Implementation:

MD5:

```
import hashlib

hash_obj = hashlib.md5(b'Hello, Python!')

print(hash_obj.hexdigest())
```

Output

```
a0af7810eb5fcb84c730f851361de06a
```

SHA1:

```
import hashlib

str = "HelloWorld"

result = hashlib.sha1(str.encode())

# printing the equivalent hexadecimal value.

print("The hexadecimal equivalent of SHA1 is : ")

print(result.hexdigest())
```

Output

```
The hexadecimal equivalent of SHA1 is :

db8ac1c259eb89d4a131b253bacfca5f319d54f2
```

Conclusion:

MD5 and SHA1 are cryptographic hash methods that are used to ensure security. MD5 generates a 128-bit hash result and is faster, however it provides insufficient security, making it outdated because of its weaknesses. SHA1 generates a 160-bit hash value and provides higher security, but it is slower and has been discovered vulnerable to attacks over time. Due to these drawbacks, stronger hash algorithms such as SHA-256 are now recommended.