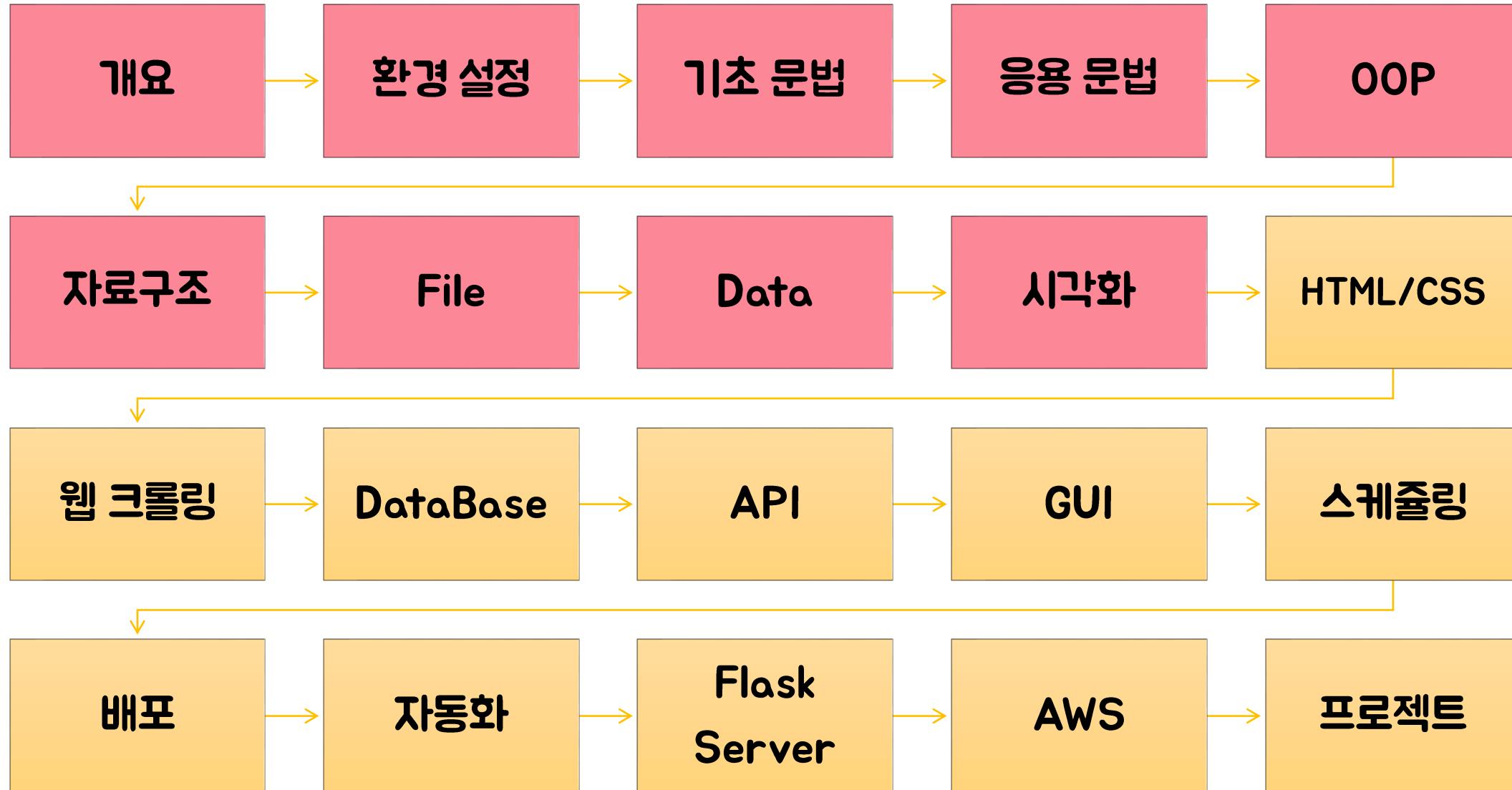




대한상공회의소
서울기술교육센터

나예호 교수



웹 크롤링(Web Crawling) 및 스크래핑(Scraping) 개념

- 웹사이트에서 데이터를 자동으로 가져오는 기술.
- 검색 엔진도 크롤링 및 스크래핑을 활용하여 정보를 수집함.

크롤링 vs 스크래핑

- 크롤링: 여러 웹페이지를 탐색하면서 정보를 가져옴.
- 스크래핑: 특정 웹페이지에서 필요한 데이터만 추출함.

robots.txt

- 액세스 하거나 정보수집을 해도 되는 페이지 등을 알려주는 .txt (텍스트) 파일
 - 검색엔진은 해당 txt의 내용을 기반으로 허용되지 않는 페이지로부터 정보수집을 원칙적으로 하지 않음
-
- * 우리도 확인 후 정보 수집을 하자
 - * 해당 사이트에서 제공되는 API가 있다면 활용하자

<https://naver.com/robots.txt>

User-agent: *

Disallow: /

Allow: /\$

Allow: /.well-known/privacy-sandbox-attestations.json

User-agent: *

- 모든 웹 크롤러(봇)를 대상으로 하는 규칙
- *는 모든 검색 엔진 및 크롤링 봇(Googlebot, Bingbot, Naverbot 등)을 의미

Disallow: /

- 웹사이트의 모든 페이지에 대한 접근을 금지
- 즉, 기본적으로 사이트 전체 **크롤링이 차단**

Allow: /\$

- 단, 루트 페이지(홈페이지)(<https://www.naver.com/>)는 접근을 허용
- 여기서 \$는 정규표현식의 끝을 의미하여, 정확히 루트 페이지만 허용

Allow: /.well-known/privacy-sandbox-attestations.json

- 이 특정 JSON 파일에 대한 접근은 허용됩니다.
- 이는 보안 관련 설정 정보를 포함한 파일로, 인증 및 프라이버시 기능에 사용

requests 라이브러리 활용

- 웹페이지 데이터를 가져오기 위한 HTTP 요청 라이브러리
- GET, POST 요청을 사용하여 데이터 수집

구분	GET 방식	POST 방식
데이터 전송 방식	URL에 데이터를 포함하여 전송 (쿼리스트링)	HTTP 메시지 본문(Body)에 데이터를 포함
URL 표시 여부	데이터가 URL에 표시됨	데이터가 URL에 표시되지 않음
보안성	낮음 (데이터가 URL에 노출됨)	높음 (데이터가 본문에 숨겨짐)
전송 데이터 크기 제한	브라우저 및 서버에 따라 제한 존재 (약 2048자)	상대적으로 큰 데이터 전송 가능
용도	데이터 조회, 검색 등 (읽기 전용 요청)	로그인, 회원가입, 파일 업로드 등 (데이터 변경 요청)
멀티파트 전송	지원하지 않음	지원함 (파일 업로드 등 멀티파트 데이터 전송 가능)
속도	빠름 (간단한 데이터 전송)	상대적으로 느림 (본문 처리 필요)
Idempotent (멱등성)	예 (같은 요청을 반복해도 결과가 동일함)	아니오 (같은 요청이라도 결과가 달라질 수 있음)

requests 라이브러리 활용 GET 요청

```
import requests

# 네이버 홈페이지 HTML 가져오기
url_naver = "https://www.naver.com"
response = requests.get(url_naver)
print(response)
print(response.status_code)
```

```
<Response [200]>
200
```

HTTP 상태 코드

- 200 OK: 요청 성공.
- 404 Not Found: 페이지 없음.
- 403 Forbidden: 접근 금지.
- 500 Internal Server Error: 서버 오류.

요청을 통한 응답 데이터 처리

- `response.text` : 응답 본문을 문자열(str)로 반환
- `response.content` : 응답 본문을 바이트(byte)로 반환
- `response.json()` : 응답 본문을 딕셔너리(dict)로 반환

User-Agent

- HTTP 요청 헤더의 일부로, 사용자가 사용하는 브라우저나
기기의 정보를 서버에 전달하는 문자열
- 웹사이트 서버는 이 정보를 통해 요청을 보낸 주체가
웹 브라우저인지, 모바일인지, 혹은 봇인지를 식별

```
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 11.0;  
Win64; x64)"}
```

User-Agent

```
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 11.0; Win64; x64)"}
response = requests.get(url=url_naver, headers=headers)
print("User-Agent 설정 후 상태 코드:", response.status_code)
```

User-Agent 설정 후 상태 코드: 200

예외 처리 `raise_for_status()`

- 응답 결과가 4XX(클라이언트 오류) 혹은 5XX(서버 오류)일 시
`except`구문 호출
- `if`문을 안 써도 된다는 장점!!

```
try:  
    response = requests.get(url=url_naver, headers=headers)  
    response.raise_for_status()  
except requests.exceptions.HTTPError as http_err:  
    print(f"HTTP 오류 발생: {http_err}")
```

예외 처리 `raise_for_status()`

```
try:  
    response = requests.get(url=url_naver, headers=headers)  
    response.raise_for_status()  
except requests.exceptions.HTTPError as http_err:  
    print(f"HTTP 오류 발생: {http_err}")  
except requests.exceptions.ConnectionError:  
    print("연결 오류 발생!")  
except requests.exceptions.Timeout:  
    print("요청 시간 초과!")  
except requests.exceptions.RequestException as err:  
    print(f"기타 오류 발생: {err}")
```

BeautifulSoup

- HTML/XML을 파싱하여 데이터를 쉽게 추출하는 라이브러리
- find()와 find_all() 메서드를 활용하여 특정 태그 검색 가능

```
from bs4 import BeautifulSoup as soup
```

BeautifulSoup

html_exam.html

```
<html>
  <head>
    <title>예제 페이지</title>
  </head>
  <body>
    <h1>안녕하세요</h1>
    <p class="text">웹 크롤링을 배우는 중입니다.</p>
    <a href="https://www.naver.com">네이버</a>
    <a href="https://www.google.com">구글</a>
  </body>
</html>
```

BeautifulSoup

```
with open("exam.html", "r", encoding="utf-8") as file:  
    html_exam = file.read()  
  
print(html_exam)
```

BeautifulSoup

find() : 특정 태그에서 첫 번째 요소 가져오기.

```
from bs4 import BeautifulSoup as bs

soup = bs(html_exam, "html.parser")
title = soup.find("title").text
print("페이지 제목 :", title)
```

페이지 제목 : 예제 페이지

BeautifulSoup

Q. <h1>태그 안의 내용을 h1 변수에 저장하고 출력하시오

```
h1 = soup.find("h1").text  
print("h1 내용 :", h1)
```

h1 내용 : 안녕하세요

BeautifulSoup

Q. 특정 페이지에서 <h1>태그 안의 내용을 출력해보시오

```
kccistc = "https://www.kccistc.net/cms/cmsDetail.do?rootMenuId=3902&menuId=3908&cms_id=58"

# 특정 페이지에서 <h1> 태그 가져오기
response = requests.get(kccistc, headers=headers)
soup2 = bs(response.text, "html.parser")
h1_tag = soup2.find("h1").text
print("제목 :", h1_tag.strip())
```

BeautifulSoup

`find_all(태그)`: 특정 태그에서 모든 요소 가져오기

Q. `<a>` 태그 모두 가져오기

```
a_all = soup.find_all("a")
for a in a_all:
    print("a태그 내용 :", a.text)
```

a태그 내용 : 네이버

a태그 내용 : 구글

BeautifulSoup

find_all(태그): 특정 태그에서 모든 요소 가져오기

Q. naver <a> 태그를 처음 5개만 출력해주세요

```
response = requests.get("https://www.naver.com")
soup = bs(response.text, "html.parser")
all_links = soup.find_all("a")
for link in all_links[:5]: # 처음 5개만 출력
    print("링크 :", link.get("href"))
```

BeautifulSoup

find_all(태그): 특정 태그에서 모든 요소 가져오기

Q. nate news에서 기사 제목 5개를 출력해주세요

```
# 네이트 뉴스 사이트에서 기사 제목 가져오기
response = requests.get("https://news.nate.com/recent?mid=n0100", headers=headers)
soup = bs(response.text, "html.parser")
headlines = soup.find_all("h2")
for headline in headlines[:5]: # 처음 5개만 출력
    print("기사 제목 :", headline.text)
```

BeautifulSoup

선택자를 활용해 데이터 가져오기

```
response = requests.get("https://news.nate.com/recent?mid=n0100", headers=headers)
soup = BeautifulSoup(response.text, "html.parser")
news_list = soup.select(".tit") # 클래스명
for news in news_list[:5]: # 처음 5개만 출력
    print("뉴스 제목:", news.get_text())
```

네이트 뉴스 기사를 10페이지 수집 후
AI 키워드가 들어간 기사의 제목과 링크를 필터링한
`filtered_news.csv`를 저장하자

이후, DataFrame 형태로 read 후
내용을 확인해보자

<h1>나만의 간단한 웹 페이지를 만들어 보자</h1>

HTTP : Hyper Text Transfer Protocol

인터넷에서 하이퍼텍스트 문서를
교환하기 위하여 사용되는 통신 규약

HTML : Hyper Text Markup Language

웹 페이지에 정보를 담아 표시하기 위한 마크업 언어

HTTP : Hyper Text Transfer Protocol

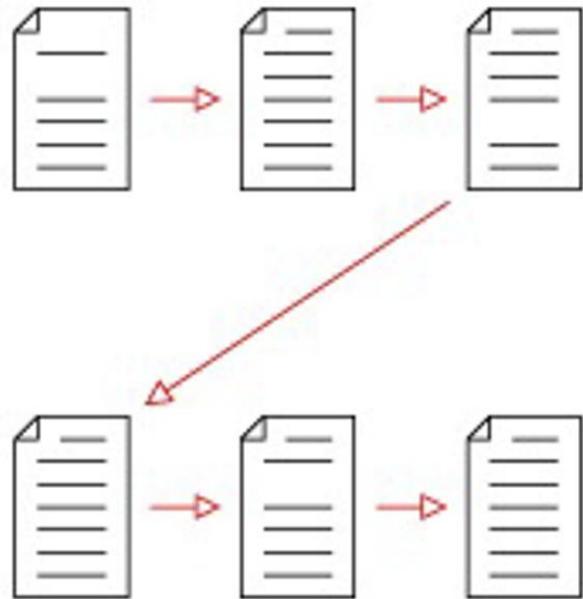
인터넷에서 하이퍼텍스트 문서를
교환하기 위하여 사용되는 통신 규약

Hyper Text

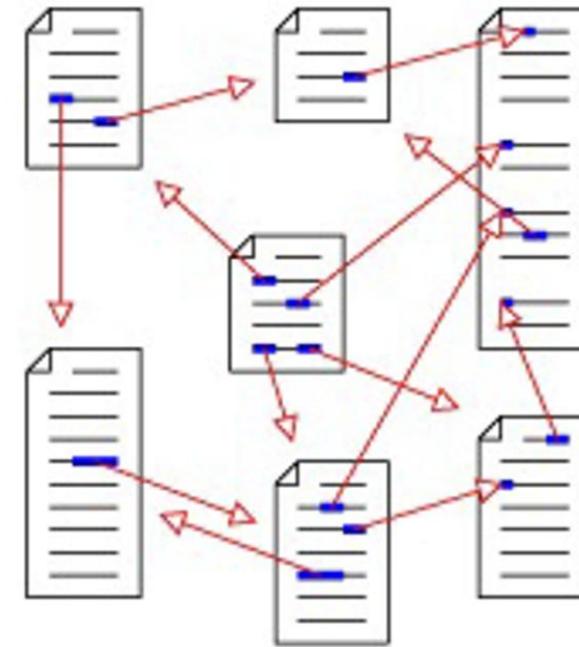
현재 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트

HTML : Hyper Text MarkUp Language

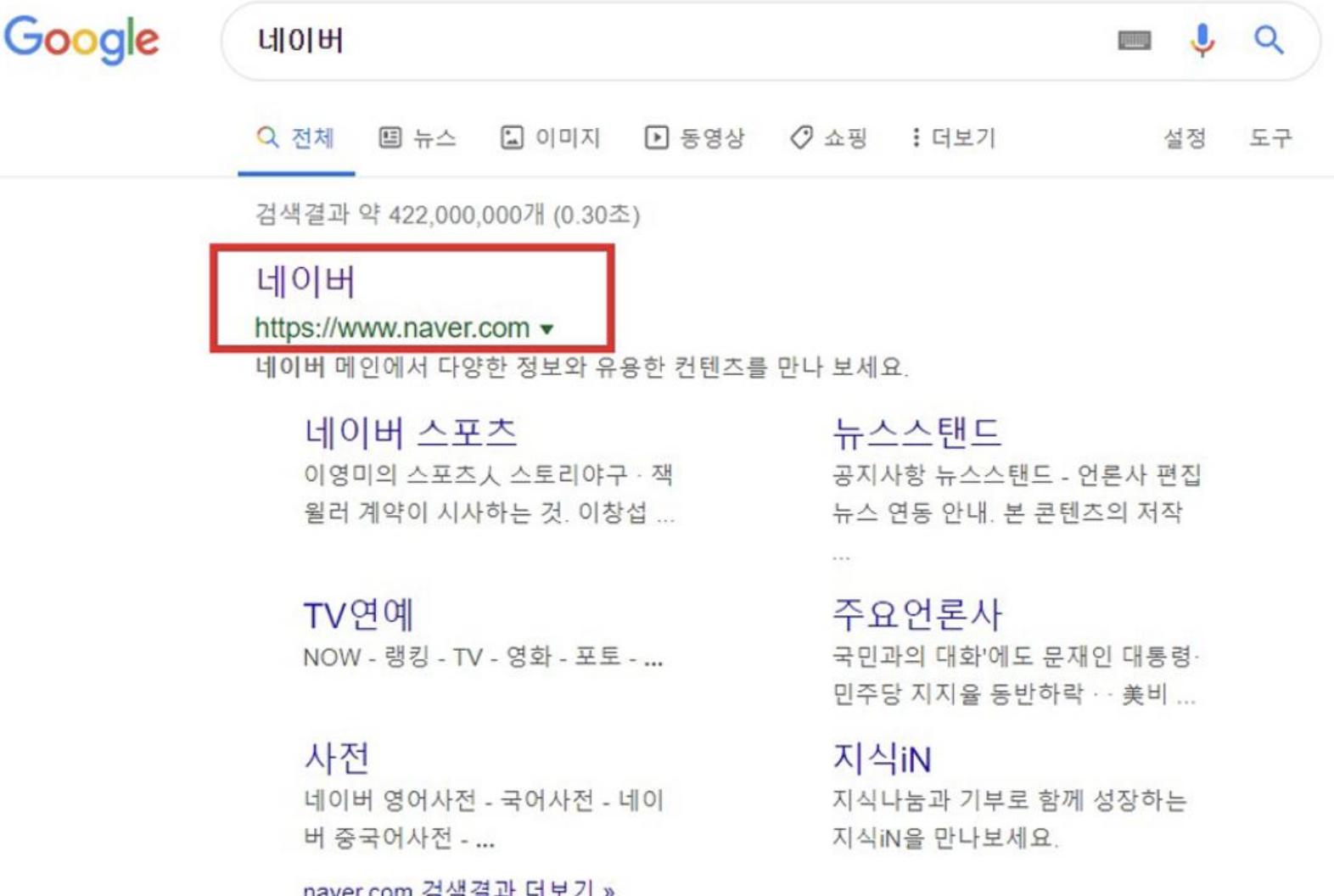
웹 페이지에 정보를 담아 표시하기 위한 마크업 언어



Text



HyperText



Google 네이버

전체 뉴스 이미지 동영상 쇼핑 더보기 설정 도구

검색결과 약 422,000,000개 (0.30초)

네이버
[https://www.naver.com ▾](https://www.naver.com)

네이버 메인에서 다양한 정보와 유용한 컨텐츠를 만나 보세요.

네이버 스포츠
이영미의 스포츠人 스토리야구 · 책
윌러 계약이 시사하는 것. 이창섭 ...

뉴스스탠드
공지사항 뉴스스탠드 - 언론사 편집
뉴스 연동 안내. 본 콘텐츠의 저작
...

TV연예
NOW - 랭킹 - TV - 영화 - 포토 - ...

주요언론사
국민과의 대화'에도 문재인 대통령·
민주당 지지율 등반하락 · 美비 ...

사전
네이버 영어사전 - 국어사전 - 네이
버 중국어사전 - ...

지식iN
지식나눔과 기부로 함께 성장하는
지식iN을 만나보세요.

naver.com 검색결과 더보기 »



팀 버너스 리 (HTML의 창시자)

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

What's out there?

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

Help

on the browser you are using

Software Products

A list of W3 project components and their current state. (e.g. [Line Mode](#) , [X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

Technical

Details of protocols, formats, program internals etc

Bibliography

Paper documentation on W3 and references.

People

A list of some people involved in the project.

History

A summary of the history of the project.

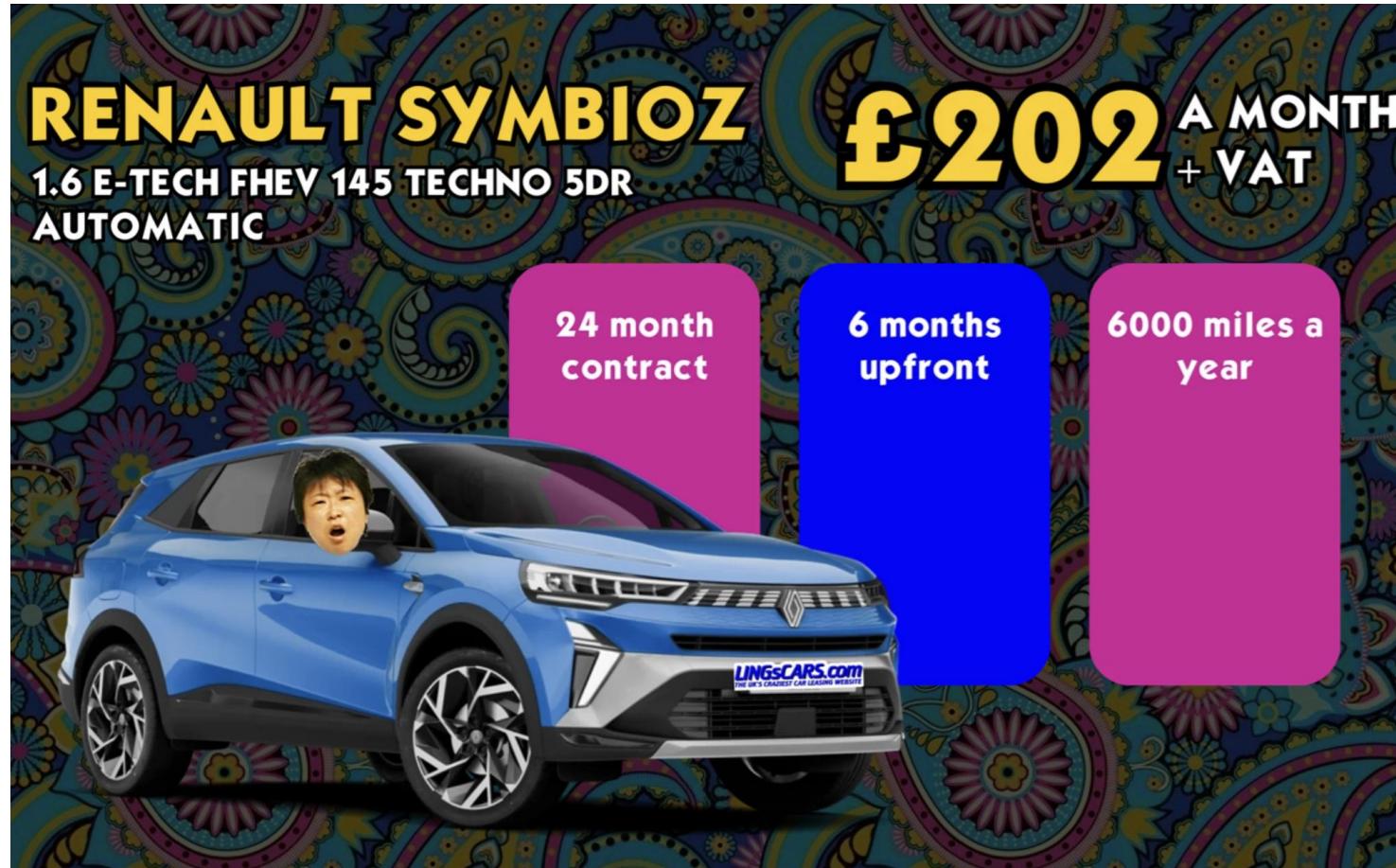
How can I help ?

If you would like to support the web..

Getting code

Getting the code by [anonymous FTP](#) , etc.

최초의 웹페이지



최악의 웹페이지

<https://www.lingscars.com/>

MarkUp

어딘가에 Mark! 즉, 표시를 해두는 것

실업자를 대상으로 고용촉진과 고용안정을 도모하기 위해 취업에 필요한 기술·기능을 교육 교육실시 장소 서울기술교육센터 입학자격(과정별 상이할 수 있음) 자격 : 대학(교) 이상 교육과정 졸업(예정)자로 미취업자(고용보험 미가입자) 및 관련 분야 경력자 입학특전 협약기업 취업 알선 교육비 전액(입학금, 수업료 등 전액) 정부지원 훈련장려금 지급(과정별 상이하며, 규정 변경시 미지급될 수 있음) 교재 무료 지급 실습위주 PROJECT 수업 진행 훈련기간 3개월~10개월

실업자를 대상으로 고용촉진과 고용안정을 도모하기 위해 취업에 필요한 기술·기능을 교육

- 교육실시 장소
서울기술교육센터

- 입학자격(과정별 상이할 수 있음)
자격 : 대학(교) 이상 교육과정 졸업(예정)자로 미취업자(고용보험 미가입자) 및 관련분야 경력자

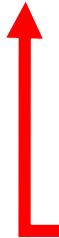
- 입학특전
협약기업 취업 알선
교육비 전액(입학금, 수업료 등 전액) 정부지원
훈련장려금 지급(과정별 상이하며, 규정 변경시 미지급될 수 있음)
교재 무료 지급
실습위주 PROJECT 수업 진행

- 훈련기간
3개월~10개월

HTML의 구성요소

시작태그

<h1>나만의 간단한 웹 페이지를 만들어 보자</h1>



Content(내용)



끝태그



Element(요소)

HTML의 구성요소

속성(attribute)

<P align="center">Hello World!</P>

값(value)

태그

strong, u, h1, h2 ...

약 150개

구성요소

```
<!DOCTYPE html>
<head>
    <meta>
        <title>Test홈페이지</title>
    </head>
    <body>
        환영합니다
    </body>
</html>
```

구성요소

```
<!DOCTYPE html>      ----- 문서 형식 정의
<head>            ----- HTML문서의 머릿글
  <meta>          ----- HTML문서의 정보정의
  <title>Test홈페이지</title> ----- HTML 문서의 제목
</head>
<body>
  환영합니다 ----- HTML문서 내용 (텍스트, 이미지, 내용 등)
</body>
</html>
```

body 태그 속성

속성	설명
background	배경 이미지 지정
bgcolor	배경색 지정
text	글꼴 색 지정
link	링크 색 지정
vlink	방문했던 링크 색 지정
alink	링크를 클릭하는 순간의 색 지정

제목 태그

<h> 제목 </h>

: html 문서 본문 내 제목을 표현하는 태그(h1 ~ h6)

글자 태그

<P> 제목 </P>

: 본문의 내용을 단락으로 표현할 때 사용하는 태그

** 본문 **

: 본문의 내용을 문장으로 표현할 때 사용하는 태그

문단 태그

**
**

: 줄 바꿈(개행) 태그

<hr/>

: 단락 구분을 위해 사용하는 태그

hr 태그 속성

속성	설명
align	수평선의 정렬 방식(right, left, center)
color	수평선의 색
size	수평선의 굵기
width	수평선의 가로 길이
noshade	그림자가 없는 평면의 수평선

hr 태그 실습



폭 : 150 크기 : 140

폭 : 150 크기 : 120

폭 : 150 크기 : 100

폭 : 150 크기 : 80

:

글자 태그

** 텍스트 **

: 다른 텍스트와 구별할 때 사용되는 태그

** 텍스트 **

: 중요한 문구를 강조하는 태그

리스트 태그

: 번호 없는 목록을 사용할 때 사용하는 태그

: 번호 있는 목록을 사용할 때 사용하는 태그

: 공통적으로 사용되는 태그