aardio 范例: aes

```
//AES
import console;
import crypt.bin;
import crypt.aes;

//创建AES加密算法容器
var aes = crypt.aes();

//不指定加密向量时默认设为密钥的值
//aes.setInitVector("1234567812345678")

//修改工作模式为 工作模式 ECB
//aes.setKeyParamMode(2/*_CRYPT_MODE_ECB*/);

//设置密钥(最大32个字节)
aes.setPassword("1234567812345678");

//加密
var ciphertext = aes.encrypt("Test String");

//BASE64编码加密结果
console.log( crypt.bin.encodeBase64( ciphertext ) );

//解密
var plaintext = aes.decrypt(ciphertext);
console.log(plaintext);

console.pause(true);
/*
AES 加密算法多编程语言通用写法( aardio,PHP,C#,Java  ):
http://bbs.aardio.com/forum.php?mod=viewthread&tid=13818

不同编程语言中AES加解密结果要保持一致要注意以下一些要点:

1、工作模式 CBC ，填充模式 PKCS7，不同语言要保持一致。
要注意 PKCS5 与 PKCS7 的填充规则是相同的，区别是PKCS5填充1到8字节,PKCS7填充1到255字节,
而AES实际使用的数据区分组为 16 字节（128位），所以即使填充模式指定 PKCS5 - 实际使用的也是 PKCS7。
下面链接里的 JAVA代码里只能选 PKCS5 ，而 C#代码里只能选 PKCS7，这都是兼容的没有问题。
2、在下面的示例中，加密向量统一设为与密钥相同。
3、不同编程语言使用的文本编码要一致，同一个字符串，使用UTF8或GBK编码在内存中存储的实际数据可能是不一样的。
在aardio中默认编码为UTF-8,使用 string.fromto进行转换为其他编码。
4、如果加密后返回的密文用了BASE64或16进制编码，那么在解密时同样也先做对应的逆向解码。

可以添加下面的函数以支持 zero padding 填充加密（参数 str 必须指定全部待加密数据）
aes.encryptWithZeroPadding = function(str,hHash,flags){
    var buf = raw.buffer(#str + owner.blockSize - #str % owner.blockSize,str);
    var ret = owner.encrypt(buf,true,hHash,flags);
    if(#ret) return string.left(ret,-owner.blockSize-1);
}
*/
```

Markdown 格式