

Forum:[Hack Forums](#)

Thread: [Replace misordered base64 string to plain text in a file](#)

Sparkling Jack - July 10, 2018, 7:02 pm

Hello guys.

Recently I was needed to make a python script to replace misorderd base64 string to plaint text in a file.

For example:

`base64decode(textorder("Wavxmd5V3b=U")) --> iloveyou`
original base64 code of "iloveyou" is `aWxvdmV5b3U=`

Here is the code:

```
Code def changeStr(str):
    newStr = list(str)
    resStr = list(str)
    for i in xrange(0,len(newStr),2):
        if i==len(newStr)-1:
            resStr[i] = newStr[i]
        else:
            resStr[i]=newStr[i+1]
            resStr[i+1] = newStr[i]
    str = "".join(resStr)
    return str
def base64Decode(str):
    src1=changeStr(str)
    result=base64.b64decode(src1)
    return result
def find_str(s, char):
    index = 0
    res = []
    if char in s:
        c = char[0]
        for ch in s:
            if ch == c:
                if s[index:index+len(char)] == char:
                    res.append(index+len(char))
                index += 1
    return res
def find_End(str):
    index =0;
    if "\"" in str:
        for ch in str:
            if ch == "\"":
                return str[:index]
            index+=1
    return "None"
if __name__ == '__main__':
```

```

infile = open('1.txt')
outputfile = open('2.txt','w')
for line in infile:
    pos = find_str(line, "base64decode(textorder(")
    if pos:
        for i in reversed(pos):
            source = find_End(line[i+1:])
            result = base64Decode(source)
            src = "base64decode(textorder(\""+source+"\"))"
            line = line.replace(src, "\"" + result + "\"")
outputfile.write(line)

```

Hope this will be a bit of help for you!

TyrantKingBen - August 3, 2018, 8:19 pm

(Jul 10, 2018 07:02 PM)Sparkling Jack Wrote: [\[CITE\]](#) Hello guys.

Recently I was needed to make a python script to replace misorderd base64 string to plaint text in a file.

For example:

base64decode(textorder("Wavxmd5V3b=U")) --> iloveyou

original base64 code of "iloveyou" is aWxvdmV5b3U=

Here is the code:

```

Code def changeStr(str):
    newStr = list(str)
    resStr = list(str)
    for i in xrange(0,len(newStr),2):
        if i==len(newStr)-1:
            resStr[i] = newStr[i]
        else:
            resStr[i]=newStr[i+1]
            resStr[i+1] = newStr[i]
    str = "".join(resStr)
    return str
def base64Decode(str):
    src1=changeStr(str)
    result=base64.b64decode(src1)
    return result
def find_str(s, char):
    index = 0
    res = []
    if char in s:
        c = char[0]
        for ch in s:
            if ch == c:
                if s[index:index+len(char)] == char:
                    res.append(index+len(char))
                index += 1
    return res
def find_End(str):
    index =0;
    if "\"" in str:
        for ch in str:

```

```

    if ch == "\":
        return str[:index]
    index+=1
    return "None"
if __name__ == '__main__':
    infile = open('1.txt')
    outputfile = open('2.txt','w')
    for line in infile:
        pos = find_str(line, "base64decode(textorder(")
        if pos:
            for i in reversed(pos):
                source = find_End(line[i+1:])
                result = base64Decode(source)
                src = "base64decode(textorder(\""+source+"\"))"
                line = line.replace(src, "\"" + result + "\")")
    outputfile.write(line)

```

Hope this will be a bit of help for you!

The nice thing about Python is almost anything can be accomplished in very little code. Your changeStr can be rewritten to make use of Python's features.

One Liner:

Code def changeStr(scrambled):

```

    return "".join([a + b for b, a in zip(scrambled[::2], scrambled[1::2])]) + (scrambled[-1] if
(len(scrambled) % 2) else "")

```

Expanded:

Code def changeStr(scrambled):

```

    unscrambled = "".join([a + b for b, a in zip(scrambled[::2], scrambled[1::2])])

```

```

    if (len(scrambled) % 2):
        unscrambled += scrambled[-1]

```

```

    return unscrambled

```

Also, it's tempting to call the parameter 'str', but that will override the 'str' function, so best to avoid that habit.

Sparkling Jack - August 9, 2018, 1:02 am

(Aug 3, 2018 08:19 PM)TyrantKingBen Wrote: [\[CITE\]](#) The nice thing about Python is almost anything can be accomplished in very little code. Your changeStr can be rewritten to make use of Python's features.

One Liner:

Code def changeStr(scrambled):

```

    return "".join([a + b for b, a in zip(scrambled[::2], scrambled[1::2])]) + (scrambled[-1] if
(len(scrambled) % 2) else "")

```

Expanded:

Code def changeStr(scrambled):

```

    unscrambled = "".join([a + b for b, a in zip(scrambled[::2], scrambled[1::2])])

```

```

    if (len(scrambled) % 2):
        unscrambled += scrambled[-1]

```

```

    return unscrambled

```

Also, it's tempting to call the parameter 'str', but that will override the 'str' function, so best to avoid that habit.

Thank you,TyrantKingBen