

**Large Language Model**

# ChatGPT를 활용한 프로그래밍

강사 양석환



# 숫자 맞추기 게임 만들기



- 프로그램 설명

- 1부터 100까지의 임의의 수를 생성하고 생성된 임의의 수를 맞추는 게임
- 숫자를 하나 입력하면 임의로 생성된 수보다 높은지 낮은지 정답인지를 알려줌
- 정답을 맞출 경우, 정답을 몇 번 안에 맞췄는지를 이용하여 게임의 승부를 결정함

- Prompt 입력

- 파이썬으로 숫자 맞추기 게임을 만들 거야. 1~100까지 무작위 숫자를 생성하고, 내가 숫자를 입력하면 무작위 숫자랑 비교해서 숫자가 큰지 작은지 알려줘. 숫자를 맞추면 몇 회 만에 맞췄는지 알려주고 게임을 종료해



파이썬으로 숫자 맞추기 게임을 만들 거야. 1~100까지 무작위 숫자를 생성하고, 내가 숫자를 입력하면 무작위 숫자랑 비교해서 숫자가 큰지 작은지 알려줘. 숫자를 맞추면 몇 회 만에 맞췄는지 알려주고 게임을 종료해



아래는 파이썬으로 숫자 맞추기 게임을 만드는 예제 코드입니다. 이 코드는 1부터 100까지의 무작위 숫자를 생성하고, 사용자로부터 입력을 받아 숫자를 맞추는 게임입니다. 숫자를 맞출 때까지 몇 회 시도했는지 알려줍니다.

```
python Copy code

import random

# 1부터 100까지의 무작위 숫자 생성
target_number = random.randint(1, 100)

# 시도 횟수 초기화
attempts = 0

while True:
    try:
        # 사용자로부터 숫자 입력 받기
        guess = int(input("1부터 100까지의 숫자를 입력하세요: "))
        attempts += 1

        # 입력한 숫자와 목표 숫자 비교
        if guess < target_number:
            print("더 큰 숫자를 입력하세요.")
        elif guess > target_number:
            print("더 작은 숫자를 입력하세요.")
        else:
            print(f"축하합니다! {attempts}회 만에 숫자를 맞췄습니다.")
            break # 게임 종료

    except ValueError:
        print("올바른 숫자를 입력하세요.")

# 게임 종료 메시지
print("게임 종료")
```

이 코드를 실행하면 사용자가 숫자를 입력할 때까지 계속해서 비교를 하고, 숫자를 맞출 때까지 반복합니다. 숫자를 맞췄을 때, 몇 회만에 맞추었는지를 알려준 후 게임을 종료합니다. 사용자가 숫자가 아닌 다른 입력을 할 경우에 대비해서 예외 처리도 추가되어 있습니다.

```
import random
```

```
# 1부터 100까지의 무작위 숫자 생성
target_number = random.randint(1, 100)
```

```
# 시도 횟수 초기화
attempts = 0
```

```
while True:
```

```
    try:
```

```
        # 사용자로부터 숫자 입력 받기
```

```
        guess = int(input("1부터 100까지의 숫자를 입력하세요: "))
```

```
        attempts += 1
```

```
# 입력한 숫자와 목표 숫자 비교
```

```
if guess < target_number:
```

```
    print("더 큰 숫자를 입력하세요.")
```

```
elif guess > target_number:
```

```
    print("더 작은 숫자를 입력하세요.")
```

```
else:
```

```
    print(f"축하합니다! {attempts}회 만에 숫자를 맞췄습니다.")
```

```
    break # 게임 종료
```

```
except ValueError:
```

```
    print("올바른 숫자를 입력하세요.")
```

```
# 게임 종료 메시지
```

```
print("게임 종료")
```

```
Windows PowerShell
PS C:\workspace> python -m venv numgame
PS C:\workspace> cd .\numgame\
PS C:\workspace\numgame> .\Scripts\activate
(numgame) PS C:\workspace\numgame> python
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>>
>>> # 1부터 100까지의 무작위 숫자 생성
>>> target_number = random.randint(1, 100)
>>>
>>> # 시도 횟수 초기화
>>> attempts = 0
>>>
>>> while True:
...     try:
...         # 사용자로부터 숫자 입력 받기
...         guess = int(input("1부터 100까지의 숫자를 입력하세요: "))
...         attempts += 1
...
...         # 입력한 숫자와 목표 숫자 비교
...         if guess < target_number:
...             print("더 큰 숫자를 입력하세요.")
...         elif guess > target_number:
...             print("더 작은 숫자를 입력하세요.")
...         else:
...             print(f"축하합니다! {attempts}회 만에 숫자를 맞췄습니다.")
...             break # 게임 종료
...
...     except ValueError:
...         print("올바른 숫자를 입력하세요.")
...
>>>
```

```
...
1부터 100까지의 숫자를 입력하세요: # 게임 종료 메시지
올바른 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: print("게임 종료")
올바른 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 37
더 큰 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 70
더 큰 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 85
더 작은 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 80
더 작은 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 75
더 작은 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 73
더 큰 숫자를 입력하세요.
1부터 100까지의 숫자를 입력하세요: 74
축하합니다! 7회 만에 숫자를 맞췄습니다.
>>> |
```

# GUI 기반 음악 재생 프로그램 만들기



- 프로그램 설명
  - 음악 파일을 재생하는 GUI 기반의 프로그램을 만들자
- Prompt 입력
  - 파이썬으로 음악을 재생하는 GUI 프로그램을 만들어줘



파이썬으로 음악을 재생하는 GUI 프로그램을 만들어줘



음악을 재생하는 GUI 프로그램을 만들기 위해서는 Python의 GUI 라이브러리 중 하나를 사용해야 합니다. 여기서는 Tkinter를 사용한 간단한 예제를 제공하겠습니다. 먼저, 'tkinter'와 'pygame' (음악 재생을 위한 라이브러리)를 설치해야 합니다. 다음은 Tkinter를 사용하여 음악을 재생하는 간단한 프로그램의 예제입니다:

먼저, 필요한 라이브러리를 설치합니다:

```
bash
```

Copy code

```
pip install pygame
```

그런 다음, 아래의 코드를 Python 파일로 저장하고 실행합니다.

```
python
```

Copy code

```
import tkinter as tk
from tkinter import filedialog
import pygame

# Tkinter 초기화
root = tk.Tk()
root.title("음악 재생기")

# Pygame 초기화
pygame.mixer.init()

# 음악 파일 선택 함수
def open_file():
    file_path = filedialog.askopenfilename(filetypes=[("음악 파일", "*.mp3 *.mp2 *.wav")])
    if file_path:
        pygame.mixer.music.load(file_path)
```

```
# 재생 함수
def play_music():
    pygame.mixer.music.play()

# 일시 정지 함수
def pause_music():
    pygame.mixer.music.pause()

# 정지 함수
def stop_music():
    pygame.mixer.music.stop()

# GUI 요소 생성
open_button = tk.Button(root, text="파일 열기", command=open_file)
play_button = tk.Button(root, text="재생", command=play_music)
pause_button = tk.Button(root, text="일시 정지", command=pause_music)
stop_button = tk.Button(root, text="정지", command=stop_music)

open_button.pack()
play_button.pack()
pause_button.pack()
stop_button.pack()

root.mainloop()
```

이 코드는 Tkinter를 사용하여 간단한 GUI를 만들고, 사용자가 음악 파일을 선택하고 재생, 일시 정지, 정지할 수 있도록 합니다. 파일을 열기 위해서는 "파일 열기" 버튼을 사용하고, "재생", "일시 정지", "정지" 버튼을 사용하여 음악을 제어할 수 있습니다.

물론, 이것은 기본적인 예제일 뿐이며, 보다 복잡한 음악 재생기를 만들려면 더 많은 기능과 개선이 필요할 것입니다.



```
pip install pygame
```

```
import tkinter as tk
from tkinter import filedialog
import pygame
```

```
# Tkinter 초기화
root = tk.Tk()
root.title("음악 재생기")
```

```
# Pygame 초기화
pygame.mixer.init()
```

```
# 음악 파일 선택 함수
```

```
def open_file():
    file_path = filedialog.askopenfilename(
        filetypes=[("음악 파일", "*.mp3 *.wav")])
    if file_path:
        pygame.mixer.music.load(file_path)
```

```
# 재생 함수
```

```
def play_music():
    pygame.mixer.music.play()
```

```
# 일시 정지 함수
```

```
def pause_music():
    pygame.mixer.music.pause()
```

```
# 정지 함수
```

```
def stop_music():
    pygame.mixer.music.stop()
```

```
# GUI 요소 생성
```

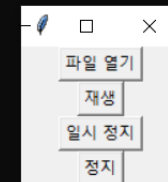
```
open_button = tk.Button(root, text="파일 열기", command=open_file)
play_button = tk.Button(root, text="재생", command=play_music)
pause_button = tk.Button(root, text="일시 정지", command=pause_music)
stop_button = tk.Button(root, text="정지", command=stop_music)
```

```
open_button.pack()
play_button.pack()
pause_button.pack()
stop_button.pack()
```

```
root.mainloop()
```

```
Windows PowerShell
.metadata
  Downloading pygame-2.5.1-cp311-cp311-win_amd64.whl.metadata (13 kB)
Downloading pygame-2.5.1-cp311-cp311-win_amd64.whl (10.6 MB)
10.6/10.6 MB 43.7 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.5.1
(musicplay) PS C:\workspace\musicplay> python
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tkinter as tk
>>> from tkinter import filedialog
>>> import pygame
pygame 2.5.1 (SDL 2.28.2, Python 3.11.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
>>>
>>> # Tkinter 초기화
>>> root = tk.Tk()
>>> root.title("음악 재생기")
>>>
>>> # Pygame 초기화
>>> pygame.mixer.init()
>>>
>>> # 음악 파일 선택 함수
>>> def open_file():
...     file_path = filedialog.askopenfilename(filetypes=[("음악 파일", "*.mp3 *.wav")])
...     if file_path:
...         pygame.mixer.music.load(file_path)
... 
```

```
>>> # 재생 함수
>>> def play_music():
...     pygame.mixer.music.play()
...
>>> # 일시 정지 함수
>>> def pause_music():
...     pygame.mixer.music.pause()
...
>>> # 정지 함수
>>> def stop_music():
...     pygame.mixer.music.stop()
...
>>> # GUI 요소 생성
>>> open_button = tk.Button(root, text="파일 열기", command=open_file)
>>> play_button = tk.Button(root, text="재생", command=play_music)
>>> pause_button = tk.Button(root, text="일시 정지", command=pause_music)
>>> stop_button = tk.Button(root, text="정지", command=stop_music)
>>>
>>> open_button.pack()
>>> play_button.pack()
>>> pause_button.pack()
>>> stop_button.pack()
>>>
>>> root.mainloop()
```



THANK  
YOU

