



인공지능과 함께하는 배리어프리 키오스크 만들기

중흥중학교

Contents

01

Barrier-Free가 무엇일까?

02

인공지능이 뭘까?

03

생성형 인공지능은 어떻게 발전해 왔을까?

01 Barrier-Free가 무엇일까?

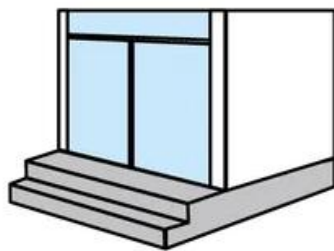
◆ Barrier-free



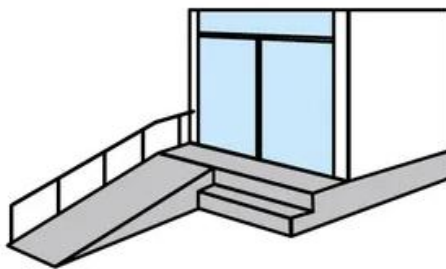
01 Barrier-Free가 무엇일까?

◆ 배리어프리 (Barrier-free)

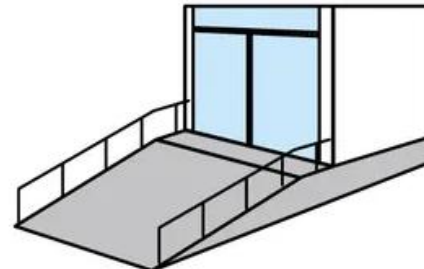
- 장애인, 노인, 임산부 등 사회적 약자들이 불편을 느끼지 않도록 물리적, 심리적 장벽을 제거하는 것
- 배리어프리 디자인 (Barrier-free design): 장애인을 배려한 디자인
- 유니버설 디자인 (Universal design): 모든 사람을 위한 디자인



기존의 디자인



배리어프리 디자인



유니버설 디자인

01 Barrier-Free가 무엇일까?

◈ 배리어프리 (Barrier-free)



배리어프리 디자인



유니버설 디자인

01 Barrier-Free가 무엇일까?

◇ 배리어프리 (Barrier-free)



저상버스



저상버스 내부

사회 곳곳에서 배리어프리와 유니버설 디자인이 확산하고 있어요!

01 Barrier-Free가 무엇일까?

◆ 키오스크 (Kiosk)

- 정식 명칭 : 인터랙티브 키오스크(Interactive Kiosk), 터치스크린이 탑재된 안내기거나 무인주문기 형태
- 고객이 셀프로 주문, 결제를 할 수 있도록 제작한 시스템
- 2020년대 들어 폭발적으로 성장

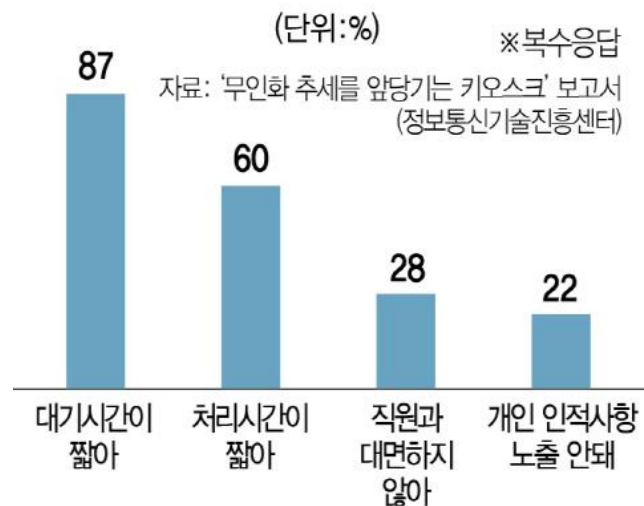


01 Barrier-Free가 무엇일까?

❖ 키오스크 (Kiosk)

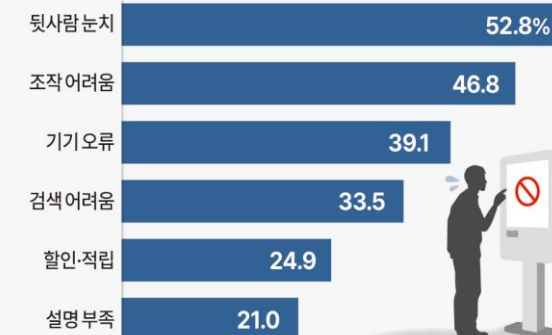
- 장점: 비대면 주문 가능, 복잡한 주문 가능, 소통 오류 개선, 주문속도 향상, 인건비 절감
- 단점: 비효율적 설계 시, 또는 하드웨어 성능 미달 시 주문속도 저하 및 고객 불만 유발
키오스크에 없는 주문 불가, **노년층/장애인의 접근성 문제**

키오스크가 편리한 이유



키오스크 이용 불편 사례

2022년 7월 엠브레인퍼블릭 키오스크 소비자 500명 설문조사(중복응답) 500명 중 233명 46.6% 이용 불편 경험



자료: 한국소비자원

김민지 기자, 안예지 인턴 20221124

01 Barrier-Free가 무엇일까?

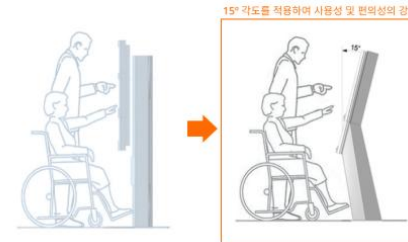
❖ 배리어프리 키오스크 (Barrier-free Kiosk)

- 장애인, 어린이, 노약자를 포함한 모두가 이용할 수 있도록 사용자 접근성을 보장한 키오스크
- 2025년 2월, 면적 50m² 이상 사업장 대상 배리어프리 키오스크 설치 의무화



휠체어 사용자를 위한

자동 높낮이 특허 보유
사용자 높이 감지 UI 하향 조절(세로형)



고령자 또는 시각장애인을 위한

크기조절 제공
고대비모드 제공



시각장애인을 위한

매뉴얼 점자 디스플레이 제공
모든 시각 정보를 음성으로 안내

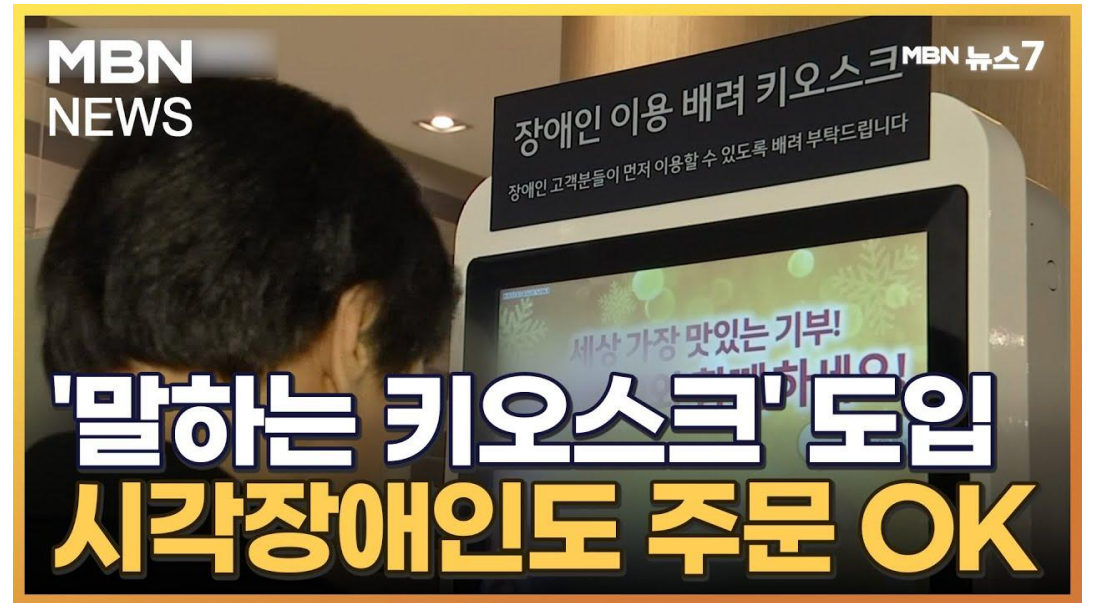
청각장애인을 위한

모든 청각 정보를 시각 정보로 제공



01 Barrier-Free가 무엇일까?

◆ 시각장애인을 위한 음성지원 키오스크 만들기



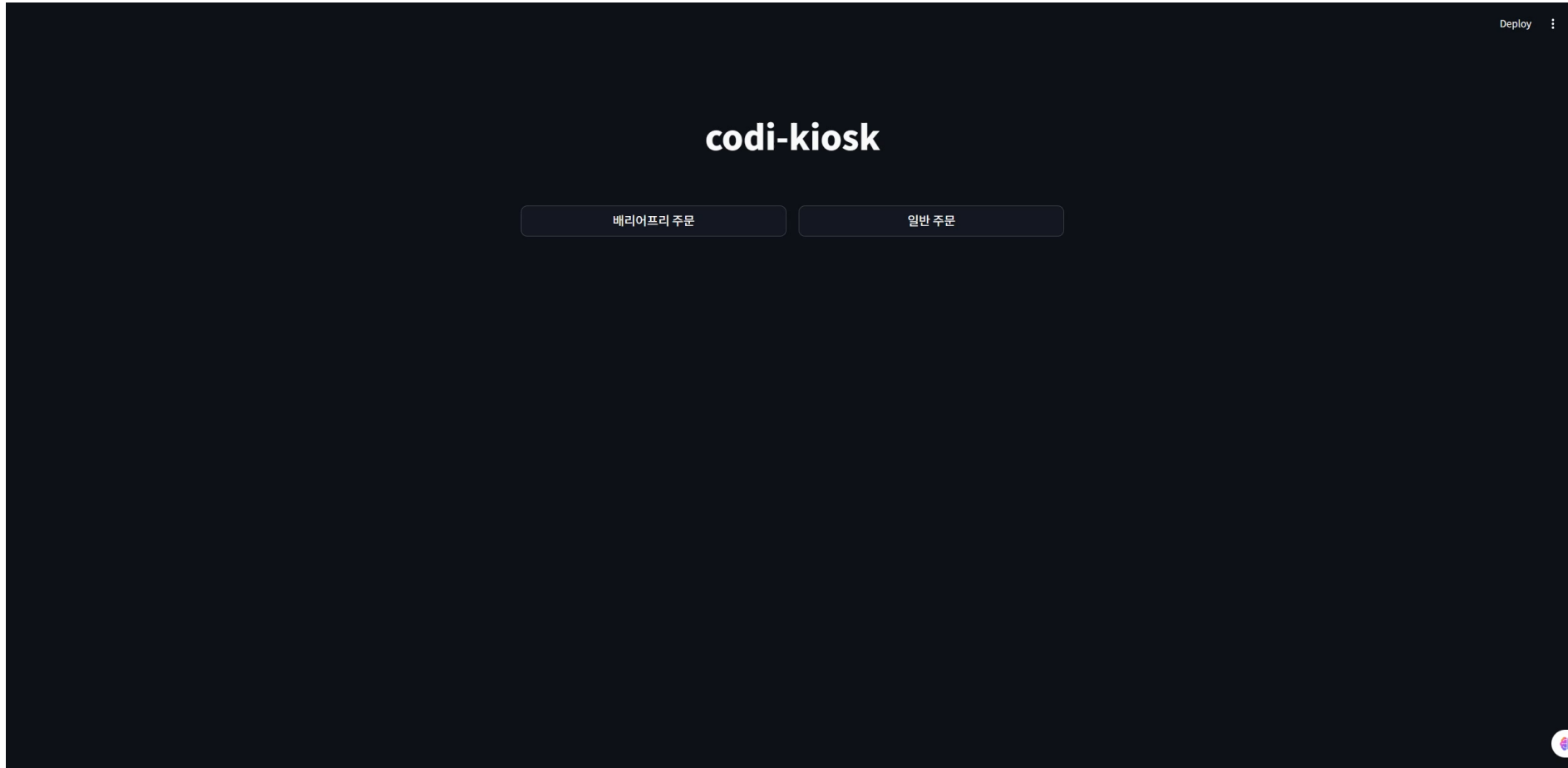
01 Barrier-Free가 무엇일까?

◈ 여러분이 만들게 될 배리어프리 키오스크 예시



01 Barrier-Free가 무엇일까?

❖ 여러분이 만들게 될 배리어프리 키오스크 예시



❖ 배리어프리 키오스크를 만들기 위한 도구들



Visual Studio Code

코드 인터프리터



ANACONDA®

개발환경 세팅



UX/UI

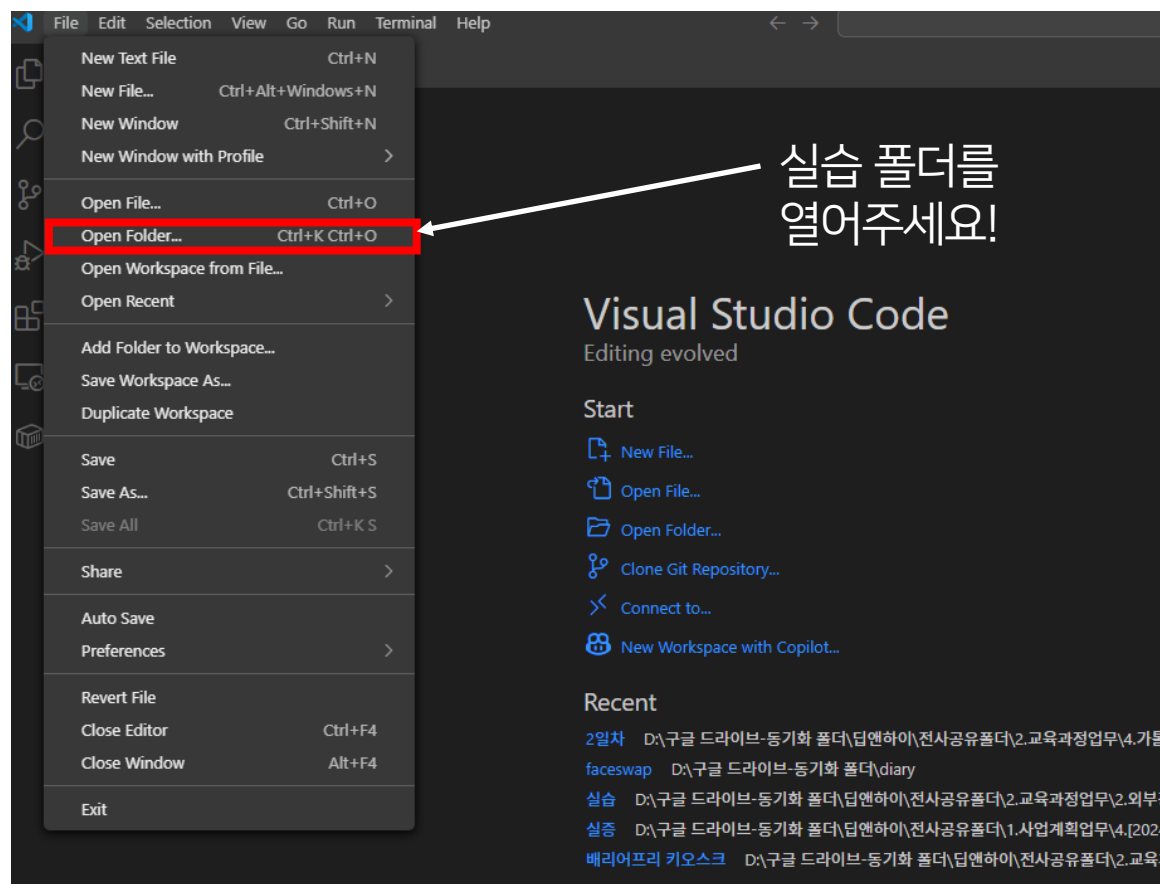


언어 모델



음성인식 및 출력

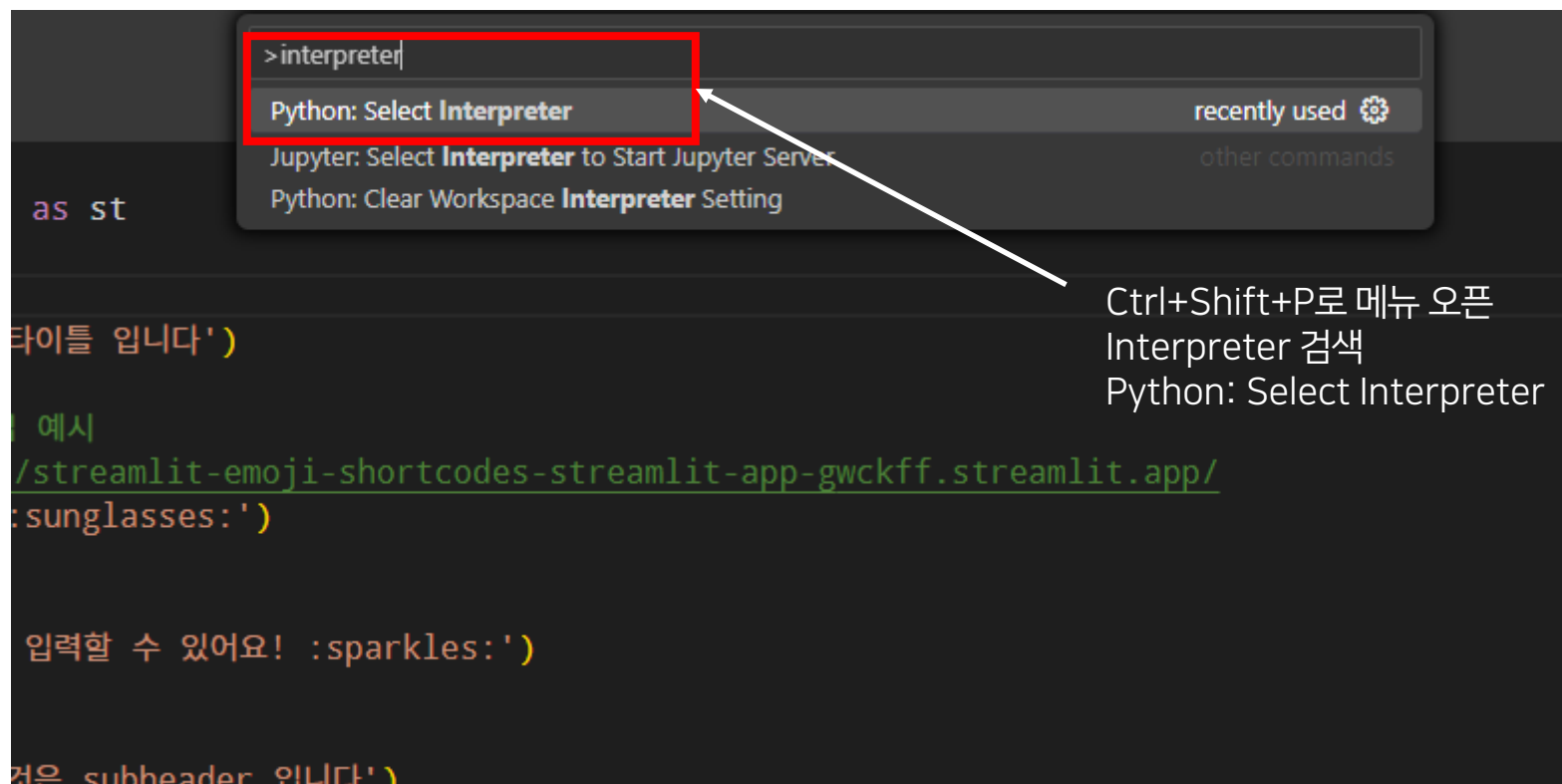
VSCode와 Anaconda로 개발환경 세팅하기



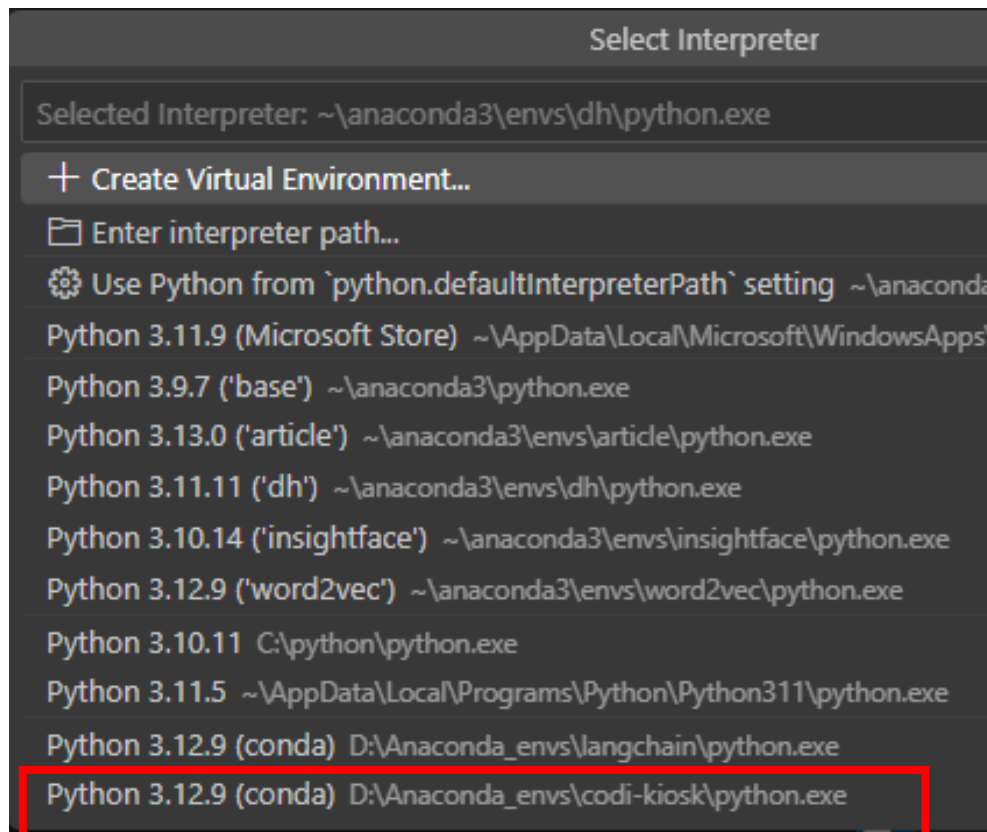
VSCode와 Anaconda로 개발환경 세팅하기



VSCode와 Anaconda로 개발환경 세팅하기

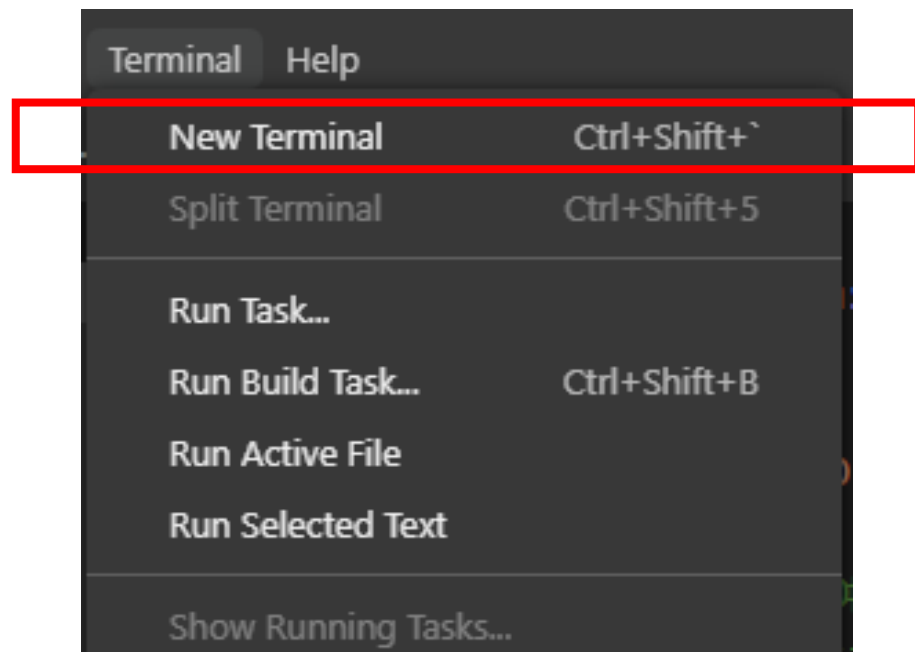


VSCode와 Anaconda로 개발환경 세팅하기



codi-kiosk 선택

◈ Streamlit으로 화면을 만들어 보자!



Streamlit으로 화면을 만들어 보자!

streamlit run 00-text.py 입력

```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

(D:\Anaconda_envs\codi-kiosk) D:\구글 드라이브-동기화 폴더\코드\2일차>Streamlit run 00-text.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.5:8501
```

여러분의 Local URL로 웹 페이지를 열 수 있어요!

Streamlit으로 화면을 만들어 보자!

이것은 타이틀 입니다

스마일 🕶️

헤더를 입력할 수 있어요! ✨

이것은 subheader 입니다

캡션을 한 번 넣어 봤습니다

```
def function():  
    print('hello, world')
```

일반적인 텍스트를 입력해 보았습니다.

streamlit은 마크다운 문법을 지원합니다.

텍스트의 색상을 초록색으로, 그리고 파란색 볼드체로 설정할 수 있습니다.

$\sqrt{x^2 + y^2} = 1$ 와 같이 latex 문법의 수식 표현도 가능합니다 📐

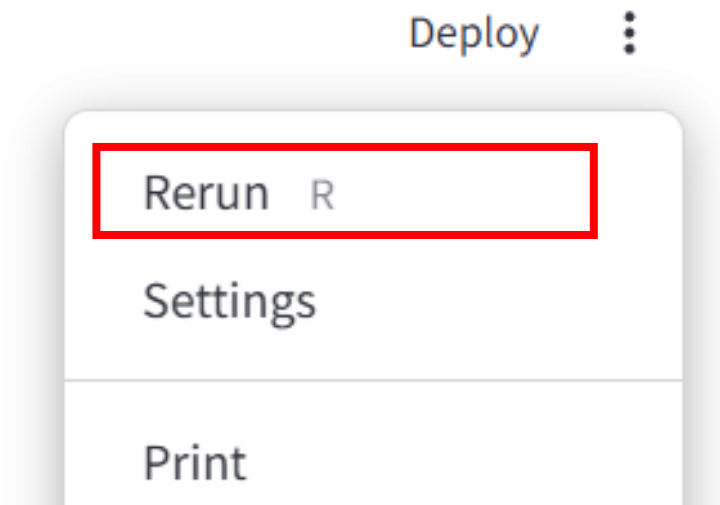
$$\sqrt{x^2 + y^2} = 1$$



```
1 import streamlit as st  
2  
3 # 타이틀 적용 예시  
4 st.title('이것은 타이틀 입니다')  
5  
6 # 특수 이모티콘 삽입 예시  
7 # emoji: https://streamlit-emoji-shortcodes-streamlit-app  
8 st.title('스마일 :sunglasses:')  
9  
10 # Header 적용  
11 st.header('헤더를 입력할 수 있어요! :sparkles:')  
12  
13 # Subheader 적용  
14 st.subheader('이것은 subheader 입니다')  
15  
16 # 캡션 적용  
17 st.caption('캡션을 한 번 넣어 봤습니다')  
18  
19 # 코드 표시  
20 sample_code = '''  
21 def function():  
22     print('hello, world')  
23 '''  
24 st.code(sample_code, language="python")  
25
```

웹 페이지와 코드를 비교해 가면서 streamlit 동작 코드를 익혀 보세요!

◈ Streamlit으로 화면을 만들어 보자!



웹 페이지 실행 중에 코드를 고쳐야 한다면,
고친 후 웹 페이지의 Deploy-Rerun 으로 페이지를 다시 로딩해 주세요.

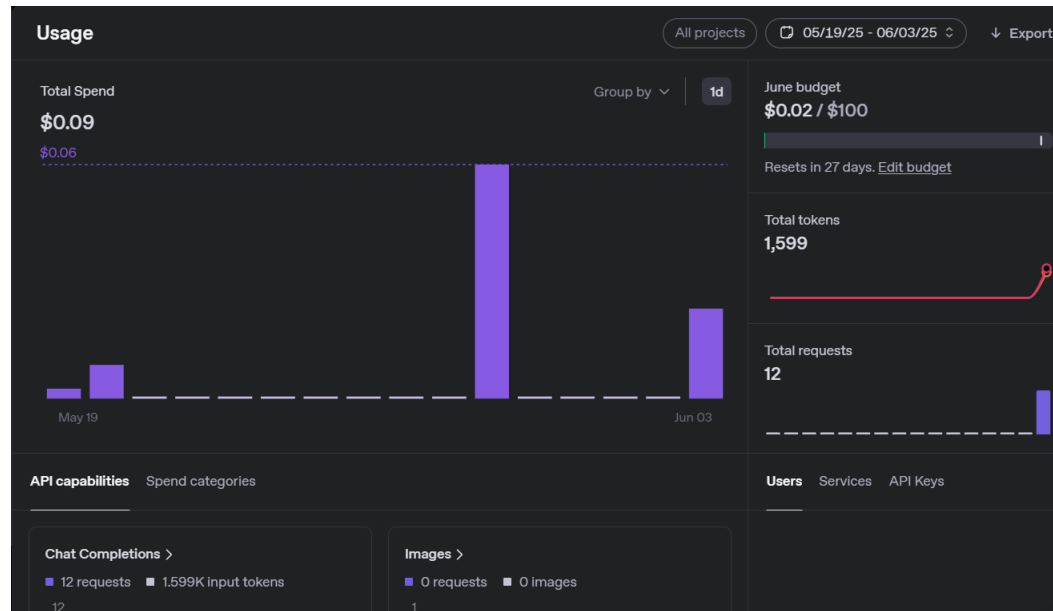
◆ ChatGPT API로 코드에서 활용해 보자!

- API (Application Programming Interface): 소프트웨어 애플리케이션이 서로 통신하며 데이터, 특징, 기능을 교환할 수 있도록 하는 규칙 또는 프로토콜 집합
- ChatGPT API : User Input을 넣으면 ChatGPT의 출력을 포함한 내용을 서버로부터 회신받을 수 있음



ChatGPT API로 코드에서 활용해 보자!

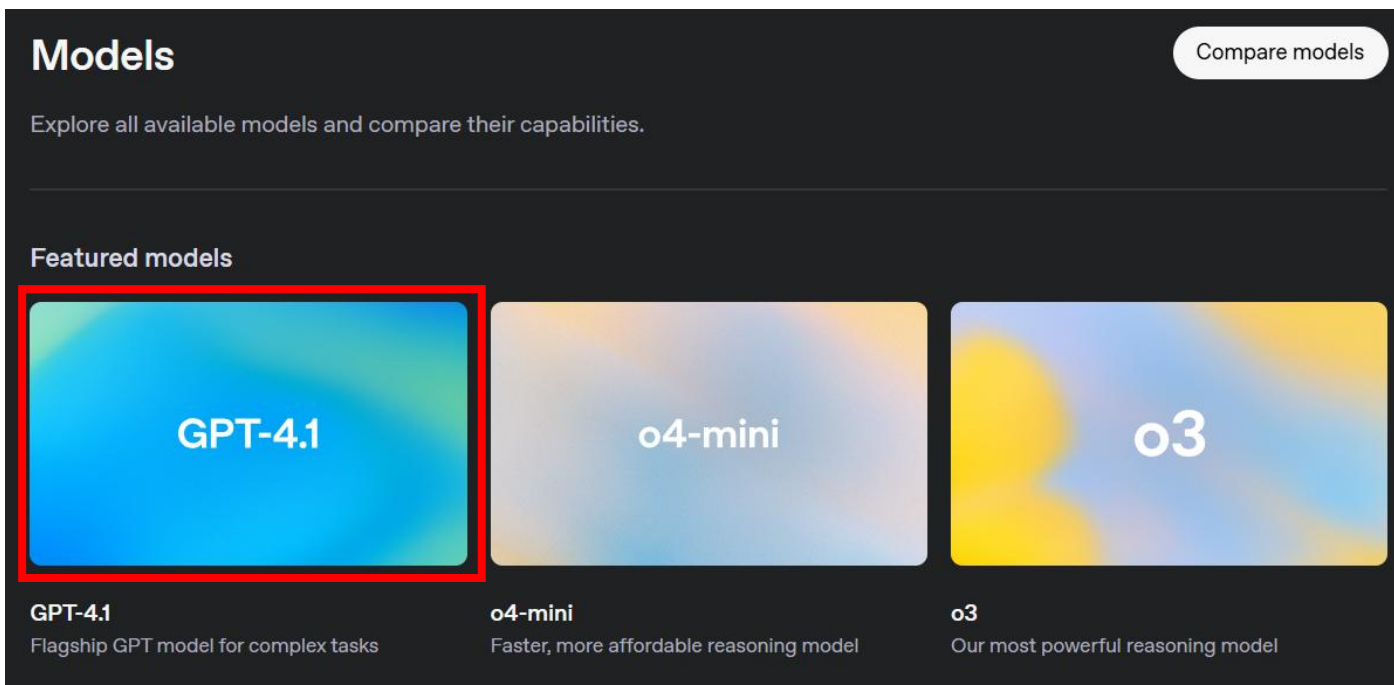
- ChatGPT API Key : User가 서버와 통신할 수 있게 해주는 접근 키 역할.
- 긴 문자열로 정의되어 있으며, 인터넷 접속 비밀번호같이 남들에게 알려주면 안 됨! (사용한 토큰에 비례해서 API 요금이 과금되기 때문)



◈ ChatGPT API로 코드에서 활용해 보자!

- 사용 가능한 GPT 모델 확인 주소 :

https://platform.openai.com/docs/models?utm_source=chatgpt.com



ChatGPT API로 코드에서 활용해 보자!

- 사용 가능한 GPT 모델 확인 주소 :

https://platform.openai.com/docs/models?utm_source=chatgpt.com

The screenshot shows the OpenAI Models page for GPT-4.1. The model is highlighted with a red box. Below the model name, there are five categories: INTELLIGENCE, SPEED, PRICE, INPUT, and OUTPUT. Each category has a visual representation (dots, lightning bolts, or icons) and a text label. At the bottom, there is a description of the model and its capabilities.

< Models

GPT-4.1 GPT-4.1 Flagship GPT-4

Default

✓ gpt-4.1-2025-04-14

Compare Try in Playground

INTELLIGENCE	SPEED	PRICE	INPUT	OUTPUT
Higher	Medium	\$2 · \$8 Input · Output	Text, image	Text

GPT-4.1 is our flagship model for complex tasks. It is well suited for problem solving across domains.

- ✦ 1,047,576 context window
- ➡ 32,768 max output tokens
- 📅 Jun 01, 2024 knowledge cutoff

◈ ChatGPT API로 코드에서 활용해 보자!

```
# 토큰 정보로드
load_dotenv(dotenv_path="./.env")
client = OpenAI()
completion = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {
            "role": "system",
            "content": "당신은 파이썬 프로그래머입니다.",
        },
        {
            "role": "user",
            "content": "피보나치 수열을 생성하는 파이썬 프로그램을 작성해주세요.",
        },
    ],
)
```

<System Prompt>

기본적인 프롬프트 외에, ChatGPT에게
추가적으로 넣을 수 있는 지침들

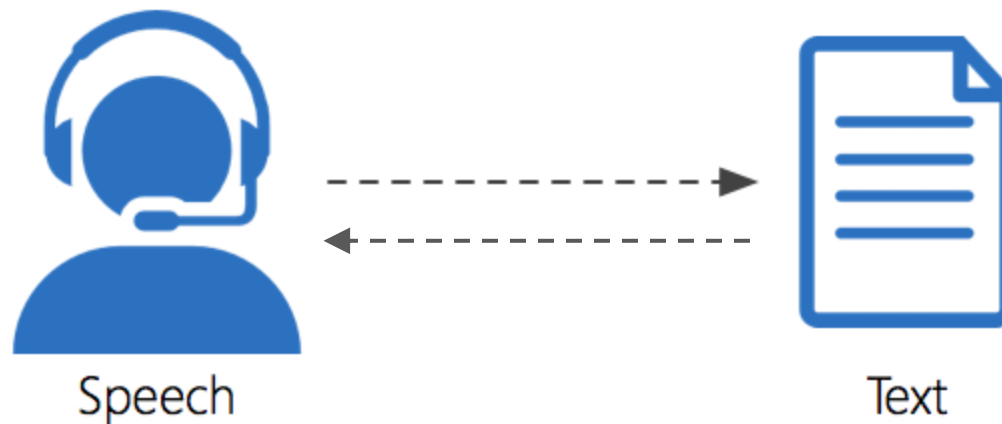
자유롭게 지침 전달 가능

단, 간결하고 논리적으로 작성

작성할 프롬프트와 충돌하면 안 됨

◈ Whisper를 이용해서 음성과 텍스트를 연결시켜 보자!

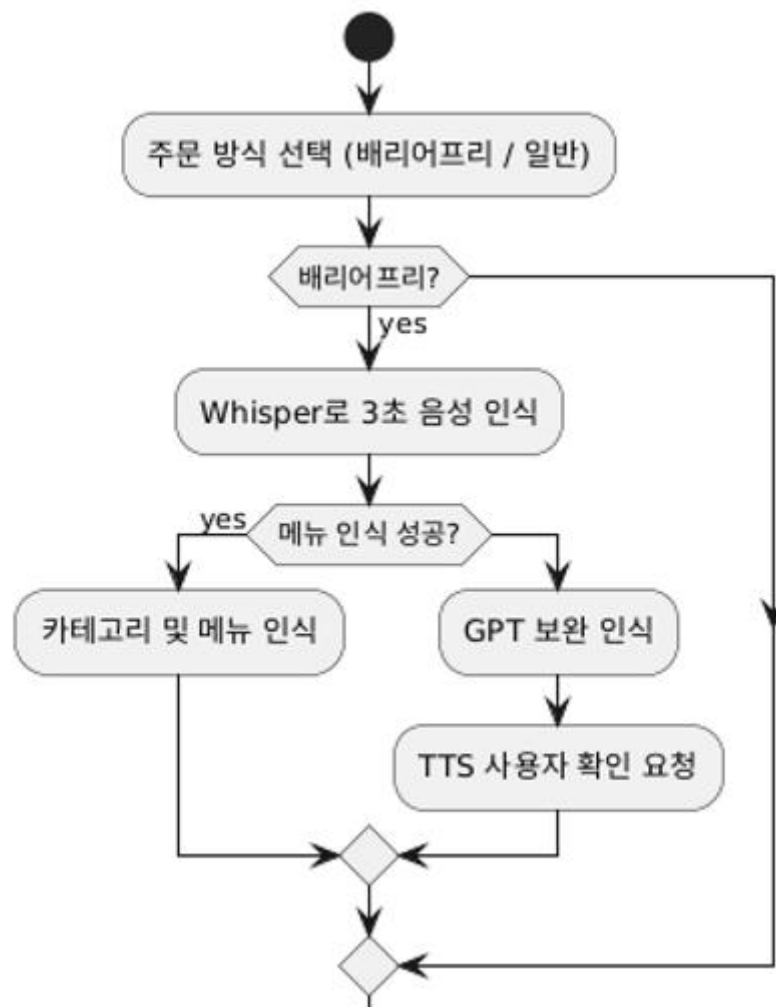
- STT: Speak-to-text의 약자. 음성 데이터를 텍스트로 변환. Whisper 사용
- TTS: Text-to-speak의 약자. 텍스트 데이터를 음성으로 변환하여 출력. pyttsx3 사용



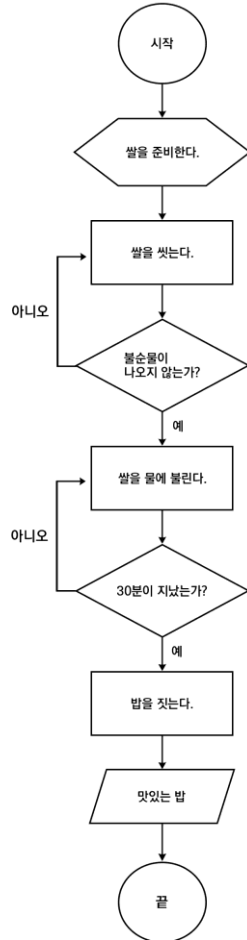
❖ 키오스크를 만들려면 무엇부터 해야할까?



◆ 프로그램 만들기의 기본, 플로우차트 (Flowchart)



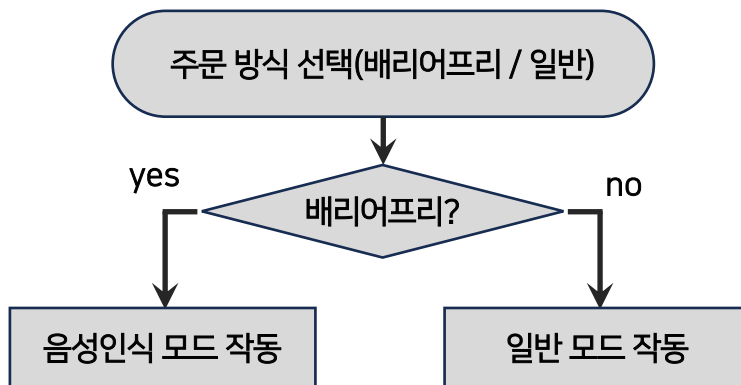
◆ 프로그램 만들기의 기본, 플로우차트 (Flowchart)



의미	도형	설명
시작/종료 (Terminator)	타원형	프로그램 시작과 끝을 나타냅니다.
처리 (Process)	직사각형	연산, 처리, 동작 등을 나타냅니다.
준비 (Prepare)	육각형	작업 시작 전 해야 할 작업을 나타냅니다.
입력/출력 (I/O)	평행사변형	사용자 입력, 결과 출력 등을 나타냅니다.
조건/판단 (Decision)	마름모	예/아니오, 조건 분기 처리에 사용합니다.
흐름 (Flow)	화살표	다음 단계로 진행 방향을 나타냅니다.

◆ 팀별 실습

- 여러분들이 구현해 볼 배리어프리 키오스크의 플로우차트를 그려 보세요.
- 처음 배리어프리 선택 화면까지 주어집니다. 이후의 차트를 완성하세요.
- 음성인식 모드와 일반 모드는 동일하게 작동해야 합니다.
- 카페여도 좋고, 식당이어도 좋습니다. 여러분의 가게와 상품을 만들어 보세요.
- 만든 후 강사 선생님들께 확인받아 주세요!



수업 끝!
수고하셨습니다 ^-^