

DEEP LEARNING SECTION : 1
TEAM NAME : INNOVATORS

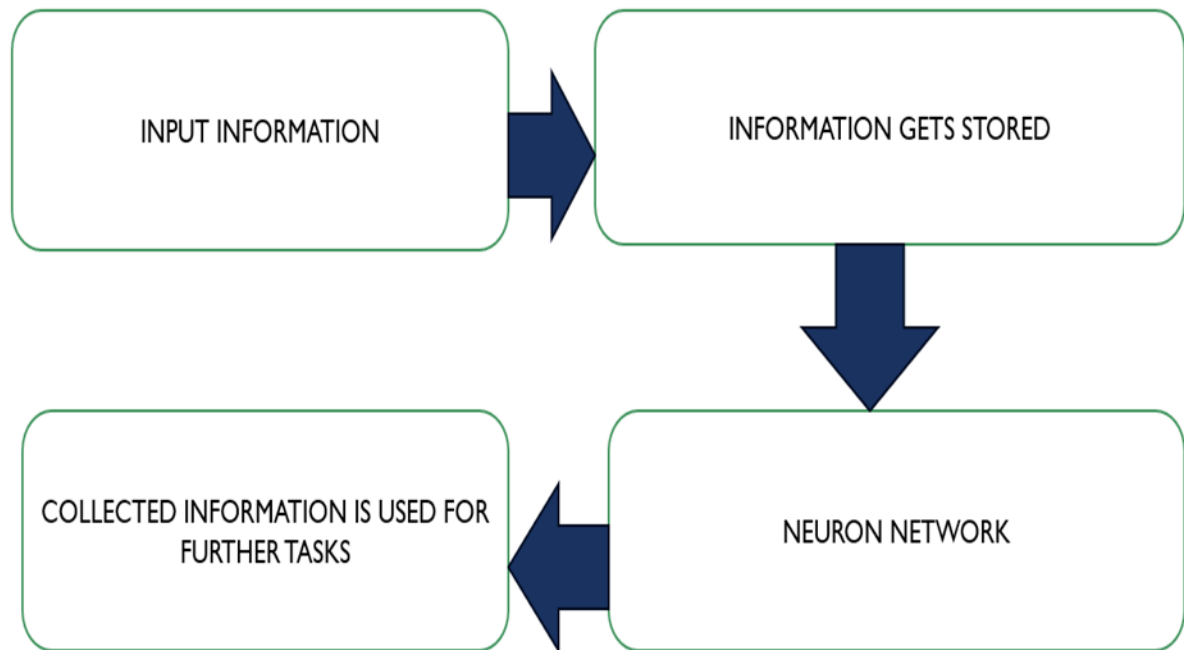
STATEMENT

- ☐ The statement focuses on to create a fine tuned model in which the information is given to the system ant it accumulates the raw data and it utilises it for further tasks that to be given

OBJECTIVE

- ☐ The main objective of the mentioned statement to give information to the database and it utilises that it for final specified tasks
- ☐ It's like teaching the child and the child utilises the knowledge in his future endeavours
- ☐ The datas were collected and stored in a network called neuron network
- ☐ The detailed information of the above mentioned were explained briefly in the upcoming.

FLOW DIAGRAM



WEBSCRAPPING

File Edit View Run Settings Add-ons Help

+ ✂ 📄 📋 ▶ ▶▶ Run All Code ▾

● Draft Session (11h:59m) HDD CPU RAM 🔌 ↺ ⋮

```
[1]:
import requests
from bs4 import BeautifulSoup
import re
import csv
```

▶

+ Code + Markdown

Method 1

Resisting the Immediate Temptation

Download Article


2

Scratch or press around the area that is irritated but not on it. The gate control theory of pain suggests that applying pressure and stimulation to another area can distract you from the itch and actually alleviate some of the pain.^[1]

• Snap a rubber band on your wrist when you feel the urge to scratch. Some people press an X into their skin near an itchy spot such as a mosquito bite. These are both examples of the gate control theory of pain at work to stop you from scratching.

ADVERTISEMENT

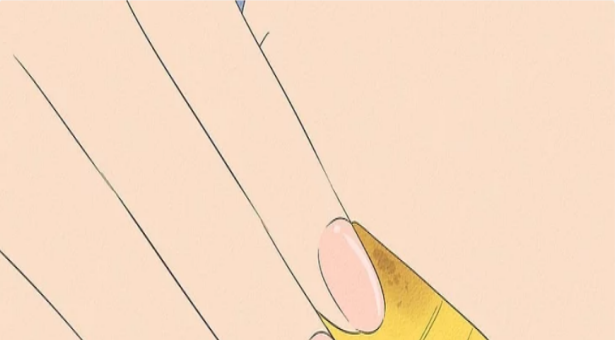
Ad



Best Pillow (2024)

Derila

Open >



to do anything...

WikiHow

Scratch or press around the area that is irritated but not on it.

"The gate control theory of pain suggests that applying pressure and stimulation to another area can distract you from the itch and actually alleviate some of the pain."

Sup id="ref-1" class="reference" aria-label="Link to Reference 1"

Sup

Ul

Li

Ul

Div class="clearall"

Div class="al_method">Advertisement

Script

WH.ads.addBodyAd('step2_ad_1')

Div class="scrolltmarker" data-isstep="1"

Div class="wh_ad_inner wh_ad_active wh_ad_spacing w_label al_grey" style="min-height: 90px;"

Li id="step-id-02"

Li id="step-id-03"

Li id="step-id-04"

Li id="step-id-05"

Div class="al_method">Advertisement

Script

WH.ads.addBodyAd('mobilemethod_ad_1')

Ol

Div class="clearall"

Div class="al_method" data-isstep="1"

Styles

Computed

Layout

Filter

:hov .cls

element.style {

.content

.section_text

.steps_list_2 ul li, .content .ingredients ul li, .content .thingsyoullneed ul li {

margin-bottom: 0;

.content #intro li,

.content

.section_text ul li {

margin-bottom: 5px;

@media only screen and (min-width: 728px) {

.stable .content p, .stable .content li,

.stable .content dl {

line-height: 25px;

.stable .content p,

.stable .content li,

.stable .content dl {

line-height: 25px;

.steps_list_2 li li

{

border: none;

margin: 0;

list-style: disc;

padding: 0;

.steps_list_2 li {

margin-bottom: 1em;

font-size: 16px;

line-height: normal;

list-style: none;

div#steps_1.section_text

ol.steps_list_2 li#step-id-01

div.steps_2 ul li

Console

Issues

```
1 if(len(subheadings)):
2     with open('/kaggle/working/wikiHow.csv', mode='a', newline='', encoding='utf-8') as csv_file:
3         writer = csv.writer(csv_file)
4         for i in range (len(subheadings)):
5             writer.writerow([article_title, subheadings [i], paragraphs[i]])
```

```
Cell In[13], line 1
      1 if(len(subheadings)):
```

SyntaxError: invalid syntax

+ Code

+ Markdown

```

html_content = response.content

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, 'html.parser')
article_title = soup.find('title').text.strip()
print(article_title + " " + str(count))

# Extract the subheadings and paragraphs using the appropriate HTML tags
subheadings = []
paragraphs = []
steps = soup.find_all('div', {'class': 'step'})

for step in steps:
    subheading_element = step.find('b')
    if(subheading_element is not None):
        subheading_text = subheading_element.text.strip().replace('\n', '')
        subheading_text = subheading_text.encode('ascii', errors='ignore').decode()
        subheading_text = re.sub(" ", subheading_text)
        subheadings.append(subheading_text)
        subheading_element.extract()
    for span_tag in step.find_all('span'):
        span_tag.extract()
    paragraph_text = step.text.strip().replace('\n', '').replace(' ', '')
    paragraph_text = paragraph_text.encode('ascii', errors='ignore').decode()
    paragraph_text = re.sub(" ", paragraph_text)
    paragraphs.append(paragraph_text)
if(len(subheadings)):
    with open('/kaggle/working/wikiHow.csv', mode="a", newline="", encoding='utf-8') as csv_file:
        writer = csv.writer(csv_file)
        for i in range(len(subheadings)):
            writer.writerow([article_title, subheadings[i], paragraphs[i]])

```

