**数据结构实验设计报告**

**题 目：­­**基于马尔科夫链的写作机器人

**专 业：**计算机类

**小组成员：**计科A2001 邓岳

计科A2001 熊圳

计科A2001 赵千慧

计科A2001 赵凯

# 任务书

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 数据结构实验 | | |
| 设计题目 | 基于马尔科夫链的写作机器人 | | |
| 指导教师 | 彭四伟 | 时 间 | 2022年1月3日-7日 |
| **一、教学要求**  1. 巩固学生使用程序设计语言进行程序设计、编写和调试的能力； 2. 掌握数据结构常用经典算法原理、实现及其应用； 3. 提高计算机专业技能的自学能力； 4. 掌握书写程序设计说明文档的能力；  **二、设计要求及成果**  1. 分析课程设计题目的功能需求，完成一个题目； 2. 写出详细设计说明（至少**包括功能实现分析**和**模块流程图**） 3. 编写程序代码，调试程序使其能正确运行（代码书写要规范，标示符要见名知意，要有必要的注释）； 4. 设计完成的软件要便于操作和使用，在软件界面上有作者版权信息； 5. 设计完成后提交设计报告（请严格按照模板进行排版）；  **三、评分标准**  1. 使用的算法（20分）：使用了哪些数据结构课程中学到的经典算法；  2. 程序功能完成情况（20分）：功能是否能用，是否易用，是否完备；  3. 程序界面（15分）：界面是否美观大方，设计合理；  4. 课程设计报告（30分）：报告是否内容充实，图表丰富，格式正确，没有错别字；  5. 程序健壮性（15分）：运行过程中发现程序一个BUG，扣3分，扣完为止；  **四、郑重提示**  原则上，只要编码并实现了一个可用的基本程序即可及格！但如果发现非自己编写、拷贝他人（含网络），直接不及格！ | | | |

# 目 录

[任务书 I](#_Toc9021)

[目 录 II](#_Toc27954)

[第一章 需求分析 3](#_Toc11897)

[1.1 引言 3](#_Toc2148)

[1.2 任务概述 3](#_Toc28625)

[1.3 数据描述 3](#_Toc11953)

[1.4 功能需求 3](#_Toc21642)

[1.5 运行需求 3](#_Toc13764)

[1.6 任务计划 3](#_Toc27446)

[第二章 概要设计 4](#_Toc2865)

[2.1 总体设计 4](#_Toc18233)

[2.2 面向对象设计 4](#_Toc26726)

[2.3 运行界面设计 4](#_Toc18813)

[第三章 详细设计 5](#_Toc19218)

[3.1 系统设计与实现 5](#_Toc16310)

[3.2 模块设计 5](#_Toc32206)

[第四章 测试分析 6](#_Toc5868)

[第五章 用户手册（可选） 7](#_Toc6928)

[第六章 课程设计总结 8](#_Toc800)

[附录1 工作量及成绩自评表 9](#_Toc1195)

[附录2：程序代码（仅电子版需要、纸质版不需要） 10](#_Toc19781)

# 需求分析

## 引言

自动生成文本是自然语言处理中一大研究领域，目前常用的生成方法包括深度学习和马尔可夫链等。在本次数据结构实验设计中，我们利用马尔可夫链作为基本原理制作了一个写作机器人。

马尔可夫链是描述可能事件序列的随机模型，其中每个事件的概率仅取决于前一个事件中获得的状态。马尔可夫链文本生成的思想与此相同，即试图找出某个词出现在另一个词之后的概率。为了确定转换的概率，我们用一些文本来训练模型。它使用历史文本，将其分割成单个单词（或单词集），然后随机选择一个单词，之后基于之前选择的单词序列随机选择一个相似的单词。这个过程一直继续，直到生成一段由规定数量的句子组成的文本段落。

利用马尔可夫链文本生成方法，我们可以用一个轻便简易的方式来模仿智能，生成一些看起来有意义的文本段落。随机文本生成本来是个比较麻烦的问题，但是运用马尔科夫链的数学统计模型知识重新进行表述后，可以使整个问题大大简化。

## 任务概述

本实验通过对一段输入文本进行分词、词频分析、构建散列表数据结构后，采用适当的随机方法，自动生成并输出一段与原文章在词法、语义、主题上近似的文段。

## 数据描述

数据以.txt文件形式保存，UTF-8进行编码，是英文形式的自然语言字符串。内置文学主题包含：哈利波特七部曲、银河系漫游指南；同时接受用户自定义的输入文章。

## 功能需求

用户可以选择使用内置的演示文档和自行上传文档，实现对两类文档的文本解析，生成相应数据集之后，自动返回生成的结果文本到网页上。

## 运行需求

开发环境：Python=3.9、Flask=2.0.2

开发工具：Pycharm 2021.3(Professional Edition)、Visual Studio Code、Postman

## 任务计划

**前端：**

1. 设计网页，完成选择、上传、提交等功能

**后端：**

1. 解析文本：利用马尔可夫链算法，对已有文本进行分词，统计频率
2. 语段生成：根据解析的文本，进行随机文本生成
3. 返回前端：将生成的文本输出在网页上

**第二章 概要设计**

## 总体设计

本项目分为前端和后端两个部分，前端部分主要处理接受用户的请求和参数，并且把参数传给后端，后端部分主要实现解析文本、生成文本，并将生成的文本返回给前端。

首先在前端，用户可以选择我们给出的演示文档，或者自行上传.txt文件进行解析，选择要生成的句子的数量，后端就可以开始解析。后端对于给定的文本，会首先将文本进行分词，再逐词解析，生成文中每个单词的频数统计表。根据该数据集，每生成一个单词，会根据马尔可夫链随机生成下一个与之有关连的单词，直到句子结束。最终生成指定数量的句子，返回给前端，前端再将其展示在页面中。

## 主要技术问题

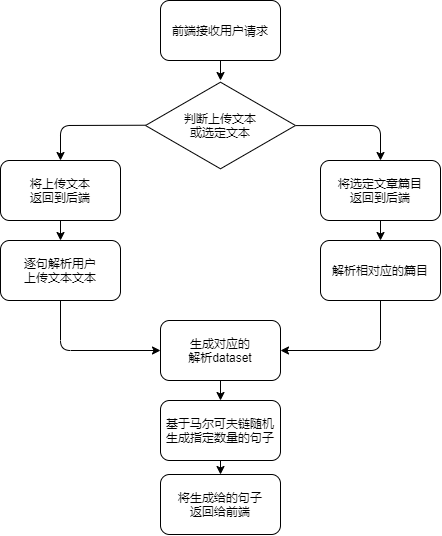
1. 对于输入文章的格式化与文章句段的切分问题
2. 对于汉语的分词问题
3. 在语段分析后，建立高效读写数据结构的问题

## 运行界面设计



# 第三章 详细设计

## 系统设计与实现



接口设计

**1. 获取生成中文句子**

**1）功能说明**

通过选定的主题和句子数量生成相应

**2）接口说明**

请求方式：POST

请求地址：http://ip:port/login

请求参数： 无

**3）请求参数**

示例

127.0.0.1:5000/login

**4）响应参数**

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| result | string | 返回生成的文本 |

**示例**

{

"result": "我只有一个儿子，求你行行好救活他吧，我想笑却笑不出来，我走过去，拍拍她让她歇一歇了，天黑后路上的石子绊着凤霞心里疼得受不了我对爹说，你这些日子忙坏了，你眼睛都睁圆眼睛看着我喊，你死了，那几个红卫兵挥挥手。"

}

**2. 获取生成英文句子**

**1）功能说明**

通过选定的主题和句子数量生成相应的英文句子

**2）接口说明**

请求方式：POST

请求地址：http://ip:port/loginen

请求参数： 无

**3）请求参数**

示例

127.0.0.1:5000/login

4）响应参数

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| result | string | 返回生成的文本 |

**示例**

{

"result": "Relaxing in his manic desperation and covered with white furry creature bounded through think it's could neither judge nor have died this bobbing, glistening in the din "Most of muscle movements…"

}

## 模块设计

**前端**

前端通过web实现，页面由html搭建框架，通过css美化。页面主要有三大部分：

第一部分：页面的信息

包括页面标题、logo、Github链接以及中英文跳转链接。

第二部分：输入部分

用户输入通过一个form表单进行输入以及提交。 表单中有四个部分，<select>下拉框，文件域，number字段和一个<button>按钮。用户可以通过下拉框选择预置好的文章或者点击自行上传按钮上传一篇文章，然后在文本框中输入要生成的句子数量（限制了类型为数字，输入范围是0~1000），最后点击生成按钮。此时表单数据会通过flask框架传递给后端。自行上传按钮本质上是一个文件域，通过css重新设计样式使其和生成按钮统一。

第三部分：输出部分

后端将生成的句子返回前端，前端在目标区域显示出生成的句子。

**后端**

**解析文本模块**

解析收到的txt格式文本文件，首先消除文本中的\n和类似“Page”字样，然后利用正则表达式将文本根据文本结束符（句号、感叹号、问号等）分隔句子。对每个句子添加”^”作为句首标记，”$”作为句尾标记，循环遍历每个句子，对每个单词和其后的单词，如果这样的组合从未出现，则将其放入字典中，并将count设置为1；如果这样的组合已经出现，则将对应的count加1。这样可以得到一个以字典为基础的数据集。

**生成文本模块**

针对生成的数据集，设定一个句子的句首必须为”^”，以此为首单词，根据马尔可夫链的思想，生成下一个单词，随机生成的过程如下：首先，对某一单词的下一单词列表的count值进行遍历求和，得到下一单词出现的频数总和，生成一个随机数范围在1到频数总和之间，继续遍历下一单词的字典，得到下一单词，直到生成的下一单词为”$”，停止生成。

**Service模块**

将后端的两个实现类模块放在dao中，使用article\_service和text\_service分别实现利用演示文档生成文章和利用用户上传文档生成文章的两个业务逻辑。根据前端复选框中选择的文本名称和句子数量，对应不同的文本生成对应数量的句子。

将实现类和业务逻辑分离，降低前后端的耦合度，方便修改后端进行进一步的升级调整。

# 第四章 测试分析

测试方法：针对由程序提供文章的生成方式，我们对生成句的可行数据进行了测试；针对由用户上传文章的生成方式，我们广泛的测试了不同格式的文件上传，从而检查程序的健壮性。

**发现的问题：**

1、存在输入数据不符合实际约束条件(如输入的数小于等于0或为浮点数，甚至其他字符串)的风险问题

2、如果用户上传的文本并非.txt格式的文件，网页会报错。因为后端无法解析这样的文字，有一部分可以被后端解析的文字中间含有非法字符会导致程序崩溃。

**解决方案：**

1、考虑到页面显示与最终效果，我们将生成句数约束在0～1000的正整数中，并预设定为10。

2、首先在前端的上传选项中只保留.txt格式的文件，其次在后端使用try/except但检测到异常时，直接返回提示文字。

**【对最终成果的评价】**

在目前的实验中，通过上传文本方式或选定文本的方式，程序都能顺利的解析文章并生成与之对应的数据集，再由用户输入的句子数量，按马尔可夫过程，产生在词法、主题、形式上与原文相近的文章内容。但就自然语言的逻辑性，生成的文段效果仍具有较大提升空间。

# 第五章 用户手册（可选）

**基于马尔可夫链的写作机器人**

**A Writing bot based on Markov Chain**

该写作机器人基于马尔可夫链设计，利用flask框架在前端进行演示。

马尔可夫链是随机过程中的概念，基于该算法可以根据已有的文本生成一些看似真实的文字，每生成一个词，它可以根据下一个词出现的频数随机选择下一个出现的字，直到把他们连成一个句子。

**Feature**

共有六篇文章/小说作为演示文档，其中三篇中文文章，四篇英文文章

用户可以上传.txt文件，在线进行文章解析和数据集的生成

用户可以选定生成的文章句子数量

**Usage**

要使用该程序，讲文件克隆到本地后，使用该命令，配置好环境

pip install -r requirements.txt

使用该命令进入演示网页

python main.py

在网页中，你可以在表单中选择我们提供的演示文章，或者自己上传本地的.txt文件进行解析。接着填写需要生成的文章的句数，点击生成即可生成由马尔可夫链随机生成的文章。

# 小结

随着这短暂而充实的一周接近尾声，我们的数据结构实验课告一段落。在此次实验中，我们小组选定了基于马尔可夫链的写作机器人这一课题，以展开我们算法拓展学习与训练。该写作机器人基于马尔可夫链设计，利用Flask框架在前端进行演示。马尔可夫链是随机过程中的概念，基于该算法可以根据已有的文本生成一些看似真实的文字，每生成一个词，它可以根据下一个词出现的频数随机选择下一个出现的字，直到把他们连成一个句子。

在实践过程中，我们主要遇到的困难及解决过程如下：

前端部分看似简单，但为了保证设计在细节上的完美我们下了很大的功夫。包括实现居中、对齐、字号调节、图形大小调节、模板设计以及基于需求的功能模块构图的设计等。由于大家相对来说都对前端部分不算太熟悉，我们查阅了大量的电子资料、参考了一些排版合理的代码模板等，经过多次改版，最终制成了我们最终的前端页面。

后端部分基于Python的算法功能实现，则是我们这一课题的核心内容。考虑到中文分词的困难性，我们小组首先尝试实现了英文的文本生成功能。这一过程的主要困难在于如何依据标点进行分词、提高文本生成的语意连贯性、在句首大写、在句末合理添加标点符号等。在基本实现英文段落生成的功能后，我们也在原代码的基础上进行改善，尝试进行了中文单字分词的文本生成代码设计，后续也使用第三方库jieba对中文部分的文本分词进行了完善。此外，在完成了对本地文本文件的解析之后，我们也增设了对用户上传的文本文件或文本段落的解析。

在Web框架的搭建方面，最初我们考虑使用Django来实现。但在多日的学习之后发现，Django框架的大部分功能对于我们的项目目标来说稍显冗余，且实现过程较为复杂。因此，在讨论之后我们改用了Flask进行框架搭建。不仅满足了我们现阶段的项目需求，更简化了程序代码。

通过此次实验，我们更是深刻体会到了小组合作的重要性。为了完成一份完整的作品，我们需要实现后端、前端以及Web框架的搭建。如果仅靠一个人的力量，是很难在短时间内得到一份相对满意的成果的。然而，通过小组内的合理分工，我们每个人都可以高效地实现各自手中负责地功能实现，并在多方思维碰撞的过程中精益求精，在成果的细节展示上以期得到更好的效果。同时，我们也利用了GitHub以尽可能高效地实现多方协作项目开发。

在这次的实验课题项目设计中我们又学到了许多知识，不仅仅是编程语言的知识，更多的是我们在学习过程中掌握的一些学习、制作项目与测试的简易技巧、手段和方法。虽然我们的成果还不是很完美，但不可否认的是，这一过程让我们收获良多，更启发我们在未来继续拓展学习课外知识，以实现自我提升。

# 附录1 工作量及成绩自评表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **班级** | **学号** | **姓名** | **工作量简述**  **（代码量）** | **成绩自评** |
| 计科A2001 | 2019080184 | 邓岳 |  | A+ |
| 计科A2001 | 2020020360 | 熊圳 |  | A+ |
| 计科A2001 | 2019010025 | 赵千慧 |  | A+ |
| 计科A2001 | 2017060315 | 赵凯 |  | A+ |
| 备注：  　1、成绩自凭栏可选填：A+、A、A-、B+、B、B-、C+、C、C-、D、F | | | | |

# 附录2：程序代码（仅电子版需要、纸质版不需要）

./dao/analysis.py

1. #!/usr/bin/env python
2. # -\*- coding: utf-8 -\*-
3. # @Time    : 2022-01-02 12:09
4. # @Author  : Dysprosium
5. # @File    : analysis.py
7. **import** re
8. **import** jieba

11. **class** Analysis:
13. **def** \_\_init\_\_(self, string):
14. self.string = string
15. self.key\_len = 2
16. self.sentences = []
18. **def** split\_string\_into\_sentences(self):
19. text = self.string
20. # with open(self.txt, "r", encoding='utf-8') as f:
21. #     for line in f:
22. #         if line == '\n' or line.startswith('Page '):
23. #             continue
24. #
25. #         text += line.rstrip() + ' '
27. # Filtered = filter(None, re.split("\“|\” |; |, |\? |! |\. |\.\n", text))
28. Filtered = filter(None, re.split("\“|\” |; |! |\. |\n |，|。|；|, |：|: |\?|” ", text))
29. sentences = list(Filtered)
30. self.sentences = sentences
31. **return** None
33. **def** split\_text\_into\_sentences(self):
34. lines = []
35. with open(self.string, "r", encoding='utf-8') as f:
36. **for** line **in** f:
37. **if** line == '\n' **or** line.startswith('Page '):
38. **continue**
39. lines.append(line)
41. text = ' '.join(lines).replace('\n', '')
42. # Filtered = filter(None, re.split("\“|\” |; |, |\? |! |\. |\.\n", text))
43. # Filtered = filter(None, re.split("\“|\” |; |! |\. |，|。|；", text))
44. Filtered = filter(None, re.split("\“|\” |; |! |\. |\n |，|。|；|, |：|: |\?|” ", text))
45. sentences = list(Filtered)
46. self.sentences = sentences
47. **return** None
49. **def** prepare\_dataset(self, sentences):
50. dataset = {}
51. sent\_head = {}
52. **if** sentences == None:
53. sentences = self.sentences
55. LANG = 'en'
56. **for** \_char **in** sentences[0]:
57. **if** '\u4e00' <= \_char <= '\u9fa5':
58. LANG = 'ch'
60. **for** sentence **in** sentences:
61. sentence = '^ ' + sentence + ' $'
62. **if** LANG == 'en':
63. words\_list = sentence.split(' ')
64. **else**:
65. words\_list = list(jieba.cut(sentence))
67. keys = words\_list[:self.key\_len]
69. **if** keys[1] **in** sent\_head.keys():
70. sent\_head[keys[1]] += 1
71. **else**:
72. sent\_head[keys[1]] = 1
74. **for** word **in** words\_list[self.key\_len:]:
75. # print(p, q)
76. **if** tuple(keys) **in** dataset.keys():
77. **if** word **in** dataset[tuple(keys)]:
78. dataset[tuple(keys)][word] += 1
79. **else**:
80. dataset[tuple(keys)][word] = 1
81. **else**:
82. dataset[tuple(keys)] = {}
83. dataset[tuple(keys)][word] = 1
84. keys.pop(0)
85. keys.append(word)
86. **return** dataset, sent\_head, LANG

89. **if** \_\_name\_\_ == '\_\_main\_\_':
90. txt = input("Please enter the text file to be analyzed:")
91. analysis = Analysis(txt)
92. sentences = analysis.split\_text\_into\_sentences()
93. **print**(sentences)

./dao/generator.py

1. #!/usr/bin/env python
2. # -\*- coding: utf-8 -\*-
3. # @Time    : 2021-12-30 21:28
4. # @Author  : Dysprosium
5. # @File    : generator.py
6. **from** random **import** randint
7. **import** re

10. # from analysis import Analysis
12. **class** Generator:
13. **def** \_\_init\_\_(self, dataset, head):
14. self.dataset = dataset
15. self.key\_len = 2
16. self.sent\_head = head
18. **def** get\_next\_word(self, word):
19. sum\_of\_candidates = 0
20. # number of frequencies
21. # for keys in self.dataset[tuple(word)].keys():
22. #     sum\_of\_candidates += self.dataset[tuple(word)][keys]
23. sum\_of\_candidates = sum(self.dataset[tuple(word)].values())
24. random = randint(1, sum\_of\_candidates)
25. t = 0
26. # select a number from candidates randomly
27. **for** w **in** self.dataset[tuple(word)].keys():
28. t += self.dataset[tuple(word)][w]
29. **if** random <= t:
30. **return** w
32. **def** generate(self, word\_separator):
33. sentences = ['^']
34. r = randint(1, sum(self.sent\_head.values()))
35. **for** item **in** self.sent\_head.items():
36. **if** r >= item[1]:
37. r -= item[1]
38. **else**:
39. sentences.append(item[0])
40. **break**
42. **while** True:
43. end = sentences[-self.key\_len:]
44. next\_word = self.get\_next\_word(end)
45. sentences.append(next\_word)
46. **if** next\_word == '$':
47. **break**
48. **return** word\_separator.join(sentences[1:-1])

51. **if** \_\_name\_\_ == '\_\_main\_\_':
52. txt = input("Please enter the text file to be analyzed:")
53. analysis = Analysis(txt)
54. sentences = analysis.split\_text\_into\_sentences()
55. dataset, head = analysis.prepare\_dataset(sentences)
56. g = Generator(dataset, head)
57. number = 1
58. result = ''
59. **for** i **in** range(0, number):
60. result += g.generate(' ') + '\n'
62. **print**(result)

./service/article\_service.py

1. #!/usr/bin/env python
2. # -\*- coding: utf-8 -\*-
3. # @Time    : 2022-01-05 20:35
4. # @Author  : Dysprosium
5. # @File    : article\_service.py
7. **from** dao.generator **import** Generator
8. **from** dao.analysis **import** Analysis

11. **class** ArticleService:
13. **def** \_\_init\_\_(self, theme, number):
14. self.theme = theme
15. self.number = number
17. **def** generate\_text(self):
18. """
19. 根据构造的TextSerive对象生成文本
20. :return: 生成的文本
21. """
22. theme = self.theme
23. number = self.number
24. analysis = None
25. **if** theme == '1':
26. analysis = Analysis("./data/huozhe.txt")
27. **elif** theme == '2':
28. analysis = Analysis("./data/yueliang.txt")
29. **elif** theme == '3':
30. analysis = Analysis("./data/hp.txt")
31. **elif** theme == '4':
32. analysis = Analysis("./data/hhgttg.txt")
33. **elif** theme == '5':
34. analysis = Analysis("./data/aiwl.txt")
35. **elif** theme == '6':
36. analysis = Analysis("./data/omas.txt")
37. **elif** theme == '7':
38. analysis = Analysis("./data/sh.txt")
39. sentences = analysis.split\_text\_into\_sentences()
40. dataset, head, LANG = analysis.prepare\_dataset(sentences)
41. # create a generator
42. g = Generator(dataset, head)
43. result = ""
44. **if** theme == '1':
45. **for** i **in** range(0, number):
46. result += g.generate('') + '。'
47. **elif** theme == '2':
48. **for** i **in** range(0, number):
49. result += g.generate('') + '。'
50. **else**:
51. **for** i **in** range(0, number):
52. result += g.generate(' ') + '. '
53. **return** result

./service/text\_service.py

1. #!/usr/bin/env python
2. # -\*- coding: utf-8 -\*-
3. # @Time    : 2022-01-05 20:41
4. # @Author  : Dysprosium
5. # @File    : text\_serivce.py

8. **from** dao.generator **import** Generator
9. **from** dao.analysis **import** Analysis

12. **class** TextService:
13. **def** \_\_init\_\_(self, number):
14. self.number = number
16. **def** generate\_text(self, text):
17. """
18. 根据上传的文件解析后生成文本
19. :param text: 上传的文本
20. :return: 生成的文本
21. """
22. number = self.number
23. analysis = Analysis(text)
24. sentences = analysis.split\_string\_into\_sentences()
25. dataset, head, LANG = analysis.prepare\_dataset(sentences)
26. # create a generator
27. g = Generator(dataset, head)
28. result = ""
29. **if** LANG == 'ch':
30. **for** i **in** range(0, number):
31. result += g.generate('') + '。'
32. **else**:
33. **for** i **in** range(0, number):
34. result += g.generate(' ') + '. '
35. **return** result

main.py

1. **from** flask **import** Flask, url\_for, request, render\_template, redirect
2. **from** service.article\_service **import** ArticleService
3. **from** service.text\_serivce **import** TextService
5. app = Flask(\_\_name\_\_, static\_url\_path='/static')

8. @app.route('/')
9. **def** index():
10. **return** redirect(url\_for('login'))

13. @app.route('/login', methods=['POST', 'GET'])
14. **def** login():
15. **if** request.method == 'GET':
16. **return** render\_template('login.html', image="../logo/markov\_grey.png")
17. **else**:
18. theme = request.form.get('theme')
19. number = int(request.form.get('number'))
20. **if** request.files["myfile"].filename == '':
21. article\_service = ArticleService(theme, number)
22. result = article\_service.generate\_text()
23. **else**:
24. **try**:
25. file = request.files["myfile"]
26. str = bytes.decode(file.read(), encoding="utf-8")
27. text\_service = TextService(number)
28. result = text\_service.generate\_text(str)
29. **except** Exception:
30. **return** render\_template('login.html', str="请上传.txt文件")
32. **return** render\_template('login.html', str=result)

35. @app.route('/loginen', methods=['POST', 'GET'])
36. **def** loginen():
37. **if** request.method == 'GET':
38. **return** render\_template('loginen.html')
39. **elif** request.method == 'POST':
40. **print**(request.form.get('theme'))
41. theme = request.form.get('theme')
42. number = int(request.form.get('number'))
43. **if** request.files["myfile"].filename == '':
44. article\_service = ArticleService(theme, number)
45. result = article\_service.generate\_text()
46. **else**:
47. **try**:
48. file = request.files["myfile"]
49. str = bytes.decode(file.read(), encoding="utf-8")
50. **except** UnicodeDecodeError:
51. **return** render\_template('loginen.html', str="请上传.txt文件")
52. text\_service = TextService(number)
53. result = text\_service.generate\_text(str)
54. **return** render\_template('loginen.html', str=result)

57. @app.route('/github', methods=['POST', 'GET'])
58. **def** github():
59. **if** request.method == 'GET':
60. **return** redirect("https://github.com/Dysprosium0626/Markov\_bot")


64. **if** \_\_name\_\_ == "\_\_main\_\_":
65. app.run(debug=True)

login.html

1. **<html>**
2. **<head>**
3. **<title>**Marakov Gen
4. **</title>**
5. **<meta** http-equiv="content-type" content="txt/html; charset=utf-8"**/>**
6. **<meta** name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0"**>**
7. **</head>**
8. **<body>**
9. **<div** id="主体"**>**
10. **<span** id="slogan1"**>**MARKOV **</span>**
11. **<span** id="slogan2"**>**GEN **</span>**
12. **<br>**
13. **<div** id="标题"**>**基于马尔可夫链的写作机器人**</div>**
14. **<img** id="logo" src="{{ url\_for('static', filename='markov\_gray.png')}}"**/>**
15. **<div** id="intro"**><b>**Markov Gen**</b>**是一个基于**<b>**马尔可夫链**</b>**的写作机器人，选择要生成的演示文章，或自行上传**<b>**.txt文件**</b>**，**<br>**输入要生成的句子数量，点击生成，
16. 就可以根据马尔可夫链生成一些**<b>**看似真实的句子**</b>**。**</div>**
17. **<div** id="表单"**>**
18. **<form** action="/login" method="post" enctype="multipart/form-data"**>**
19. **<div** id="选择文章"**>**
20. **<div** id="提示"**>**选择文章**</div>**
21. **<select** name="theme" class="select"**>**
22. **<option** value="1"**>**活着**</option>**
23. **<option** value="2"**>**月亮与六便士**</option>**
24. **</select>**
25. **</div>**
26. **<div** id="或"**>**
27. 或
28. **</div>**
29. **<div** id="上传文章"**>**
30. **<div** id="提示"**>**上传文章**</div>**
31. **<div** class="file"**>**
32. 上传文章
33. **<input** type="file" name="myfile" accept="text/plain" id="upload"**></input>**
34. **</div>**
35. **</div>**
36. **<div** class="group" id="句数"**>**
37. **<div** id="提示"**>**文章句数**</div>**
38. **<input** type="number" min="0" max="1000" value="10" class="input" name="number"**/>**
39. **</div>**
40. **<div** class="btn-box" id="生成"**>**
41. **<button** type="submit" class="button"**>**生成**</button>**
42. **</div>**
43. **</form>**
44. **</div>**
45. **<div** id="文章"**>**
46. {{str}}
47. **</div>**
48. **<div** id="英文"**>**
49. **<a** href={{url\_for('loginen')}} class="a"**>**English**</a>**
50. **</div>**
51. **<div** id="github"**>**
52. **<a** href={{url\_for('github')}} class="a"**>**Github**</a>**
53. **</div>**
54. **</div>**
56. **</body>**
57. **</html>**
59. **<style>**
60. body {
61. **background-color**: rgb(243, 243, 243);
62. }
63. #slogan1 {
64. **font-weight**: normal;
65. **font-size**: 72px;
66. **font-family**: "Icons20", sans-serif;
67. }
68. #slogan2 {
69. **font-weight**: bolder;
70. **font-size**: 72px;
71. **font-family**: "Icons16", sans-serif;
72. }
73. #主体 {
74. **border**: 2px solid #0c0c0c;
75. **text-align**: center;
76. **margin**: 0 auto;
77. }
79. #标题 {
80. **font-size**: 48px;
81. **font-weight**: 1000;
82. **font-family**: "Microsoft YaHei UI";
83. **color**: #0d0f0d;
84. **margin-bottom**: 20px;
85. }
87. #logo {
88. **margin-bottom**: 20px;
89. }
91. #英文 {
92. **font-family**: "Icons16", sans-serif;
93. **font-weight**: 100;
94. **text-align**: end;
95. **margin-right**: 10px;
96. **margin-bottom**: 1px;
97. **font-size**: 18px;
98. }
100. #github {
101. **font-family**: "Icons16", sans-serif;
102. **font-weight**: 100;
103. **text-align**: end;
104. **margin-right**: 10px;
105. **margin-bottom**: 1px;
106. **font-size**: 18px;
107. }
109. .a {
110. **font-style**: normal;
111. **color**: #000;
112. }
114. #表单 {
115. **margin**: 0 auto;
116. **text-align**: center;
117. **padding**: 10px;
118. **height**: 200px;
119. **width**: fit-content;
120. **font-family**: "Microsoft YaHei UI";
121. }
123. #提示 {
124. **font-size**: 24px;
125. **font-weight**: 400;
126. **margin-bottom**: 20px;
127. }
129. #选择文章 {
130. **float**: left;
131. **width**: 200px;
132. **padding**: 25px;
133. **text-align**: center;
134. }
136. #或 {
137. **float**: left;
138. **font-size**: 24px;
139. **line-height**: 60px;
140. **padding-bottom**: 25px;
141. **margin-top**: 77px;
142. }
144. #上传文章 {
145. **float**: left;
146. **width**: 200px;
147. **padding**: 25px;
148. **text-align**: center;
149. **position**: relative;
150. **overflow**: hidden;
151. }
153. #句数 {
154. **float**: left;
155. **width**: 200px;
156. **padding**: 25px;
157. **text-align**: center;
158. }
160. #生成 {
161. **float**: left;
162. **width**: 180px;
163. **height**: 60px;
164. **padding**: 25px;
165. **margin-top**: 52px;
166. }
168. #文章 {
169. **font-size**: 18px;
170. /\* border: 2px dashed #0c0c0c ; \*/
171. **text-align**: center;
172. **text-indent**: 2em;
173. **margin**: 5px 150px 20px 150px;
174. }
176. .file {
177. **position**: relative;
178. **display**: inline-block;
179. -webkit-appearance: none;
180. -moz-appearance: none;
181. appearance: none;
182. **padding**: 18px 54px;
183. **font-size**: 18px;
184. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;
185. **color**: rgba(48, 52, 84, 1);
186. **border**: none;
187. border-radius: 16px;
188. **background-color**: rgba(238, 240, 245, 1);
189. **cursor**: pointer;
190. **outline**: none;
191. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1);
192. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -2px -2px 2px rgba(255, 255, 255, 0.9), -4px -4px 6px rgba(255, 255, 255, 0.9), -8px -8px 16px rgba(255, 255, 255, 0.9), -12px -12px 24px rgba(255, 255, 255, 0.9), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 2px 2px 2px rgba(12, 13, 18, 0.08), 4px 4px 6px rgba(12, 13, 18, 0.08), 8px 8px 16px rgba(12, 13, 18, 0.08), 12px 12px 24px rgba(12, 13, 18, 0.08);
193. -webkit-transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
194. transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
195. }
197. .file input {
198. **position**: absolute;
199. **font-size**: 50px;
200. right: 0;
201. top: 0;
202. opacity: 0;
203. }
205. .file:hover {
206. **color**: rgba(0, 80, 220, 1);
207. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1), 0 0 1px rgba(255, 255, 255, 0.7), 0 0 4px rgba(0, 150, 255, 0.08), 0 0 8px rgba(0, 150, 255, 0.08);
208. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -3px -3px 3px rgba(255, 255, 255, 1), -5px -5px 6px rgba(255, 255, 255, 1), -10px -10px 16px rgba(255, 255, 255, 1), -14px -14px 24px rgba(255, 255, 255, 1), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 3px 3px 3px rgba(12, 13, 18, 0.085), 5px 5px 6px rgba(12, 13, 18, 0.085), 10px 10px 16px rgba(12, 13, 18, 0.085), 14px 14px 24px rgba(12, 13, 18, 0.085);
209. }
211. .file:active {
212. box-shadow: 1px 1px 1px rgba(255, 255, 255, 1), inset -2px -2px 2px rgba(255, 255, 255, 0.9), inset -3px -3px 4px rgba(255, 255, 255, 0.9), inset -6px -6px 12px rgba(255, 255, 255, 0.9), inset -8px -8px 16px rgba(255, 255, 255, 0.9), -1px -1px 1px rgba(12, 13, 18, 0.06), inset 2px 2px 2px rgba(12, 13, 18, 0.08), inset 3px 3px 4px rgba(12, 13, 18, 0.08), inset 6px 6px 12px rgba(12, 13, 18, 0.08), inset 8px 8px 16px rgba(12, 13, 18, 0.08);
213. }
215. .button {
216. **display**: inline-block;
217. -webkit-appearance: none;
218. -moz-appearance: none;
219. appearance: none;
220. **padding**: 18px 72px;
221. **font-size**: 18px;
222. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;;
223. **color**: rgba(48, 52, 84, 1);
224. **border**: none;
225. border-radius: 16px;
226. **background-color**: rgba(238, 240, 245, 1);
227. **cursor**: pointer;
228. **outline**: none;
229. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1);
230. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -2px -2px 2px rgba(255, 255, 255, 0.9), -4px -4px 6px rgba(255, 255, 255, 0.9), -8px -8px 16px rgba(255, 255, 255, 0.9), -12px -12px 24px rgba(255, 255, 255, 0.9), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 2px 2px 2px rgba(12, 13, 18, 0.08), 4px 4px 6px rgba(12, 13, 18, 0.08), 8px 8px 16px rgba(12, 13, 18, 0.08), 12px 12px 24px rgba(12, 13, 18, 0.08);
231. -webkit-transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
232. transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
233. }
235. .button:hover {
236. **color**: rgba(0, 80, 220, 1);
237. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1), 0 0 1px rgba(255, 255, 255, 0.7), 0 0 4px rgba(0, 150, 255, 0.08), 0 0 8px rgba(0, 150, 255, 0.08);
238. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -3px -3px 3px rgba(255, 255, 255, 1), -5px -5px 6px rgba(255, 255, 255, 1), -10px -10px 16px rgba(255, 255, 255, 1), -14px -14px 24px rgba(255, 255, 255, 1), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 3px 3px 3px rgba(12, 13, 18, 0.085), 5px 5px 6px rgba(12, 13, 18, 0.085), 10px 10px 16px rgba(12, 13, 18, 0.085), 14px 14px 24px rgba(12, 13, 18, 0.085);
239. }
241. .button:active {
242. box-shadow: 1px 1px 1px rgba(255, 255, 255, 1), inset -2px -2px 2px rgba(255, 255, 255, 0.9), inset -3px -3px 4px rgba(255, 255, 255, 0.9), inset -6px -6px 12px rgba(255, 255, 255, 0.9), inset -8px -8px 16px rgba(255, 255, 255, 0.9), -1px -1px 1px rgba(12, 13, 18, 0.06), inset 2px 2px 2px rgba(12, 13, 18, 0.08), inset 3px 3px 4px rgba(12, 13, 18, 0.08), inset 6px 6px 12px rgba(12, 13, 18, 0.08), inset 8px 8px 16px rgba(12, 13, 18, 0.08);
243. }
245. .input {
246. **display**: block;
247. **margin**: 0 auto;
248. **height**: 60px;
249. **padding**: 0 24px;
250. box-sizing: border-box;
251. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;
252. **font-size**: 18px;
253. **line-height**: 60px;
254. border-radius: 16px;
255. **border**: none;
256. **color**: rgb(48, 52, 84);
257. **background**: rgba(240, 242, 247, 1);
258. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 0.4), inset -1px -1px 1px rgba(12, 13, 18, 0.04), inset 0 0 0 2px rgba(238, 240, 245, 1), inset -2px -2px 2px 2px rgba(255, 255, 255, 0.4), inset -4px -4px 4px 2px rgba(255, 255, 255, 0.4), -1px -1px 4px 0px rgba(255, 255, 255, 0.4), -2px -2px 8px 0px rgba(255, 255, 255, 0.4), inset 2px 2px 2px 2px rgba(12, 13, 18, 0.04), inset 4px 4px 4px 2px rgba(12, 13, 18, 0.04), 1px 1px 4px 0px rgba(12, 13, 18, 0.04), 2px 2px 8px 0px rgba(12, 13, 18, 0.04);
259. }
261. .input[type="number"] {
262. **width**: 180;
263. **font-family**: digital;
264. **font-size**: 18px;
265. **line-height**: 18px;
266. **text-align**: center;
267. }
269. .input:focus {
270. **outline**: none;
271. }
273. .select {
274. **position**: relative;
275. **height**: 60px;
276. **padding**: 0 48px 0 24px;
277. box-sizing: border-box;
278. **font-family**: 'Montserrat';
279. **border**: 0;
280. border-radius: 16px;
281. **background-color**: rgb(238, 240, 245);
282. **background-image**: url('https://s3-us-west-2.amazonaws.com/s.cdpn.io/1329536/icon\_select\_s.svg');
283. **background-position**: calc(100% - 16px) center;
284. **background-repeat**: no-repeat;
285. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 0.6), inset -1px -1px 1px rgba(12, 13, 18, 0.06), -1px -1px 4px 0px rgba(255, 255, 255, 0.6), -2px -2px 8px 0px rgba(255, 255, 255, 0.6), 1px 1px 4px 0px rgba(12, 13, 18, 0.06), 2px 2px 8px 0px rgba(12, 13, 18, 0.06);
286. **font-size**: 18px;
287. **line-height**: 60px;
288. -webkit-appearance: none;
289. -moz-appearance: none;
290. appearance: none;
291. }
293. .select:focus {
294. **outline**: none;
295. }
296. **</style>**

loginen.html

1. **<html>**
2. **<head>**
3. **<title>**Marakov Gen
4. **</title>**
5. **<meta** http-equiv="content-type" content="txt/html; charset=utf-8"**/>**
6. **<meta** name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0"**>**
7. **</head>**
8. **<body>**
9. **<div** id="主体"**>**
10. **<span** id="slogan1"**>**MARKOV **</span>**
11. **<span** id="slogan2"**>**GEN **</span>**
12. **<div** id="标题"**>**A Writing bot based on **<b>**Markov Chain**</b></div>**
13. **<img** id="logo" src="{{ url\_for('static', filename='markov\_gray.png')}}"**/>**
14. **<div** id="intro"**><b>**Markov Gen**</b>**is a writing bot based on**<b>**Markov Chain**</b>**.Please choose an article or upload**<b>**.txt file**</b>**by yourself.**<br>**
15. Enter the number of sentences you want to generate and press **<b>**Generate**</b>**. Then our bot can generate some **<b>**seemingly realistic sentences**</b>**.**</div>**


19. **<div** id="表单"**>**
20. **<form** action="loginen" method="post" enctype="multipart/form-data"**>**
21. **<div** id="选择文章"**>**
22. **<div** id="提示"**>**Choose An article**</div>**
23. **<select** name="theme" class="select"**>**
24. **<option** value="3"**>**Harry Potter**</option>**
25. **<option** value="4"**>**The Hitchhiker's Guide to the Galaxy**</option>**
26. **<option** value="5"**>**Alice in Wonderland**</option>**
27. **<option** value="6"**>**The Old Man and Sea**</option>**
28. **<option** value="7"**>**Sherlock Holmes**</option>**
29. **</select>**
30. **</div>**
31. **<div** id="或"**>**
32. OR
33. **</div>**
34. **<div** id="上传文章"**>**
35. **<div** id="提示"**>**Upload An article**</div>**
36. **<div** class="file"**>**
37. Upload
38. **<input** type="file" name="myfile" accept="text/plain" id="upload"**></input>**
39. **</div>**
40. **</div>**
41. **<div** class="group" id="句数"**>**
42. **<div** id="提示"**>**Number of sentences**</div>**
43. **<input** type="number" min="0" max="1000" value="10" class="input" name="number"**/>**
44. **</div>**
45. **<div** class="btn-box" id="生成"**>**
46. **<button** type="submit" class="button"**>**Generate**</button>**
47. **</div>**
48. **</form>**
49. **</div>**
50. **<div** id="文章"**>**
51. {{str}}
52. **</div>**
53. **<div** id="英文"**>**
54. **<a** href={{url\_for('login')}} class="a"**>**简体中文**</a>**
55. **</div>**
56. **<div** id="github"**>**
57. **<a** href={{url\_for('github')}} class="a"**>**Github**</a>**
58. **</div>**
59. **</div>**
60. **</body>**
61. **</html>**
62. **<style>**  a
63. body {
64. **background-color**: rgb(243, 243, 243);
65. }
66. #slogan1 {
67. **font-weight**: normal;
68. **font-size**: 72px;
69. **font-family**: "Icons20", sans-serif;
70. }
71. #slogan2 {
72. **font-weight**: bolder;
73. **font-size**: 72px;
74. **font-family**: "Icons16", sans-serif;
75. }
76. #主体 {
77. **border**: 2px solid #0c0c0c;
78. **text-align**: center;
79. **margin**: 0 auto;
80. }
82. #标题 {
83. **font-size**: 48px;
84. **font-weight**: normal;
85. **font-family**: "Helvetica Neue", Helvetica, Arial, sans-serif;
86. **color**: #0d0f0d;
87. **margin-bottom**: 20px;
88. }
89. #logo {
90. **margin-bottom**: 20px;
91. }
92. #英文 {
93. **font-family**: "Icons16", sans-serif;
94. **font-weight**: 100;
95. **text-align**: end;
96. **margin-right**: 10px;
97. **margin-bottom**: 1px;
98. **font-size**: 18px;
99. }
100. #github {
101. **font-family**: "Icons16", sans-serif;
102. **font-weight**: 100;
103. **text-align**: end;
104. **margin-right**: 10px;
105. **margin-bottom**: 1px;
106. **font-size**: 18px;
107. }
108. .a {
109. **font-style**: normal;
110. **color**: #000;
111. }
113. #表单 {
114. **margin**: 0 auto;
115. **text-align**: center;
116. **padding**: 10px;
117. **height**: 200px;
118. **width**: fit-content;
119. **font-family**: "Helvetica Neue", Helvetica, Arial, sans-serif;
120. }
122. #提示 {
123. **font-size**: 24px;
124. **font-weight**: 400;
125. **margin-bottom**: 20px;
126. }
128. #选择文章 {
129. **float**: left;
130. **padding**: 25px;
131. **text-align**: center;
133. }
135. #或 {
136. **float**: left;
137. **font-size**: 24px;
138. **line-height**: 60px;
139. **padding-bottom**: 25px;
140. **margin-top**: 77px;
141. }
143. #上传文章 {
144. **float**: left;
145. **padding**: 25px;
146. **text-align**: center;
147. **position**: relative;
148. **overflow**: hidden;
149. }
151. #句数 {
152. **float**: left;
153. **padding**: 25px;
154. **text-align**: center;
155. }
157. #生成 {
158. **float**: left;
159. **width**: 180px;
160. **height**: 60px;
161. **padding**: 25px;
162. **margin-top**: 52px;
163. }
165. #文章 {
166. **font-size**: 18px;
167. /\* border: 2px dashed #0c0c0c ; \*/
168. **text-align**: center;
169. **text-indent**: 2em;
170. **margin**: 5px 150px 20px 150px;
171. }
173. .file {
174. **position**: relative;
175. **display**: inline-block;
176. -webkit-appearance: none;
177. -moz-appearance: none;
178. appearance: none;
179. **padding**: 19.6px 61.475px;
180. **font-size**: 18px;
181. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;;
182. **color**: rgba(48, 52, 84, 1);
183. **border**: none;
184. border-radius: 16px;
185. **background-color**: rgba(238, 240, 245, 1);
186. **cursor**: pointer;
187. **outline**: none;
188. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1);
189. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -2px -2px 2px rgba(255, 255, 255, 0.9), -4px -4px 6px rgba(255, 255, 255, 0.9), -8px -8px 16px rgba(255, 255, 255, 0.9), -12px -12px 24px rgba(255, 255, 255, 0.9), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 2px 2px 2px rgba(12, 13, 18, 0.08), 4px 4px 6px rgba(12, 13, 18, 0.08), 8px 8px 16px rgba(12, 13, 18, 0.08), 12px 12px 24px rgba(12, 13, 18, 0.08);
190. -webkit-transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
191. transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
192. }
194. .file input {
195. **position**: absolute;
196. **font-size**: 50px;
197. right: 0;
198. top: 0;
199. opacity: 0;
200. }
202. .file:hover {
203. **color**: rgba(0, 80, 220, 1);
204. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1), 0 0 1px rgba(255, 255, 255, 0.7), 0 0 4px rgba(0, 150, 255, 0.08), 0 0 8px rgba(0, 150, 255, 0.08);
205. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -3px -3px 3px rgba(255, 255, 255, 1), -5px -5px 6px rgba(255, 255, 255, 1), -10px -10px 16px rgba(255, 255, 255, 1), -14px -14px 24px rgba(255, 255, 255, 1), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 3px 3px 3px rgba(12, 13, 18, 0.085), 5px 5px 6px rgba(12, 13, 18, 0.085), 10px 10px 16px rgba(12, 13, 18, 0.085), 14px 14px 24px rgba(12, 13, 18, 0.085);
206. }
208. .file:active {
209. box-shadow: 1px 1px 1px rgba(255, 255, 255, 1), inset -2px -2px 2px rgba(255, 255, 255, 0.9), inset -3px -3px 4px rgba(255, 255, 255, 0.9), inset -6px -6px 12px rgba(255, 255, 255, 0.9), inset -8px -8px 16px rgba(255, 255, 255, 0.9), -1px -1px 1px rgba(12, 13, 18, 0.06), inset 2px 2px 2px rgba(12, 13, 18, 0.08), inset 3px 3px 4px rgba(12, 13, 18, 0.08), inset 6px 6px 12px rgba(12, 13, 18, 0.08), inset 8px 8px 16px rgba(12, 13, 18, 0.08);
210. }
212. .button {
213. **display**: inline-block;
214. -webkit-appearance: none;
215. -moz-appearance: none;
216. appearance: none;
217. **padding**: 19.6px 52.475px;
218. **font-size**: 18px;
219. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;;
220. **color**: rgba(48, 52, 84, 1);
221. **border**: none;
222. border-radius: 16px;
223. **background-color**: rgba(238, 240, 245, 1);
224. **cursor**: pointer;
225. **outline**: none;
226. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1);
227. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -2px -2px 2px rgba(255, 255, 255, 0.9), -4px -4px 6px rgba(255, 255, 255, 0.9), -8px -8px 16px rgba(255, 255, 255, 0.9), -12px -12px 24px rgba(255, 255, 255, 0.9), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 2px 2px 2px rgba(12, 13, 18, 0.08), 4px 4px 6px rgba(12, 13, 18, 0.08), 8px 8px 16px rgba(12, 13, 18, 0.08), 12px 12px 24px rgba(12, 13, 18, 0.08);
228. -webkit-transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
229. transition: color 0.3s, text-shadow 0.3s, box-shadow 0.6s;
230. }
232. .button:hover {
233. **color**: rgba(0, 80, 220, 1);
234. **text-shadow**: 0 -1px 0 rgba(12, 13, 18, 0.08), 0 1px 0 rgba(255, 255, 255, 1), 0 0 1px rgba(255, 255, 255, 0.7), 0 0 4px rgba(0, 150, 255, 0.08), 0 0 8px rgba(0, 150, 255, 0.08);
235. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 1), -3px -3px 3px rgba(255, 255, 255, 1), -5px -5px 6px rgba(255, 255, 255, 1), -10px -10px 16px rgba(255, 255, 255, 1), -14px -14px 24px rgba(255, 255, 255, 1), inset -1px -1px 1px rgba(12, 13, 18, 0.06), 3px 3px 3px rgba(12, 13, 18, 0.085), 5px 5px 6px rgba(12, 13, 18, 0.085), 10px 10px 16px rgba(12, 13, 18, 0.085), 14px 14px 24px rgba(12, 13, 18, 0.085);
236. }
238. .button:active {
239. box-shadow: 1px 1px 1px rgba(255, 255, 255, 1), inset -2px -2px 2px rgba(255, 255, 255, 0.9), inset -3px -3px 4px rgba(255, 255, 255, 0.9), inset -6px -6px 12px rgba(255, 255, 255, 0.9), inset -8px -8px 16px rgba(255, 255, 255, 0.9), -1px -1px 1px rgba(12, 13, 18, 0.06), inset 2px 2px 2px rgba(12, 13, 18, 0.08), inset 3px 3px 4px rgba(12, 13, 18, 0.08), inset 6px 6px 12px rgba(12, 13, 18, 0.08), inset 8px 8px 16px rgba(12, 13, 18, 0.08);
240. }
242. .input {
243. **display**: block;
244. **margin**: 0 auto;
245. **height**: 60px;
246. **padding**: 0 24px;
247. box-sizing: border-box;
248. **font-family**: 'Montserrat', 'Helvetica', 'Helvetica Neue', 'Segoe UI', 'Arial', sans-serif;
249. **font-size**: 18px;
250. **line-height**: 60px;
251. border-radius: 16px;
252. **border**: none;
253. **color**: rgb(48, 52, 84);
254. **background**: rgba(240, 242, 247, 1);
255. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 0.4), inset -1px -1px 1px rgba(12, 13, 18, 0.04), inset 0 0 0 2px rgba(238, 240, 245, 1), inset -2px -2px 2px 2px rgba(255, 255, 255, 0.4), inset -4px -4px 4px 2px rgba(255, 255, 255, 0.4), -1px -1px 4px 0px rgba(255, 255, 255, 0.4), -2px -2px 8px 0px rgba(255, 255, 255, 0.4), inset 2px 2px 2px 2px rgba(12, 13, 18, 0.04), inset 4px 4px 4px 2px rgba(12, 13, 18, 0.04), 1px 1px 4px 0px rgba(12, 13, 18, 0.04), 2px 2px 8px 0px rgba(12, 13, 18, 0.04);
256. }
258. .input[type="number"] {
259. **width**: 180;
260. **font-family**: digital;
261. **font-size**: 18px;
262. **line-height**: 18px;
263. **text-align**: center;
264. }
266. .input:focus {
267. **outline**: none;
268. }
270. .select {
271. **position**: relative;
272. **height**: 60px;
273. **padding**: 0 48px 0 24px;
274. box-sizing: border-box;
275. **font-family**: 'Montserrat';
276. **border**: 0;
277. border-radius: 16px;
278. **background-color**: rgb(238, 240, 245);
279. **background-image**: url('https://s3-us-west-2.amazonaws.com/s.cdpn.io/1329536/icon\_select\_s.svg');
280. **background-position**: calc(100% - 16px) center;
281. **background-repeat**: no-repeat;
282. box-shadow: inset 1px 1px 1px rgba(255, 255, 255, 0.6), inset -1px -1px 1px rgba(12, 13, 18, 0.06), -1px -1px 4px 0px rgba(255, 255, 255, 0.6), -2px -2px 8px 0px rgba(255, 255, 255, 0.6), 1px 1px 4px 0px rgba(12, 13, 18, 0.06), 2px 2px 8px 0px rgba(12, 13, 18, 0.06);
283. **font-size**: 18px;
284. **line-height**: 60px;
285. -webkit-appearance: none;
286. -moz-appearance: none;
287. appearance: none;
288. }
290. .select:focus {
291. **outline**: none;
292. }
293. **</style>**