



Streamlit 웹 페이지 정보 가져오기

하나의 페이지 정보를 가져오기

st.write 페이지의 코드 부분 가져오기

코드

st.markdown 페이지의 코드 부분 가져오기

결과

하나의 페이지에서 링크 정보를 가져와보기

Text elements의 각 링크 정보를 가져오기

결과

몇 가지 요소의 링크 정보를 가져와 보자.

5개의 element의 요소 정보를 가져와 보기

5개의 요소의 링크 정보와 기타 정보를 가져온다.

결과는 다음과 같다.

자 이제 각 요소의 링크 정보를 활용하여 해당 부분의 CODE블록을 가져와 보자.

LangChain 이용하여 링크에서 text 요약을 수행해 본다.

하나의 페이지 정보를 가져오기

st.write 페이지의 코드 부분 가져오기

URL : <https://docs.streamlit.io/library/api-reference/write-magic/st.write>

코드

Example

Its basic use case is to draw Markdown-formatted text, whenever the input is a string:

```
import streamlit as st

st.write('Hello, *World!* :sunglasses:')
```

```
import requests
from bs4 import BeautifulSoup

# Streamlit 문서 페이지의 URL
url = 'https://docs.streamlit.io/library/api-reference/write-magic/st.write'

# 페이지에 GET 요청을 보냄
response = requests.get(url)

# 페이지의 HTML 내용을 파싱
soup = BeautifulSoup(response.content, 'html.parser')

# <pre class="doctest-block"> 태그를 가진 모든 예제 코드 섹션을 찾음
example_codes = soup.find_all('pre', {'class': 'doctest-block'})

# 모든 예제 코드를 추출하고 출력
if example_codes:
    for example_code in example_codes:
        example_code_text = example_code.get_text()
        print(example_code_text)
        print("\n---\n") # 다른 코드 블록 사이에 구분자
else:
    print("예제 코드 섹션이 발견되지 않음.")
```

st.markdown 페이지의 코드 부분 가져오기

URL : <https://docs.streamlit.io/library/api-reference/text/st.markdown>

```
import requests
from bs4 import BeautifulSoup

# Streamlit 문서 페이지의 URL
url = 'https://docs.streamlit.io/library/api-reference/text/st.markdown'

# 페이지에 GET 요청을 보냄
response = requests.get(url)

# 페이지의 HTML 내용을 파싱
soup = BeautifulSoup(response.content, 'html.parser')

# <section class="code_Container__pq3Nr"> 태그의 모든 인스턴스를 추출
code_containers = soup.find_all('section', {'class': 'code_Container__pq3Nr'})

# <pre class="doctest-block"> 태그의 모든 인스턴스를 추출
doctest_blocks = soup.find_all('pre', {'class': 'doctest-block'})

# code_Container__pq3Nr 섹션의 내용을 출력
if code_containers:
    for container in code_containers:
        container_html = str(container)
        print(container_html)
        print("\n---\n") # 다른 섹션 사이에 구분자
else:
    print("code_Container__pq3Nr 섹션이 발견되지 않음.")

# doctest-block 섹션의 내용을 출력
if doctest_blocks:
    for block in doctest_blocks:
        block_text = block.get_text()
        print(block_text)
        print("\n===\n") # 다른 블록 사이에 구분자
```

```
else:
    print("doctest-block 블록이 발견되지 않음.")
```

결과

```
<section class="code_Container__pq3Nr"><div class="code_Pre__
md = st.text_area('Type in your markdown string (without oute
    "Happy Streamlit-ing! :balloon:")

st.code(f"""
import streamlit as st

st.markdown('{md}')
""")

st.markdown(md)
</code></div></section>

---

import streamlit as st

st.markdown("**Streamlit* is **really** ***cool***.")
st.markdown(''':red[Streamlit] :orange[can] :green[write] :blue[text] :v
    :gray[pretty] :rainbow[colors].''')
st.markdown("Here's a bouquet &mdash;\n
    :tulip::cherry_blossom::rose::hibiscus::sunflower

multi = '''If you end a line with two spaces,
a soft return is used for the next line.

Two (or more) newline characters in a row will result in a ha
'''
```

```
st.markdown(multi)
```

```
===
```

위와 같이 하나의 페이지의 정보를 가져오는 것이 가능하다.

그렇다면 링크가 많이 모여있는 상위 페이지에서 링크를 가져오는 프로그램을 짜본다.

하나의 페이지에서 링크 정보를 가져와보기

Text elements의 각 링크 정보를 가져오기

```
import requests
from bs4 import BeautifulSoup

# Streamlit Text API 문서 페이지의 URL
url = 'https://docs.streamlit.io/library/api-reference/text'

# 페이지에 GET 요청을 보냄
response = requests.get(url)

# 페이지의 HTML 내용을 파싱
soup = BeautifulSoup(response.content, 'html.parser')

# <a class="refCard_Container__cRE_M refCard_Third__QZ4rX">
# 태그의 모든 인스턴스를 추출
api_cards = soup.find_all('a', {'class': 'refCard_Container__cRE_M refCard_Third__QZ4rX'})

# 추출된 내용을 출력
for card in api_cards:
    href = card['href']
    title = card.find('h4').text if card.find('h4') else 'N'
```

```

o Title'
    description = card.find('p').text if card.find('p') else 'No Description'
    example_code = card.find('code').get_text() if card.find('code') else 'No Example Code'

    print(f"Link: {href}")
    print(f"Title: {title}")
    print(f"Description: {description}")
    print(f"Example Code: {example_code}")
    print("\n---\n")

```

결과

```

Link: /library/api-reference/text/st.markdown
Title: Markdown
Description: Display string formatted as Markdown.
Example Code: st.markdown("Hello **world**!")

```

```

Link: /library/api-reference/text/st.title
Title: Title
Description: Display text in title formatting.
Example Code: st.title("The app title")

```

```

Link: /library/api-reference/text/st.header
Title: Header
Description: Display text in header formatting.
Example Code: st.header("This is a header")

```

Link: </library/api-reference/text/st.subheader>
Title: Subheader
Description: Display text in subheader formatting.
Example Code: `st.subheader("This is a subheader")`

Link: </library/api-reference/text/st.caption>
Title: Caption
Description: Display text in small font.
Example Code: `st.caption("This is written small caption text")`

Link: </library/api-reference/text/st.code>
Title: Code block
Description: Display a code block with optional syntax highlighting.
Example Code: `st.code("a = 1234")`

Link: </library/api-reference/text/st.text>
Title: Preformatted text
Description: Write fixed-width and preformatted text.
Example Code: `st.text("Hello world")`

Link: </library/api-reference/text/st.latex>
Title: LaTeX
Description: Display mathematical expressions formatted as LaTeX.
Example Code: `st.latex("\int a x^2 \, dx")`

Link: `/library/api-reference/text/st.divider`

Title: Divider

Description: Display a horizontal rule.

Example Code: `st.divider()`

몇 가지 요소의 링크 정보를 가져와 보자.

```
### Write and Magic : https://docs.streamlit.io/library/api-reference/text/st.write
### Text Element : https://docs.streamlit.io/library/api-reference/text/st.text
### Data Element : https://docs.streamlit.io/library/api-reference/data/st.data
### Chart Element : https://docs.streamlit.io/library/api-reference/chart/st.area
### input widgets : https://docs.streamlit.io/library/api-reference/widgets/st.text_input
```

5개의 element의 요소 정보를 가져와 보기

5개의 요소의 링크 정보와 기타 정보를 가져온다.

```
### 5개의 element 링크 가져오기
gubun = ["Write and Magic", "Text elements",
        "Data elements", "Chart elements"]

gubun_links = { "Write and Magic" : "https://docs.streamlit.io/library/api-reference/text/st.write",
                "Text elements" : "https://docs.streamlit.io/library/api-reference/text",
                "Data Element" : "https://docs.streamlit.io/library/api-reference/data",
                "Chart elements" : "https://docs.streamlit.io/library/api-reference/chart",
                "Input widgets" : "https://docs.streamlit.io/library/api-reference/widgets"
                }
```



```

for key in gubun_links:
    print(key, gubun_links[key])

### data Element 정보 가져오기
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Initialize an empty list to store data
data = []

for key in gubun_links:
    print(key, gubun_links[key])

    # Streamlit Text API 문서 페이지의 URL
    url = gubun_links[key]

    # 페이지에 GET 요청을 보냄
    response = requests.get(url)

    # 페이지의 HTML 내용을 파싱
    soup = BeautifulSoup(response.content, 'html.parser')

    # <a class="refCard_Container__cRE_M refCard_Third__QZ4rX
    api_cards = soup.find_all('a', {'class': 'refCard_Container__cRE_M refCard_Third__QZ4rX

    # 기본 URL
    base_url = 'https://docs.streamlit.io'

    # 추출된 내용을 출력
    for card in api_cards:
        # 완전한 URL 생성
        href = base_url + card['href']
        title = card.find('h4').text if card.find('h4') else
        description = card.find('p').text if card.find('p') else
        example_code = card.find('code').get_text() if card.f

```

```

print(f"Link: {href}")
print(f"Title: {title}")
print(f"Description: {description}")
print(f"Example Code: {example_code}")
print("\n---\n")

data.append({'Gubun': key, 'Link': href, 'Title': tit

# Create a DataFrame from the 'data' list
df = pd.DataFrame(data)
df

```

결과는 다음과 같다.

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#)...

	Gubun	Link	Title	Description	Example Code
0	Write and Magic	https://docs.streamlit.io/library/api-referenc...	st.write	Write arguments to the app.	st.write("Hello **world**!")\nst.write(my_data...
1	Write and Magic	https://docs.streamlit.io/library/api-referenc...	Magic	Any time Streamlit sees either a variable or l...	st.write
2	Text elements	https://docs.streamlit.io/library/api-referenc...	Markdown	Display string formatted as Markdown.	st.markdown("Hello **world**!")\n
3	Text elements	https://docs.streamlit.io/library/api-referenc...	Title	Display text in title formatting.	st.title("The app title")\n
4	Text elements	https://docs.streamlit.io/library/api-referenc...	Header	Display text in header formatting.	st.header("This is a header")\n
5	Text elements	https://docs.streamlit.io/library/api-referenc...	Subheader	Display text in subheader formatting.	st.subheader("This is a subheader")\n
6	Text elements	https://docs.streamlit.io/library/api-referenc...	Caption	Display text in small font.	st.caption("This is written small caption text...
7	Text elements	https://docs.streamlit.io/library/api-referenc...	Code block	Display a code block with optional syntax high...	st.code("a = 1234")\n

자 이제 각 요소의 링크 정보를 활용하여 해당 부분의 CODE블록을 가져와 보자.

각 링크 정보, Title, 링크별 간단한 설명, 한줄 예제 코드 등의 정보 수집

```

### 4개의 element 링크 가져오기
gubun = ["Write and Magic", "Text elements",
         "Data elements", "Chart elements"]

```

```

gubun_links = { "Write and Magic" : "https://docs.streamlit.io/
                "Text elements" : "https://docs.streamlit.io/
                "Data Element" : "https://docs.streamlit.io/1
                "Chart elements" : "https://docs.streamlit.io
                }

### data Element 정보 가져오기
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Initialize an empty list to store data
data = []

for key in gubun_links:
    print(key, gubun_links[key])

    # Streamlit Text API 문서 페이지의 URL
    url = gubun_links[key]

    # 페이지에 GET 요청을 보냄
    response = requests.get(url)

    # 페이지의 HTML 내용을 파싱
    soup = BeautifulSoup(response.content, 'html.parser')

    # <a class="refCard_Container__cRE_M refCard_Third__QZ4rX
    api_cards = soup.find_all('a', {'class': 'refCard_Contain

    # 기본 URL
    base_url = 'https://docs.streamlit.io'

    # 추출된 내용을 출력
    for card in api_cards:
        # 완전한 URL 생성
        href = base_url + card['href']
        title = card.find('h4').text if card.find('h4') else
        description = card.find('p').text if card.find('p') e

```

```

example_code = card.find('code').get_text() if card.f

print(f"Link: {href}")
print(f"Title: {title}")
print(f"Description: {description}")
print(f"Example Code: {example_code}")
print("\n---\n")

# 한 페이지의 상세 코드 블록을 가져오기
# Streamlit 문서 페이지의 URL
url_one = href

# 페이지에 GET 요청을 보냄
response = requests.get(url_one)

# 페이지의 HTML 내용을 파싱
soup = BeautifulSoup(response.content, 'html.parser')

# <section class="code_Container__pq3Nr"> 태그의 모든 인
code_containers = soup.find_all('section', {'class':

# <pre class="doctest-block"> 태그의 모든 인스턴스를 추출
doctest_blocks = soup.find_all('pre', {'class': 'doct

code_all = ""
# code_Container__pq3Nr 섹션의 내용을 출력
if code_containers:
    for container in code_containers:
        container_html = str(container)
        code_all = code_all + container_html + "\n---"
        #print(container_html)
        #print("\n---\n") # 다른 섹션 사이에 구분자
else:
    # print("code_Container__pq3Nr 섹션이 발견되지 않음.")
    code_all = code_all + "\n---\n"

# doctest-block 섹션의 내용을 출력
if doctest_blocks:

```

```

        for block in doctest_blocks:
            block_text = block.get_text()
            code_all = code_all + block_text + "\n---\n"
            #print(block_text)
            #print("\n===\n") # 다른 블록 사이에 구분자
    else:
        # print("doctest-block 블록이 발견되지 않음.")
        code_all = code_all + "\n---\n"

    data.append({'Gubun': key, 'Link': href, 'Title': tit
                'Description': description, 'Example Cod
                "Detail_Code" : code_all})

# Create a DataFrame from the 'data' list
df = pd.DataFrame(data)
df.to_csv("result.csv", index=False)

```

결과 이미지

A	B	C	D	E	F
Gubun	Link	Title	Description	Example Code	Detail_Code
Write and	https://docs.streamlit.io/library/api-reference/write-magic/st.write	st.write	Write arguments to the app.	<pre> st.write("Hello **world**!") st.write(my_data_frame) st.write(my_mpl_figure) </pre>	<pre> --- import streamlit as st st.write("Hello, *World!* :sunglasses:") --- import streamlit as st import pandas as pd </pre>
Write and	https://docs.streamlit.io/library/api-reference/write-magic/magic	Magic	Any time Streamlit sees either a va	st.write	<pre> <section class="code_container_pq3ivR"><div class="code_Pre_3lyyf"><code class="language- python"># Draw a title and some text to the app: ... # This is the document title This is some _markdown_ ... import pandas as pd df = pd.DataFrame({'col1': [1,2,3]}) </pre>
Text elem	https://docs.streamlit.io/library/api-reference/text/st.markdown	Markdown	Display string formatted as Markdc	<pre> st.markdown("Hello **world**!") </pre>	<pre> <section class="code_container_pq3Nr"><div class="code_Pre_3lyyf"><code class="language- python">import streamlit as st md = st.text_area("Type in your markdown string (without outer quotes)", "Happy Streamlit-ing! :balloon:") st.code(f""" import streamlit as st st.markdown("""(md)""") </pre>

LangChain 이용하여 링크에서 text 요약을 수행해 본다.