e incluso históricamente como otra vía soporte en línea.
- Además hay la guía de recursos de Python que no están en inglés, que incluye listas de correo, documentación traducida y original que no está en inglés, y otros recursos. <https://www.python.org/community/lists/>

Foros

Existe algunos recursos disponibles en formato de Foros, a continuación se listan:

- [Python Forum \(English\)](#).
- [Python-Forum.de \(German\)](#).
- [/r/learnpython \(English\)](#).

Si usted esta buscando un foro nativo en su idioma, por favor, consulte la pagina de los grupos locales en la Wiki de Python.

Comunidades locales

Comunidad(es) de Python en Perú

- Comunidad de estudiantes, profesionales e interesados en tener un espacio donde conversar, preguntar e intercambiar ideas todo acerca del lenguaje de programación Python. Más información visite <https://www.meetup.com/es/pythonperu/>

Características

Las características del lenguaje de programación Python se resumen a continuación:

- Es un lenguaje interpretado, no compilado, usa tipado dinámico, fuertemente tipado.
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.
- En Python, el formato del código (p. ej., la indentación) es estructural.

Fuertemente tipado

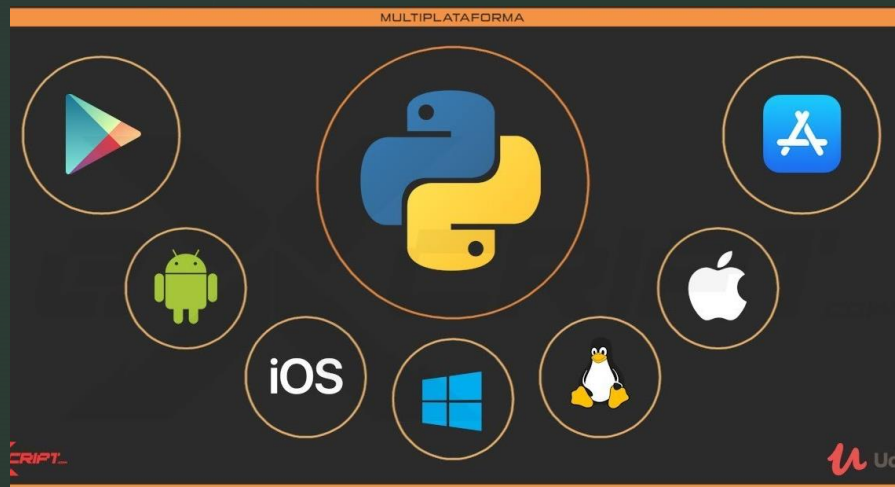
- El fuertemente tipado significa que el tipo de valor no cambia repentinamente. Un string que contiene solo dígitos no se convierte mágicamente en un número. Cada cambio de tipo requiere una conversión explícita. A continuación un ejemplo de este concepto:

```
# variable "valor1" es integer, variable "valor2" es string  
valor1, valor2 = 2, "5"  
# el metodo int() es para convertir a integer  
total = valor1 + int(valor2)  
# el metodo str() es para convertir a string  
print "El total es: " + str(total)
```

Tipado dinámico

- El tipado dinámico significa que los objetos en tiempo de ejecución (valores) tienen un tipo, a diferencia del tipado estático donde las variables tienen un tipo. A continuación un ejemplo de este concepto:

```
# "variable" guarda un valor integer
variable = 11
print variable, type(variable)
# "variable" guarda un valor string
variable = "activo"
print (variable), type(variable)
```



Multiplataforma

Multiplataforma

Python es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos, eso quiere decir, soporta las siguientes plataformas para su ejecución:

- Versiones Python para Microsoft Windows (y DOS) (arquitectura x86/x86-64 en presentación de ejecutable, archivo Zip, instalador basado en la Web).
<https://www.python.org/downloads/windows/>
- Video para mayor información: *Instalando Python en Windows.*

Multiplataforma

- Versiones Python para Mac OSX (Macintosh) (arquitectura 32bit/64bit en presentación de instalador ejecutable).
<https://www.python.org/downloads/mac-osx/>
- Video para mayor información *Instalando Python en una Mac.*

Multiplataforma

- Versiones Python en código fuente (archivo tarball del código fuente comprimido con XZ y con Gz). Para la mayoría de los sistemas Linux/UNIX, usted debe descargar y compilar el código fuente. <https://www.python.org/downloads/source/>
- Video para mayor información *Instalando Python en un Linux.*

Multiplataforma

Versiones de Implementaciones Alternativas Python, la versión “tradicional” de Python (tiene nombre código CPython). Existen un número de implementaciones alternativas que están disponibles a continuación:

- IronPython, Python ejecutando en .NET.
- Jython, Python ejecutando en el Java Virtual Machine.
- PyPy, Una rápida implementación de python con un compilador JIT.
- Stackless Python, Una rama del desarrollo del CPython que soporta microthreads.
- MicroPython, Python ejecutando en micro controladores.

Multiplataforma

Versiones de Python en otras plataformas, la versión “tradicional” de Python (tiene nombre código CPython), mas esta versión ha sido migrada a un número plataformas especializadas y/o antiguas, a continuación se destacan algunas de ellas.

- Pythonista, Python para iOS, ofrece un completo entorno de desarrollo para escribir scripts Python en su iPad o iPhone.
- ActivePython, Python para Solaris, Usted puede comprarlo (versiones comerciales y comunitarias, incluidos los módulos de computación científica, no de código abierto), o compilar desde una fuente si tiene un compilador de C. Los paquetes UNIX tienen una variedad de versiones de Python para una variedad de versiones de Solaris. Estos utilizan el estándar Sun pkgadd.

Filosofía “Incluye baterías”

- Python ha mantenido durante mucho tiempo esta filosofía de “baterías incluidas”:

“Tener una biblioteca estándar rica y versátil que está disponible de inmediato. Sin que el usuario descargue paquetes separados.”

- Esto le da al lenguaje una ventaja en muchos proyectos.
- Las “baterías incluidas” están en la librería estándar Python.
https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion10/libreria_estandar.html#python-libreria-estandar

Zen de Python

- Es una colección de 20 principios de software que influyen en el diseño del Lenguaje de Programación Python, de los cuales 19 fueron escritos por Tim Peters en junio de 1999. El texto es distribuido como dominio público.
- El Zen de Python está escrito como la entrada informativa número 20 de las propuestas de mejoras de Python (Python Enhancement Proposals - PEP), y se puede encontrar en el sitio oficial de Python.
- También se incluye como un *huevo de pascua*, el cual se puede encontrar, desde el intérprete de Python, ingresar la siguiente sentencia:

```
>>> import this
```

Zen de Python

Los principios están listados a continuación:

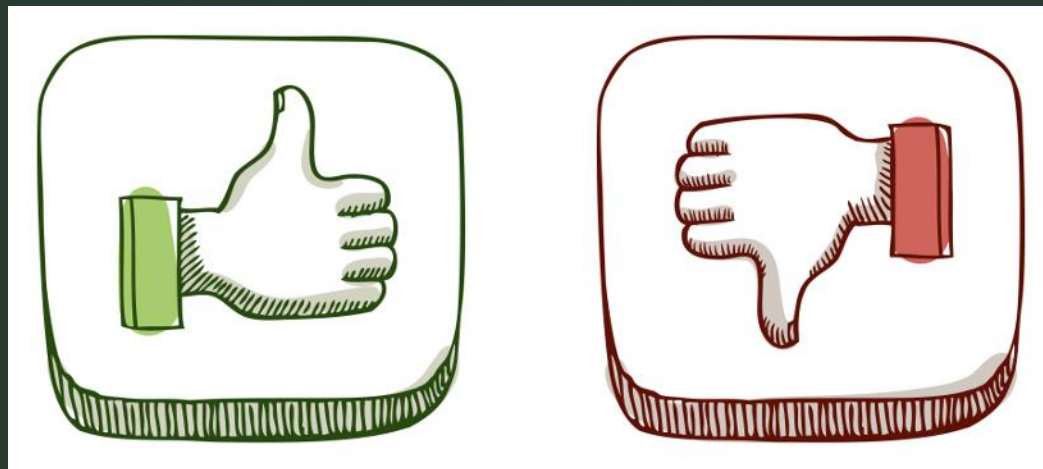
- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.

Zen de Python

- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.
- Ahora es mejor que nunca.
- Aunque nunca es a menudo mejor que ya mismo.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (namespaces) son una gran idea ¡Hagamos más de esas cosas!.

A continuación se presentan algunas ventajas y desventajas que están
presentan en el lenguaje Python

Ventajas y desventajas



Ventajas

Simplificado y rápido

- Este lenguaje simplifica mucho la programación “hace que te adaptes a un modo de lenguaje de programación, Python te propone un patrón”. Es un gran lenguaje para scripting, si usted requiere algo rápido (en el sentido de la ejecución del lenguaje), con unas cuantas líneas ya está resuelto.

```
classes.ts x
1  class Usuario {
2    nome: string;
3    sobreNome: string;
4
5    constructor(nome: string, sobreNome: string) {
6      this.nome = nome;
7      this.sobreNome = sobreNome;
8    }
9
10   darBoasVindas() {
11     return `Olá, ${this.nome} ${this.sobreNome}. Seja bem vindo!`;
12   }
13 }
14
```

Ventajas

Elegante y flexible

- El lenguaje le da muchas herramientas, si usted quiere listas de varios tipo de datos, no hace falta que declares cada tipo de datos. Es un lenguaje tan flexible usted no se preocupa tanto por los detalles.

```
var a = function (a,v) { return a+v }  
undefined  
console.log(a)  
f (a,v) { return a+v }  
undefined  
console.log(a.prototype)  
▼ {constructor: f} ⓘ  
  ► constructor: f (a,v)  
  ► __proto__: Object  
undefined  
console.log(a.__proto__)  
f () { [native code] }
```

Ventajas

Programación sana y productiva

- Programar en Python se convierte en un estilo muy sano de programar: es sencillo de aprender, direccionado a las reglas perfectas, le hace como dependiente de mejorar, cumplir las reglas, el uso de las líneas, de variables”. Además es un lenguaje que fue hecho con productividad en mente, es decir, Python le hace ser mas productivo, le permite entregar en los tiempos que me requieren.

Ventajas

Ordenado y limpio

- El orden que mantiene Python, es de lo que más le gusta a sus usuarios, es muy legible, cualquier otro programador lo puede leer y trabajar sobre el programa escrito en Python. Los módulos están bien organizados, a diferencia de otros lenguajes.

```
index.js ▶ ...  
1  const arr = [1, 2, 3, 4, 5];  
2  
3  arr.map(x => [x, x * 2]);  
4  
5  // [ Array(2), Array(2), Array(2) ]  
6  // 0: (2)[ 1, 2 ]  
7  // 1: (2)[ 2, 4 ]  
8  // 2: (2)[ 3, 6 ]  
9  
10 arr.flatMap(v => [v, v * 2]);  
11 // [1, 2, 2, 4, 3, 6, 4, 8, 5, 10]  
12
```


Ventajas

Portable

- Es un lenguaje muy portable (ya sea en Mac, Linux o Windows) en comparación con otros lenguajes. La filosofía de baterías incluidas, son las librerías que más usted necesita al día a día de programación, ya están dentro del interprete, no tiene la necesidad de instalarlas adicionalmente con en otros lenguajes.



Ventajas

Comunidad

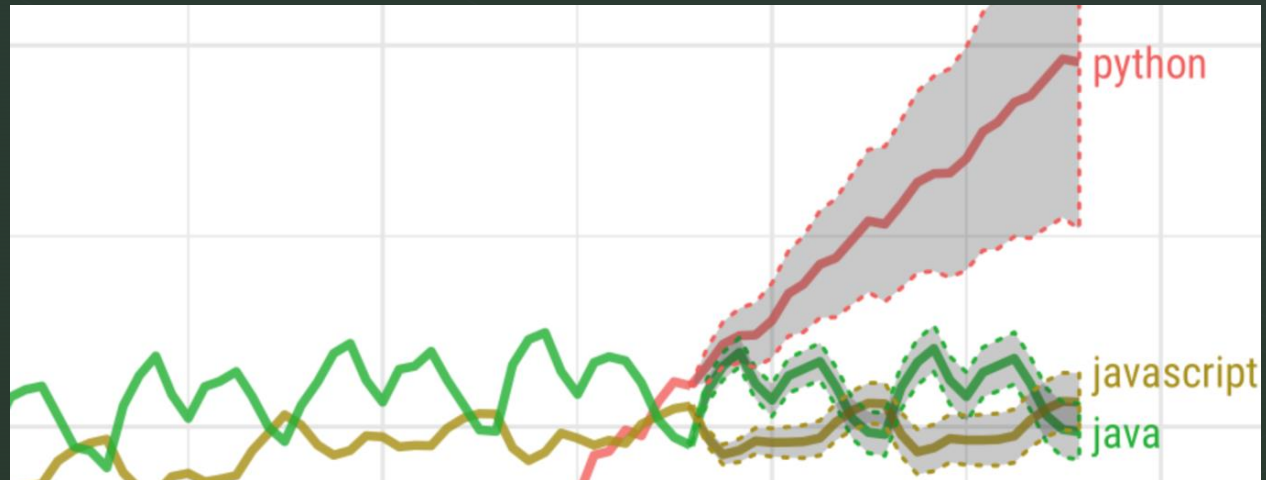
- Algo muy importante para el desarrollo de un lenguaje es la comunidad, la misma comunidad de Python cuida el lenguaje y casi todas las actualizaciones se hacen de manera democrática.



Desventajas

Curva de aprendizaje

- La “curva de aprendizaje cuando ya estás en la parte web no es tan sencilla”.



Desventajas

Hosting

- La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.



Desventajas

Librerías incluidas

- Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad, y optan a usar librerías de terceros.



Desde hace algunos años, Python es la palabra de moda en el vocabulario de cualquier programador, pero ¿por qué? Para poder responder a esta pregunta, empecemos por el principio, ¿Qué es Python?

Principales Usos de Python



Python en la Inteligencia Artificial (AI)

- Python es un lenguaje de escritura rápido, escalable, robusta y de código abierto, ventajas que hacen de Python un aliado perfecto para la Inteligencia Artificial.
- Permite plasmar ideas complejas con unas pocas líneas de código, lo que no es posible con otros lenguajes.
- Existen bibliotecas como «Keras» y «TensorFlow», que contienen mucha información sobre las funcionalidades del aprendizaje automático.
- Además, existen bibliotecas proporcionadas por Python, que se usan mucho en los algoritmos AI como Scikitl, una biblioteca gratuita de aprendizaje automático que presenta varios algoritmos de regresión, clasificación y agrupamiento.

Python en Big Data

- El uso de Python está muy extendido en el análisis de datos y la extracción de información útil para empresas.
- Además de su simplicidad, que es una gran ventaja, Python cuenta con bibliotecas de procesamiento de datos como 'Pydoop', que son de gran ayuda para los profesionales, ya que puede escribir un código de MapReduce en Python y procesar los datos en el clúster HDFS.
- Otras bibliotecas como «Dask» y «Pyspark» facilitan aún más el análisis y la gestión de datos. Python es rápido y fácilmente escalable, características que le ayudan a generar información en entornos de tiempo real y a convertir esa información a los lenguajes usados en Big Data.

Python en Data Science

- Desde la introducción de los motores numéricos de Python como 'Pandas' y 'NumPy', los investigadores han cambiado a Python desde el anterior lenguaje, MATLAB.
- Python se ocupa de los datos tabulares, matriciales y estadísticos, e incluso los visualiza con bibliotecas populares como «Matplotlib» y «Seaborn».

Python en Desarrollo Web

- Python permite construir mucho más con menos líneas de código, por lo que se crean prototipos de forma más eficiente.
- El framework Django, proporcionado por Python, es una ventaja para todos los desarrolladores, ya que puede usarse para crear aplicaciones web dinámicas y muy seguras.

Conclusiones

- La tendencia del mercado laboral muestra que la demanda de profesionales de Python crece exponencialmente cada año.
- Una prueba de ello es el informe de la web Stackoverflow, web de referencia para desarrolladores de todo el mundo, con más de 40 millones de usuarios mensuales, que ha presentado la evolución del volumen de búsquedas por tipo de lenguaje, y que muestra que, claramente, Python está a la cabeza.

Conclusiones

