Python Challenge Code

jupyter task0 （已自动保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

运行 代码

In [3]:
```
2**38
```

Out[3]:
```
274877906944
```

jupyter task1 （更改未保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

运行 代码

In [1]:
```
# Define the original message string
message = "g fmnc wms bgblr rpylqjyrc gr zw fylb. rfyrq ufyr amknsrcpq ypc dmp. " \
          "bmgle gr gl zw fylb gq glcddgagclr ylb rfyr'q ufw rfgq rcvr gq " \
          "qm jmle. sqgle qrpgle.kyicrpylq() gq pcamkkclbcb. lmu ynnjw ml " \
          "rfc spj."\
          "map"

# Define a function to shift a single character by a given number of positions
def shift_char(c, n):
    # Define the alphabet as a string of lowercase letters
    alphabet = 'abcdefghijklmnopqrstuvwxyz'
    # If c is not a lowercase letter, return it as-is
    if c not in alphabet:
        return c
    # Otherwise, shift the character by n positions and wrap around if necessary
    shifted_index = (alphabet.index(c) + n) % len(alphabet)
    return alphabet[shifted_index]

# Define a function to shift a string by a given number of positions
def shift_string(s, n):
    # Map each character in s to its shifted version and join them into a new string
    return ''.join(shift_char(c, n) for c in s)

# Shift the original message by two positions to get the translated message
translated_message = shift_string(message, 2)

# Print the translated message
print(translated_message)
```

```
i hope you didnt translate it by hand. thats what computers are for. doing it in by hand is inefficient and that's
why this text is so long. using string.maketrans() is recommended. now apply on the url.ocr
```

jupyter task2 （更改未保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

运行 代码

In [1]:
```
import urllib.request
import re

url = "http://www.pythonchallenge.com/pc/def/ocr.html"
response = urllib.request.urlopen(url)
data = response.read().decode()

# Find the string containing the hidden message
message = re.findall("<!--(.*?)-->", data, re.DOTALL)[-1]

# Count the number of occurrences of each character
count = {}
for c in message:
    if c in count:
        count[c] += 1
    else:
        count[c] = 1

# Find the character with the number of occurrences of 1, which is the hidden message
result = ""
for c in message:
    if count[c] == 1:
        result += c

print(result)
```

```
equality
```

jupyter task3 （更改未保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

运行 代码

In [1]:
```
import re
import urllib.request

url = "http://www.pythonchallenge.com/pc/def/equality.html"
response = urllib.request.urlopen(url)
data = response.read().decode("utf-8")

result = ""

pattern = r"[^A-Z][A-Z]{3}([a-z])[A-Z]{3}[^A-Z]"
matches = re.findall(pattern, data)

for match in matches:
    result += match

print(result)
```

```
linkedlist
```

jupyter task4 最新检查点: 9 分钟前 （更改未保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 可信 Python 3 (ipykernel)

运行 代码

In [6]:
```
import urllib.request
import re

url = "http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing="
nothing = "6175.2" # First value of nothing
pattern = re.compile("and the next nothing is (\d+)") # Regular expression to match the next nothing

# Loop until the last nothing is found
while True:
    response = urllib.request.urlopen(url + nothing)
    data = response.read().decode()
    print(data)
    match = pattern.search(data)
    if match is None:
        break
    nothing = match.group(1)
```

```
and the next nothing is 6927
and the next nothing is 82930
and the next nothing is 35846
and the next nothing is 31785
and the next nothing is 41846
and the next nothing is 72387
and the next nothing is 59334
and the next nothing is 65520
and the next nothing is 93781
and the next nothing is 55840
and the next nothing is 80842
and the next nothing is 59022
and the next nothing is 23298
and the next nothing is 27709
and the next nothing is 96791
and the next nothing is 75635
and the next nothing is 52899
and the next nothing is 66831
peak.html
```

jupyter task5 最新检查点: 几秒前 （已自动保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

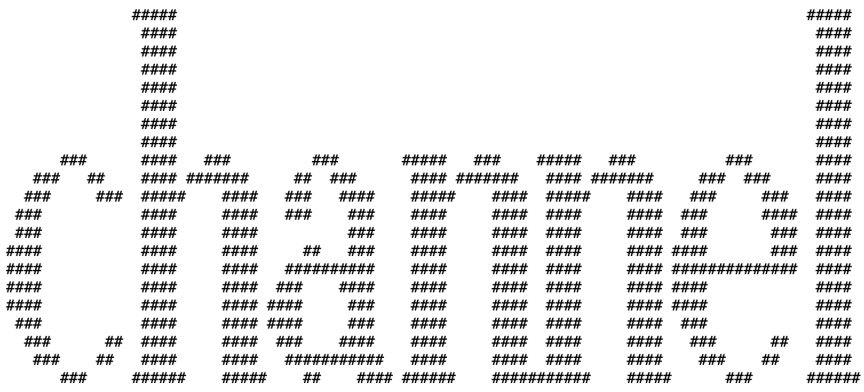
运行 代码

In [3]:
```
import pickle
import urllib.request

# Read the source code of a web page
url = "http://www.pythonchallenge.com/pc/def/banner.p"

# Download the pickle file and deserialise it
with urllib.request.urlopen(url) as f:
    data = pickle.load(f)

# Output patterns
for row in data:
    print("".join([char * count for char, count in row]))
```



jupyter task6 （更改未保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助 不可信 Python 3 (ipykernel)

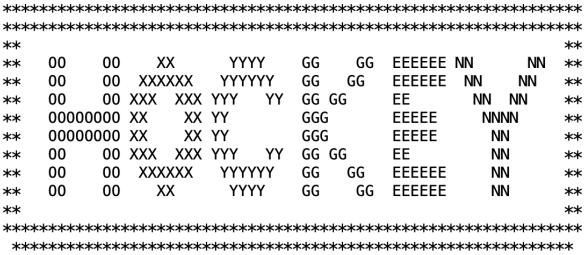
运行 代码

In [2]:
```
import zipfile
import re

zip_file = zipfile.ZipFile("channel.zip")

# In the first step, find the first file and get the numbers in the comments
file_name = "90052.txt"
comments = []
while True:
    text = zip_file.read(file_name).decode("utf-8")
    comments.append(zip_file.getinfo(file_name).comment.decode("utf-8"))
    match = re.search(r"Next nothing is (\d+)", text)
    if match:
        file_name = match.group(1) + ".txt"
    else:
        break

# The second step is to find the comments in the order of the numbers obtained
print("".join(comments))
```



jupyter task7 最新检查点: 2 小时前 （已自动保存）

注销

文件 编辑 查看 插入 单元格 内核 Widgets 帮助

内核正在启动,请等待... 可信 Python 3 (ipykernel)

运行 代码

In [6]:
```
from PIL import Image

# Open image
img = Image.open("oxygen.png")
# Get the middle row of pixel data from the bottom pixel block of the image
width, height = img.size
middle_row = [img.getpixel((x, height//2)) for x in range(width)]
gray_values = [color[0] for color in middle_row[::7]]

# Convert pixel data to characters
message = "".join([chr(i) for i in gray_values])
print(message)
```

```
smart guy, you made it. the next level is [105, 110, 116, 101, 103, 114, 105, 116, 121]pe_
```

In [7]:
```
text = "smart guy, you made it. the next level is [105, 110, 116, 101, 103, 114, 105, 116, 121]"

# Extracts a list from a string and converts the numbers in it to their corresponding characters
next_url = ''.join(chr(int(c)) for c in text.split('[')[-1].split(']')[0].split(', '))

print(next_url)
```

```
integrity
```