

# [ 1-1. ChatGPT, 파이썬 이용에 활용하는 방법 ]

## 1. 수업 소개

- 파이썬은 최근 데이터분석 분야에서 사용 빈도가 매우 높은 프로그래밍 언어 입니다.
- 다른 프로그래밍 언어 대비 상대적으로 배우기 쉽고, 라이브러리가 다양하게 제공되어 활용도가 높습니다.
- 프로그래밍 자체가 목적이 아닌, 통계학, 금융공학, 빅데이터분석 등의 분야에서 활용도가 높습니다.
  - 프로그래밍 자체에 노력을 줄이고,
  - 본래 얻고자 하는 그 목적에 초점을 맞출 수 있기 때문
- 최근에 공개된 ChatGPT는 파이썬 활용에 크게 도움이 되고 있습니다.
- 프로그래밍 자체가 목적이 아닌 경우 그 디테일한 코딩 기법들을 항상 숙지하고 있기 어렵습니다.
- 큰 틀에서 파이썬 사용법을 이해하고 있는 상태에서 디테일한 사용 기법은 ChatGPT에게 물어보면서 코딩하면 코딩 스킬을 크게 확대할 수 있습니다.

## 2. 파이썬 소개

파이썬(Python)은 1991년 네덜란드계 소프트웨어 엔지니어인 귀도 반 로섬(Guido van Rossum)에 의해 처음 공개된 프로그래밍 언어 입니다. 이 프로그래밍 언어는 코드의 가독성과 간결성을 매우 중시합니다. 파이썬의 문법 체계와 코딩 철학을 따라서 코딩을 하면 자연스럽게 코드가 간결해지고, 읽기 쉬워질 수밖에 없도록 설계되어 있습니다.

파이썬이 다른 프로그래밍 언어에 비해 활용도가 높은 특징은 아래와 같습니다.

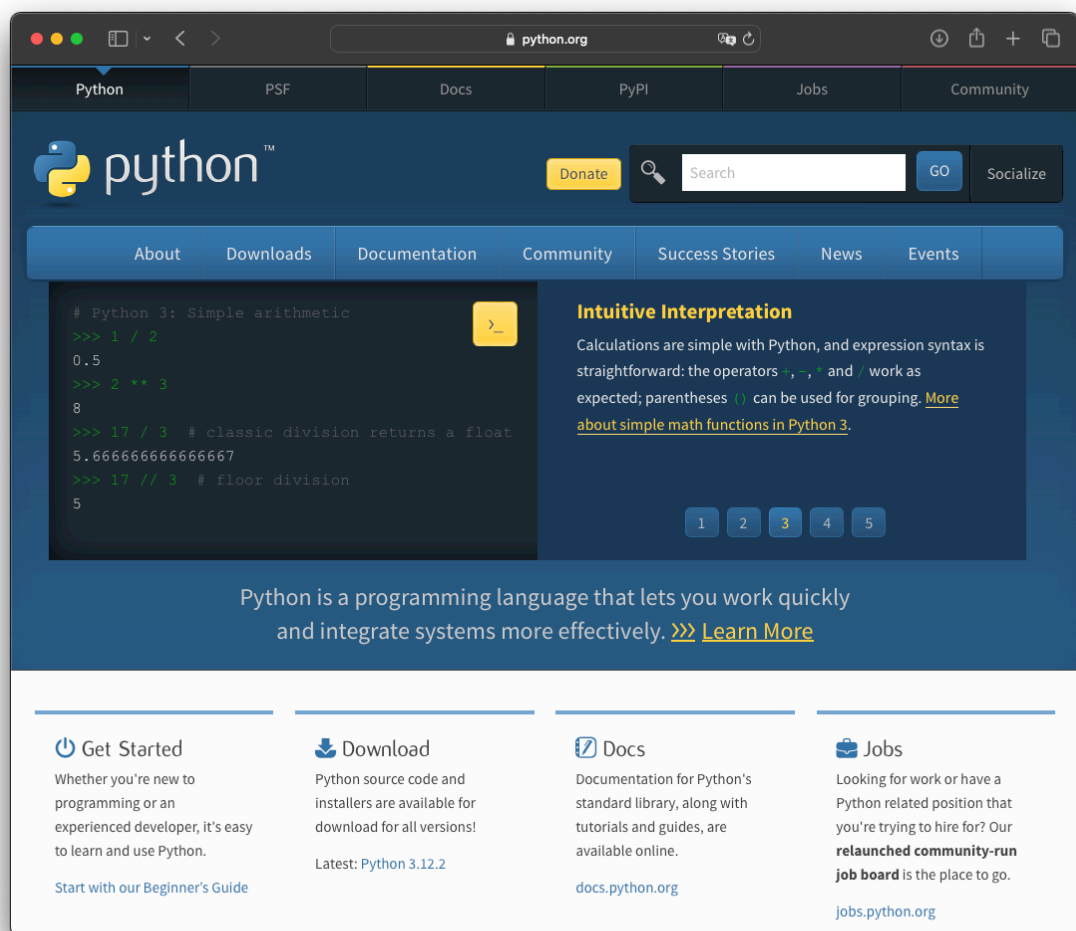
- 가독성과 간결성 :
  - 파이썬은 읽기 쉽고 이해하기 쉬운 문법 구조를 가지고 있습니다.
  - 이는 개발자가 더 적은 코드로 더 많은 작업을 할 수 있게 해주며, 코드의 유지 보수가 용이하다는 것을 의미합니다.
- 다양한 라이브러리와 프레임워크 :
  - 파이썬은 과학 계산, 데이터 분석, 인공지능, 웹 개발, 자동화 등 다양한 분야에서 사용될 수 있는 방대한 라이브러리와 프레임워크를 제공합니다.
  - 이는 파이썬의 기초 문법만 알아도 전세계에서 그 누군가가 이미 작성해 놓은 라이브러리와 프레임워크를 이용하여 수준 높은 결과물을 얻을 수 있음을 의미합니다.
  - 예를 들어, 데이터과학 관련하여 Pandas, NumPy, SciPy, Matplotlib, TensorFlow, PyTorch 등이 있습니다.
- 범용성 :
  - 파이썬은 스크립팅, 웹 개발, 데이터 분석, AI 개발 등 다양한 용도로 사용될 수 있는 범용 프

로그래밍 언어입니다.

- 이로 인해 개발자는 하나의 언어로 다양한 문제를 해결할 수 있습니다.
- 커뮤니티 지원 :
  - 파이썬은 엄청난 규모의 활발한 커뮤니티를 보유하고 있으며, 이는 다양한 문제에 대한 해결책, 튜토리얼, 라이브러리, 프레임워크 등을 쉽게 찾을 수 있음을 의미합니다.
  - 파이썬은 파이썬 소프트웨어 재단(Python Software Foundation, [www.python.org](http://www.python.org))에서 관리하고 있으며, 무료나 다름없는 Python Software Foundation을 따르고 있습니다.
- 플랫폼 독립성 :
  - 파이썬은 크로스 플랫폼 언어로서, 윈도우, 맥OS, 리눅스 등 다양한 운영체제에서 실행될 수 있습니다.
  - 코드 한번 작성으로 다양한 환경에서 실행 가능합니다.

이러한 특성 때문에 파이썬은 프로그래밍을 처음 배우는 사람들부터 전문 개발자까지 널리 사용되고 있습니다.

## [ 파이썬 소프트웨어 재단 홈페이지 ]



## [[ 참고 : 라이브러리와 프레임워크 ]]

라이브러리(Library)와 프레임워크(Framework)는 소프트웨어 개발에서 코드를 재사용하기 위해 사용되는 중요한 도구입니다. 이 둘은 비슷해 보이지만, 라이브러리를 사용할 때는 개발자가 애플리케이션의 흐름을 제어하고 필요에 따라 라이브러리를 호출합니다. 반면, 프레임워크는 프레임워크 자체가 애플리케이션의 흐름을 스스로 관리하며, 개발자는 프레임워크가 요구하는 특정 부분을 구현합니다.

### 라이브러리(Library)

라이브러리는 특정 기능을 수행하기 위해 미리 작성된 코드의 집합입니다. 개발자는 라이브러리를 사용하여 자신의 프로그램에 특정 기능을 추가할 수 있습니다. 예를 들어, 수학 연산, 문자열 관리, 네트워크 등의 작업을 위한 함수와 메서드를 포함할 수 있습니다. 라이브러리는 개발자가 필요할 때 호출하여 사용하는 형태로, 개발자가 프로그램의 흐름을 완전히 제어합니다. 즉, 라이브러리는 단순히 도구를 제공하며, 언제 어떻게 사용할지는 개발자가 결정합니다.

### 프레임워크(Framework)

프레임워크는 소프트웨어의 특정 부분을 개발하기 위한 기반 "구조" 또는 "골격"입니다. 프레임워크는 애플리케이션 개발에 있어 일관된 구조와 패턴을 제공합니다. 프레임워크는 라이브러리를 포함할 수 있지만, 그 이상의 것을 제공합니다. 즉, 애플리케이션의 기본 흐름과 구조를 정의하며, 개발자는 프레임워크가 정의한 규칙과 API를 사용하여 애플리케이션을 구축합니다. 프레임워크가 애플리케이션의 흐름을 제어하며 개발자는 프레임워크가 요구하는 부분을 구현합니다.

### 애플리케이션(Application)

사용자가 특정 작업을 수행하기 위해 사용하는 컴퓨터 소프트웨어 프로그램.

## 3. ChatGPT 소개

### [ ChatGPT에 물어본 ChatGPT 소개 및 파이썬 활용 유용성 ]

**You**

ChatGPT에대해서 소개하고, ChatGPT가 파이썬 코딩에 얼마나 유용한지 설명해줘

< 2 / 2 >

**ChatGPT**

ChatGPT는 OpenAI에 의해 개발된 대화형 인공지능(AI) 모델로, 자연어 처리(NLP)를 통해 사용자의 질문이나 명령에 대해 인간처럼 응답할 수 있습니다. 이 AI는 다양한 주제에 대한 지식을 바탕으로 대화를 진행하며, 사용자의 요구에 따라 정보를 제공하거나 문제를 해결하는 데 도움을 줍니다. GPT (Generative Pretrained Transformer)의 최신 버전을 기반으로 하며, 대규모 데이터셋에서 사전 학습된 후 다양한 사용자 입력에 대응할 수 있도록 특별히 최적화되었습니다.

### ChatGPT의 파이썬 코딩에 대한 유용성

1. **코딩 도움말:** ChatGPT는 파이썬 프로그래밍에 있어서 문법적인 질문, 코드 스니펫 제공, 디버깅 팁 등을 제공함으로써 프로그래머들에게 코딩 과정에서 발생할 수 있는 다양한 문제를 해결하는 데 도움을 줍니다.
2. **학습 자료:** 파이썬 학습자에게 기본 개념 설명, 고급 주제에 대한 심화 자료, 최신 파이썬 트렌드 및 라이브러리 사용법 등을 제공하여 지속적인 학습을 지원합니다.
3. **프로젝트 아이디어 및 가이드라인 제공:** 사용자가 새로운 프로젝트를 시작할 때 아이디어를 제공하고, 해당 프로젝트를 진행하기 위한 기술적인 가이드라인이나 설계 조언을 제공할 수 있습니다.
4. **코드 최적화 및 리팩토링:** 작성된 코드의 효율성을 높이거나 가독성을 개선하기 위한 방법을 제안함으로써, 더 나은 코드 작성을 돕습니다.
5. **라이브러리 및 프레임워크 사용법:** 다양한 파이썬 라이브러리와 프레임워크에 대한 사용법, 예제 코드, 그리고 최적의 활용 방안을 제공합니다.
6. **에러 메시지 해석 및 해결:** 코딩 중 발생한 에러 메시지를 해석하고, 가능한 해결책을 제시하여 문제를 빠르게 해결할 수 있도록 돕습니다.

종합적으로, ChatGPT는 파이썬 코딩에 있어서 다양한 단계에서 유용한 자원이 될 수 있습니다. 이는 학습부터 실제 코딩, 문제 해결에 이르기까지 폭넓게 활용될 수 있으며, 개발자들이 보다 효과적으로 코드를 작성하고, 지식을 확장하는 데 기여할 수 있습니다.



## [ ChatGPT의 한계 ]

- **지식의 시기적 한계 :** ChatGPT는 특정 시점까지의 데이터로 훈련되므로, 그 이후에 발생한 사건이나 최신 연구 결과에 대해서는 알지 못합니다. 이는 모델이 최신 정보를 제공하는 데 한계가 있다는 것을 의미합니다.
- **환각 생성 :** ChatGPT는 때때로 존재하지 않는 사실이나 데이터를 생성하는 "환각"을 보일 수 있습니다. 이는 모델이 데이터를 잘못 해석하거나, 훈련 과정에서 학습한 잘못된 패턴을 반복하기 때문에 발생합니다.
- **객관성 및 편향 :** ChatGPT는 훈련 데이터에 존재하는 객관성의 부족이나 편향을 반영할 수 있습니다. 이는 모델이 특정 주제에 대해 편향된 시각을 제시하거나, 특정 집단에 대한 부정확하거나 부적절한 표현을 생성할 위험이 있음을 의미합니다.

- **창의성과 혁신의 한계** : ChatGPT는 기존의 정보와 패턴을 기반으로 답변을 생성할 수 있지만, 진정한 의미에서 새롭거나 혁신적인 아이디어를 생성하는 데는 한계가 있습니다. 모델은 기존에 알려진 정보의 재구성에는 능숙하지만, 새로운 발견이나 창조적인 아이디어를 독자적으로 생각해 내는 데는 제한적입니다.

## [ ChatGPT의 한계에도 불구하고 파이썬 코딩에는 매우 유용함 ]

### - 문법 활용 방법 및 코드 예시 제공

- ChatGPT는 파이썬 문법의 활용 방법 및 예시를 제공합니다.
- 전문적으로 프로그래밍을 하는 사람이 아니고서는 매번 디테일한 코딩 문법을 숙지하고 익숙하게 활용하는 데에는 제한이 있습니다.
- 문법의 기본 틀만 알고 있으며, 그때 그때 활용시마다 ChatGPT에게 물어보고 그 사용 예시를 이용해서 코딩에 활용할 수 있습니다.



You

CSV파일을 읽어들이는 파이썬 문법 예시 들어줘.



ChatGPT

CSV 파일을 읽어들이는 가장 기본적인 파이썬 코드 예시는 `csv` 모듈을 사용하는 것입니다. 아래 예제는 `csv` 모듈을 사용하여 `example.csv` 파일을 읽고, 각 행을 출력하는 방법을 보여 줍니다.

python

Copy code

```
import csv

# CSV 파일 열기
with open('example.csv', newline='') as csvfile:
    # csv.reader를 사용하여 CSV 파일 읽기
    csvreader = csv.reader(csvfile, delimiter=',')

    # 헤더(첫 번째 줄) 건너뛰기
    next(csvreader)

    # 각 행(row)에 대하여 반복 처리
    for row in csvreader:
        print(row) # 행 출력
```

이 코드는 CSV 파일을 열고, 각 행을 순회하며 출력합니다. `delimiter=','`는 필드가 쉼표로 구분되어 있음을 지정합니다. 파일의 첫 번째 줄이 헤더라고 가정하고 `next(csvreader)`를 사용하여 건너뛵니다.

파이썬에는 또한 `pandas` 라는 라이브러리가 있어, 데이터 분석을 훨씬 용이하게 해주며, CSV 파일을 읽고 데이터프레임(dataframe) 형태로 다루는 것을 매우 간단하게 만들어 줍니다. 아래는 `pandas` 를 사용하여 CSV 파일을 읽는 방법입니다.

python

Copy code

```
import pandas as pd

# pandas를 사용하여 CSV 파일 읽기
df = pd.read_csv('example.csv')

# 데이터프레임 출력
print(df)
```

`pandas` 를 사용하면 데이터 조작, 필터링, 변환 등 다양한 데이터 분석 작업을 효율적으로 수행할 수 있습니다.



## - 특정 문제 해결 방법 및 코드 예시 제공

- 한정적인 문법 영역을 넘어서 특정 문제를 해결하기 위한 코딩 방법 및 예시를 제공합니다.
- 특정 문제를 해결하기 위한 코딩 예시를 제공받아서 이를 실제 코딩에 활용할 수 있습니다.

**You**

주요 재무비율을 분석하고 그 결과를 출력하는 애플리케이션을 하나의 라이브러리 형태로 작성해 줘.

**ChatGPT**

주요 재무비율 분석을 위한 파이썬 라이브러리 형태의 애플리케이션 예제를 제공합니다. 이 라이브러리는 기업의 재무제표 데이터를 입력으로 받아, 주요 재무비율을 계산하고 분석 결과를 출력하는 기능을 제공합니다.

### 재무비율 분석 라이브러리: `finance_analysis.py`

python

Copy code

```
class FinanceAnalysis:
    def __init__(self, income_statement, balance_sheet):
        self.income_statement = income_statement
        self.balance_sheet = balance_sheet

    def current_ratio(self):
        """유동비율 계산"""
        current_assets = self.balance_sheet['Current Assets']
        current_liabilities = self.balance_sheet['Current Liabilities']
        return current_assets / current_liabilities

    def quick_ratio(self):
        """당좌비율 계산"""
        quick_assets = self.balance_sheet['Current Assets'] - self.balance_sheet['Inventory']
        current_liabilities = self.balance_sheet['Current Liabilities']
        return quick_assets / current_liabilities
```

```

def return_on_equity(self):
    """자기자본수익률 계산"""
    net_income = self.income_statement['Net Income']
    equity = self.balance_sheet['Equity']
    return net_income / equity

def debt_to_equity_ratio(self):
    """부채비율 계산"""
    total_debt = self.balance_sheet['Total Debt']
    equity = self.balance_sheet['Equity']
    return total_debt / equity

def print_financial_ratios(self):
    """재무비율 결과 출력"""
    print("Financial Ratios Analysis:")
    print(f"Current Ratio: {self.current_ratio():.2f}")
    print(f"Quick Ratio: {self.quick_ratio():.2f}")
    print(f"Return on Equity (ROE): {self.return_on_equity():.2f}")
    print(f"Debt to Equity Ratio: {self.debt_to_equity_ratio():.2f}")

# 사용 예제
if __name__ == "__main__":
    # 임의의 재무제표 데이터
    income_statement = {
        'Net Income': 15000,
    }
    balance_sheet = {
        'Current Assets': 50000,
        'Current Liabilities': 20000,
        'Inventories': 10000,
        'Equity': 70000,
        'Total Debt': 30000,
    }

    finance_analysis = FinanceAnalysis(income_statement, balance_sheet)
    finance_analysis.print_financial_ratios()

```

이 코드는 `FinanceAnalysis` 클래스를 정의하여, 기업의 재무제표 데이터를 기반으로 유동비율(Current Ratio), 당좌비율(Quick Ratio), 자기자본수익률(Return on Equity, ROE), 부채비율(Debt to Equity Ratio) 등 주요 재무비율을 계산하는 메서드를 포함하고 있습니다. 또한, `print_financial_ratios` 메서드를 통해 계산된 재무비율을 출력합니다.

이 라이브러리는 재무 분석, 투자 의사 결정 지원, 교육 목적 등 다양한 상황에서 유용하게 사용될 수 있습니다. 사용자는 재무제표 데이터를 입력으로 제공하기만 하면, 간단하게 주요 재무비율을 계산하고 분석 결과를 얻을 수 있습니다.

< 2 / 2 >   

## - 디버깅 도움

- ChatGPT는 프로그래밍 중 발생한 오류 메시지를 해석하고, 가능한 해결 방안을 제안해줍니다.



- ChatGPT는 일반적인 프로그래밍 실수나 흔히 발생하는 오류 패턴을 인식하고, 수정 방안을 제시해줍니다.
- 중요한 포인트는 오류 내용을 일일이 개발자가 이해하고 ChatGPT에게 설명해주지 않아도, ChatGPT는 오류의 내용을 이해하고 어디에서 오류가 났는지 설명해주고 이를 해결하기 위한 코드를 제안해준다는 것입니다.

```

KP_Hong — IPython: Users/KP_Hong — ipython — 114x38

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[~ $ ipython
Python 3.11.5 (main, Sep 11 2023, 08:31:25) [Clang 14.0.6 ]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.15.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: people = [
...:     {"이름": "김철수", "나이": 30, "직업": "개발자", "거주지": "서울"},
...:     {"이름": "이영희", "나이": 25, "직업": "디자이너", "거주지": "부산"},
...:     {"이름": "박지민", "나이": 29, "직업": "개발자", "거주지": "대전"},
...:     {"이름": "최예나", "나이": 22, "직업": "마케터", "거주지": "서울"},
...:     {"이름": "정우성", "나이": 35, "직업": "개발자", "거주지": "광주"}
...: ]

In [2]: def filter_by_job(people, job_title):
...:     return [person for person in people if person["직업"] == jobtitle]
...:

In [3]: developers = filter_by_job(people, "개발자")

NameError                                Traceback (most recent call last)
Cell In[3], line 1
----> 1 developers = filter_by_job(people, "개발자")

Cell In[2], line 2, in filter_by_job(people, job_title)
      1 def filter_by_job(people, job_title):
----> 2     return [person for person in people if person["직업"] == jobtitle]

Cell In[2], line 2, in <listcomp>(.0)
      1 def filter_by_job(people, job_title):
----> 2     return [person for person in people if person["직업"] == jobtitle]

NameError: name 'jobtitle' is not defined

In [4]:

```



You

```
In [1]: people = [
...:     {"이름": "김철수", "나이": 30, "직업": "개발자", "거주지": "서울"},
...:     {"이름": "이영희", "나이": 25, "직업": "디자이너", "거주지": "부산"},
...:     {"이름": "박지민", "나이": 29, "직업": "개발자", "거주지": "대전"},
...:     {"이름": "최예나", "나이": 22, "직업": "마케터", "거주지": "서울"},
...:     {"이름": "정우성", "나이": 35, "직업": "개발자", "거주지": "광주"}
...: ]
...:

In [2]: def filter_by_job(people, job_title):
...:     return [person for person in people if person["직업"] == jobtitle]
...:

In [3]: developers = filter_by_job(people, "개발자")
-----
NameError                                Traceback (most recent call last)
Cell In[3], line 1
----> 1 developers = filter_by_job(people, "개발자")

Cell In[2], line 2, in filter_by_job(people, job_title)
      1 def filter_by_job(people, job_title):
----> 2     return [person for person in people if person["직업"] == jobtitle]

Cell In[2], line 2, in <listcomp>(.0)
      1 def filter_by_job(people, job_title):
----> 2     return [person for person in people if person["직업"] == jobtitle]

NameError: name 'jobtitle' is not defined

In [4]:
```



ChatGPT

`filter\_by\_job` 함수에서 사용한 매개변수 이름과 내부에서 사용한 변수 이름이 일치하지 않아서 오류가 발생했습니다. 매개변수 이름은 `job\_title` 인데, 내부에서는 `jobtitle` 로 사용하려고 했습니다. 이를 일치시켜야 합니다. 수정된 함수는 다음과 같습니다:

python

Copy code

```
def filter_by_job(people, job_title):
    return [person for person in people if person["직업"] == job_title]

developers = filter_by_job(people, "개발자")
print(developers)
```

이렇게 수정하면, 함수는 "직업"이 "개발자"인 사람들의 목록을 반환할 것입니다.



## - 프로젝트 아이디어 및 구조 제안

- 새로운 프로젝트를 시작할 때 아이디어를 제공하고, 프로젝트의 기본 구조나 설계 방향에 대한 제안을 할 수 있습니다.
- 이는 프로젝트 계획 단계에서 유용하게 활용될 수 있습니다.

## - 알고리즘과 데이터 구조 설명

- 다양한 알고리즘과 데이터 구조에 대한 설명을 제공하여, 특정 문제를 해결하기 위한 적절한 접근 방법을 이해하는 데 도움이 됩니다.

## 내용 정리

- ChatGPT의 한계를 인지하면서도, 이러한 도구를 적절히 활용한다면 파이썬 코딩 과정을 보다 효율적이고 생산적으로 만들 수 있습니다.
- 특히, 코딩에 활용하는 관점에서는 ChatGPT의 한계로 인한 제한사항이 크지 않다는 것입니다.
- 코딩 문법이 그렇게 실시간으로 변경되는 사항이 아니다 보니 "지식의 시기적 한계"는 이용의 제약사항이 되지 않으며, 코딩에 관한 질의응답에 대해서는 "환각 생성"과 "객관성 및 편향"의 문제는 사용에 제약이 될 정도의 문제가 되지 않습니다.
- 1년 넘게 ChatGPT를 활용해본 결과, ChatGPT를 활용함으로써 개인적인 코딩 실력이 크게 확장되었고, 코딩을 위해 사용되는 시간이 감소하였으며, 더욱더 전문 영역에 시간을 들일 수 있는 여유가 증가하였습니다. ChatGPT는 연봉 몇천만원 수준의 조수 역할을 충분히 해내는 것을 경험하였으며, 앞으로 재무영역과 같은 기존에는 프로그래밍의 활용도가 낮았던 영역에서도 프로그래밍을 활용하는데 크게 기여할 수 있을 것으로 판단합니다.