

Tutorial

(/es/tutorial/) Python 3 Basic (/es/tutorial/python-3-basic-tutorial/) Tkinter (/es/tutorial/tkinter-tutorial/) Python Modules (/es/tutorial/python-modules-tutorial/) JavaScript (/es/tutorial/javascript/) Python Numpy (/es/tutorial/python-numpy/) Git (/es/tutorial/git/) Matplotlib (/es/tutorial/matplotlib/) PyQt5 (/es/tutorial/pyqt5/) Data Structure (/es/tutorial/data-structure/) Algorithm (/es/tutorial/algorithm/)

HowTo

(/es/howto/) Python Pygame (/es/howto/python-pygame/) Python (/es/howto/python/) Python Tkinter (/es/howto/python-tkinter/) Batch (/es/howto/batch/) PowerShell (/es/howto/powershell/) Python Pandas (/es/howto/python-pandas/) Numpy (/es/howto/numpy/) Python Flask (/es/howto/python-flask/) Django (/es/howto/django/) Matplotlib (/es/howto/matplotlib/) Docker (/es/howto/docker/) Plotly (/es/howto/plotly/) Seaborn (/es/howto/seaborn/) Matlab (/es/howto/matlab/) Linux (/es/howto/linux/) Git (/es/howto/git/) C (/es/howto/c/) Cpp (/es/howto/cpp/) HTML (/es/howto/html/) JavaScript (/es/howto/javascript/) jQuery (/es/howto/jquery/) TypeScript (/es/howto/typescript/) Angular (/es/howto/angular/) React (/es/howto/react/) CSS (/es/howto/css/) PHP (/es/howto/php/) Java (/es/howto/java/) Go (/es/howto/go/) Kotlin (/es/howto/kotlin/) Node.js (/es/howto/node.js/) Csharp (/es/howto/csharp/) Rust (/es/howto/rust/) Ruby (/es/howto/ruby/) Arduino (/es/howto/arduino/) MySQL (/es/howto/mysql/) MongoDB (/es/howto/mongodb/) Postgres (/es/howto/postgres/) R (/es/howto/r/) VBA (/es/howto/vba/) Scala (/es/howto/scala/) Raspberry Pi (/es/howto/raspberry-pi/)

Referencia

(/es/api/) Python Pandas (/es/api/python-pandas/) Numpy (/es/api/numpy/) Scipy (/es/api/scipy/)

TABLA DE CONTENIDO

- 1. Utilice `Simple Crypt` para cifrar una cadena en Python
- 2. Utilice la biblioteca de `cryptocode` para cifrar una cadena en Python
- 3. Utilice el paquete `Cryptography` para cifrar una cadena en Python
- 4. Utilice el algoritmo `RSA` para cifrar una cadena en Python



```
import rsa

pubkey, privkey = rsa.newkeys(512)

str1 = "I am okay"

enctex = rsa.encrypt(str1.encode(),pubkey)

dectex = rsa.decrypt(enctex, privkey).decode()
print("The primordial string: ", str1)
print("The Encrypted message: ", enctex)
print("The Decrypted message: ", dectex)
```

Cifrar una cadena de Python

(<http://www.facebook.com/sharer.php?src=bm&u=https%3a%2f%2fwww.delftstack.com%2fes%2fhowto%2fpython%2fpython-encrypt-string%2f&t=Cifrar%20una%20cadena%20de%20Python>)  (<https://twitter.com/intent/tweet/?text=How-To%27s%20de%20Python-Cifrar%20una%20cadena%20de%20Python&url=https%3a%2f%2fwww.delftstack.com%2fes%2fhowto%2fpython%2fpython-encrypt-string%2f&hashtags=web,development>)  (<https://www.linkedin.com/shareArticle?mini=true&url=https%3a%2f%2fwww.delftstack.com%2fes%2fhowto%2fpython%2fpython-encrypt-string%2f&title=How-To%27s%20de%20Python-Cifrar%20una%20cadena%20de%20Python&source=https%3a%2f%2fwww.delftstack.com%2fes%2fhowto%2fpython%2fpython-encrypt-string%2f&summary=Short%20summary>)

TABLA DE CONTENIDO

1. Utilice `Simple Crypt` para cifrar una cadena en Python
2. Utilice la biblioteca de `cryptocode` para cifrar una cadena en Python
3. Utilice el paquete `Cryptography` para cifrar una cadena en Python
4. Utilice el algoritmo `RSA` para cifrar una cadena en Python

El cifrado se puede definir como el proceso que transforma el texto sin formato en texto cifrado. Esencialmente, se usa para codificar los datos. El proceso de cifrado requiere una clave, que luego se puede utilizar para descifrar el mensaje original.

Este artículo discutirá los diferentes métodos que puede utilizar para cifrar una cadena en Python.

Utilice `Simple Crypt` para cifrar una cadena en Python

Usar el módulo `Simple Crypt` es la forma más rápida y sencilla de lograr el proceso de encriptación para Python 2.7 y 3. Este método convierte el texto plano en un texto cifrado en segundos, con la ayuda de una sola línea de código.

El módulo `PyCrypto` proporciona la implementación del algoritmo para esta biblioteca y utiliza el cifrado `AES256`. Este método incorpora una verificación `HMAC` con el cifrado `SHA256` para notificar cuando el texto cifrado es alterado o modificado.

`Simple Crypt` tiene dos funciones: “cifrar” y “descifrar”. Es esencial instalar los módulos `pycrypto` y `Simple Crypt` para usar ambas funciones.

El siguiente código utiliza `simplecrypt` para cifrar una cadena en Python:

```
from simplecrypt import encrypt, decrypt
passkey = 'wow'
str1 = 'I am okay'
cipher = encrypt(passkey, str1)
print(cipher)
```

La salida de este código sería un grupo de letras aleatorias.

Para decodificar / descifrar el texto cifrado, usaremos el siguiente comando:

```
print(decrypt('wow', cipher))
```

Producción:

```
I am okay
```



Utilice la biblioteca de cryptocode para cifrar una cadena en Python

El término `cryptocode` es una biblioteca simple que nos permite cifrar y descifrar cadenas de forma segura y sencilla en Python 3 o superior. Recuerde que esta biblioteca debe instalarse manualmente; se puede hacer usando el comando `pip`.

El siguiente programa utiliza la biblioteca `cryptocode` para cifrar una cadena en Python:

```
import cryptocode

str_encoded = cryptocode.encrypt("I am okay","wow")
## And then to decode it:
str_decoded = cryptocode.decrypt(str_encoded, "wow")
print(str_decoded)
```

Producción:

```
I am okay
```

El primer parámetro de la función sería la cadena que debe cifrarse. El segundo parámetro debe ser la clave, que se utilizará con el propósito de descifrar.



Utilice el paquete Cryptography para cifrar una cadena en Python

`Cryptography` es un paquete de Python que se puede utilizar para lograr el cifrado de clave simétrica. El cifrado de clave simétrica es una forma en que usamos la misma clave para el proceso de codificación y decodificación.

Es necesario instalar la biblioteca `Cryptography` para poder utilizar este método de cifrado; esto se puede hacer usando el comando `pip`.

El siguiente código utiliza las funciones del paquete `cryptography` para cifrar una cadena en Python:

```
from cryptography.fernet import Fernet

str1 = "I am okay"
key = Fernet.generate_key()

fernet = Fernet(key)

enctex = fernet.encrypt(str1.encode())

dectex = fernet.decrypt(enctex).decode()

print("The primordial string: ", str1)
print("The Encrypted message: ", enctex)
print("The Decrypted message: ", dectex)
```

Producción:

```
The primordial string: I am okay
The Encrypted message: <a stack of random letters>
The Decrypted message: I am okay
```

En el código anterior, importamos `Fernet` del módulo `cryptography.fernet`. Luego, generamos una clave de cifrado que se utilizará tanto para codificar como para decodificar. La clase `Fernet` se instancia con la clave de cifrado. Luego, la cadena se cifra con la instancia de `Fernet`. Finalmente, se descifra con la instancia de la clase `Fernet`.

El cifrado de clave simétrica es una forma sencilla de cifrar una cadena. El único inconveniente es que es comparativamente menos seguro. Cualquiera que tenga la clave puede leer el texto cifrado.



Utilice el algoritmo RSA para cifrar una cadena en Python

El algoritmo `rsa` en Python implementa el cifrado de clave asimétrica. El cifrado de clave asimétrica utiliza dos claves diferentes para el proceso de cifrado y descifrado.

Estas dos claves son la clave privada y la clave pública. La clave pública es pública y se utiliza para el proceso de cifrado. Cualquiera que tenga la clave pública puede cifrar y enviar los datos; solo el receptor tiene la clave privada. Además, alguien con acceso a él también puede descifrar los datos.

La biblioteca `rsa` debe instalarse primero para utilizar este método. Puede utilizar el comando general `pip` para la instalación de esta biblioteca.

El siguiente código utiliza el algoritmo `rsa` para cifrar una cadena en Python:

```
import rsa

pubkey, privkey = rsa.newkeys(512)

str1 = "I am okay"

encTex = rsa.encrypt(str1.encode(),pubkey)

decTex = rsa.decrypt(encTex, privkey).decode()
print("The primordial string: ", str1)
print("The Encrypted message: ", encTex)
print("The Decrypted message: ", decTex)
```

Producción:

```
The primordial string: I am okay
The Encrypted message: <a stack of random letters>
The Decrypted message: I am okay
```

En el código anterior, se importa el módulo `rsa` y las claves pública y privada se generan con la ayuda de la función `rsa.newkeys()`. Luego, la cadena se cifra mediante la clave pública. A continuación, la cadena se puede descifrar utilizando sólo la clave privada. Las claves pública y privada en el caso de este programa son `pubkey` y `privkey`, respectivamente.

El cifrado de clave asimétrica proporciona una mayor seguridad que el cifrado de clave simétrica. Al igual que en el primero, una clave privada específica solo es accesible para el receptor y se utiliza para el propósito de descifrado. Mientras que solo existe una clave en este último, el proceso de descifrado es más fácil y menos seguro frente a ataques de terceros en medio del remitente y el receptor.

Artículo relacionado - Python String

(/es/tags/python-string/)

• [Cómo eliminar los comas de una cadena en Python \(/es/howto/python/remove-commas-from-string-in-python/\)](/es/howto/python/remove-commas-from-string-in-python/)

- [Cómo comprobar si una cadena está vacía de forma ptónica \(/es/howto/python/how-to-check-a-string-is-empty-in-a-pythonic-way/\)](/es/howto/python/how-to-check-a-string-is-empty-in-a-pythonic-way/)
- [Convertir una cadena en nombre de variable en Python \(/es/howto/python/python-string-to-variable-name/\)](/es/howto/python/python-string-to-variable-name/)
- [Cómo eliminar los espacios en blanco de una cadena en Python \(/es/howto/python/how-to-remove-whitespace-in-a-string/\)](/es/howto/python/how-to-remove-whitespace-in-a-string/)
- [Cómo extraer números de una cadena en Python \(/es/howto/python/python-extract-number-from-string/\)](/es/howto/python/python-extract-number-from-string/)
- [Cómo convertir una cadena a datetime en Python \(/es/howto/python/how-to-convert-string-to-datetime/\)](/es/howto/python/how-to-convert-string-to-datetime/)

Artículo relacionado - Python Encryption

(/es/tags/python-encryption/)

• [Cómo cifrar con RSA en Python \(/es/howto/python/rsa-encryption-python/\)](/es/howto/python/rsa-encryption-python/)

