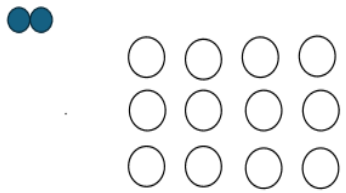
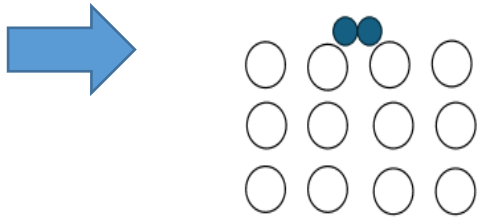
PCT 다이어그램을 그리기 위한 매뉴얼

Yoo Jiheon

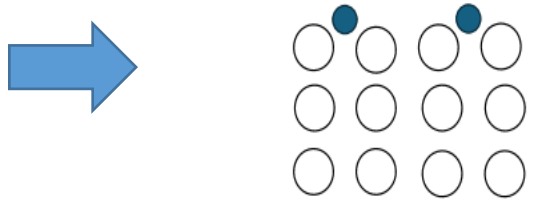
Hydride가 만들어지는 과정



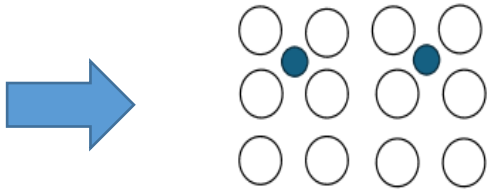
- 기존 상태



- 외부와 금속의 화학퍼텐셜 차이에 의해 수소가 금속 쪽으로 접근하기 시작함
- 화학퍼텐셜이 더 높으면 입자가 들어가기 힘들다.
- 자연은 안정적인 방향으로 이동하려 하기 때문에 화학퍼텐셜이 낮아진 금속 쪽으로 수소를 넣게 된다.



- 수소 분자가 입자로 되기 위해 활성화 에너지를 가해야 한다.
- 또한 입자가 금속 내부로 들어갈 수 있도록 금속과 수소 사이의 장벽을 뚫을 수 있는 금속과 수소 사이의 활성화 에너지도 가해야 한다.



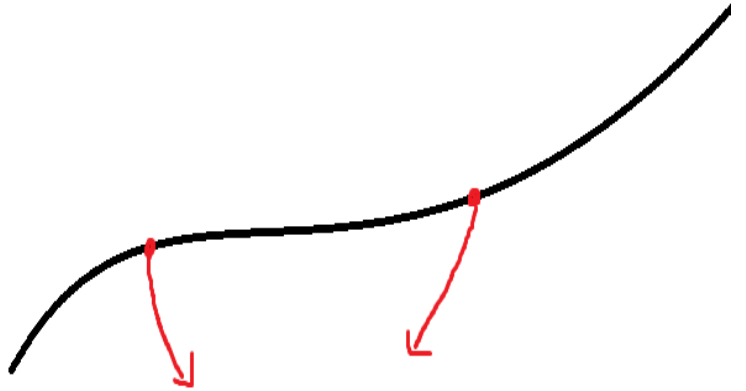
- 수소가 금속 속으로 들어가면서 금속 내부의 부피가 팽창하게 되고 Hydride가 만들어지게 된다.

전환점

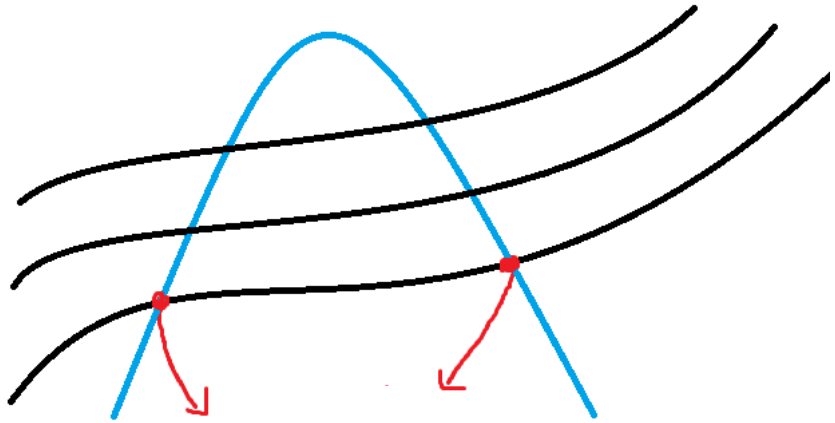
- 상이 전환되기 시작하는 지점
  알파상 -> 평형상태, 평형상태 -> 베타상
  총 2개의 전환점이 존재한다.

- 전환점은 단일상에서의 농도 식과 변형 반트호프 방정식의
  농도와 압력이 같을 때의 지점을 뜻한다.

- 밑에 사진을 보면 온도가 높아질수록 전환점은 점점
  올라가게 되며 평형 상태는 줄어들어
  전환 돔 곡선은 뒤집어진 이차 곡선의 형태를 나타나게 된다.

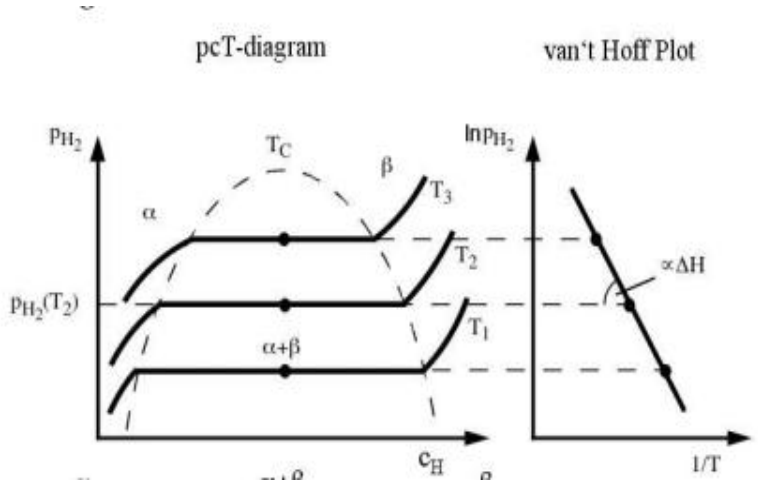


전환점

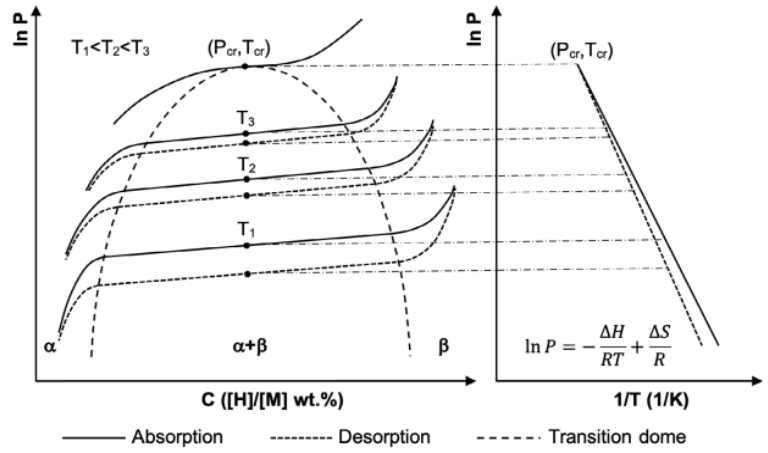


전환점

평형 상태 기울기



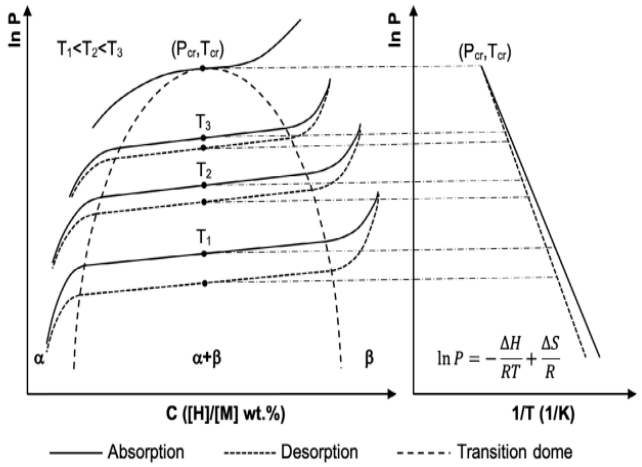
이론적 상태

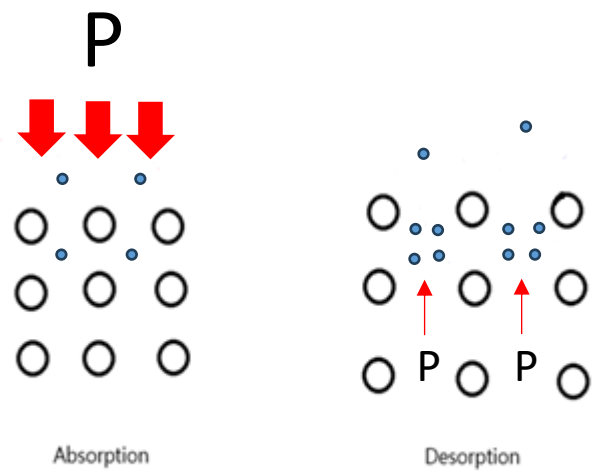


실제 상태

- 왼쪽의 사진들을 보면 이론적 상태에서는 플루토 구간 즉 평형상태에서는 평평함을 볼 수 있지만 실제 상태에서는 약간의 기울기가 있음을 볼 수 있다.
- 그 이유는 화학적 불균일 때문이다.
- 원래는 전환점에서 자연적으로 상이 변해야 하는데 화학적 불균일 때문에 β상으로 수소가 들어가지 않고 밖에 남게 되어 이를 베타상으로 넣어주기 위하여 ΔH를 가해야 한다. 즉 압력을 더 주어야 한다. 또한 수소를 방출할 때에는 ΔH를 낮춰줘야 하기 때문에 압력을 줄여야 한다.
- 이 때의 흡수 곡선과 방출 곡선의 위치가 다른데 이는 히스테리시스로 다음 챕터에서 다루겠다.
- 이 기울기를 고려해서 만든 반트호프 식이 변형 반트호프 식이다.

히스테리시스





- 히스테리시스란 시스템의 상태가 과거의 이력에 따라 달라지는 현상으로 왼쪽 사진을 보면 흡수 곡선과 방출 곡선이 다름을 볼 수 있다.
- 그 이유는 왼쪽 아래의 사진처럼 수소가 흡수 되면서 금속은 부피가 팽창하게 되기 때문에 큰 압력을 주어야하며 방출 때는 이미 팽창 된 공간에서 쉽게 나가기 때문에 적은 압력으로도 방출이 가능해지기 때문에 히스테리시스가 생긴다.
- 히스테리시스를 통해 알 수 있는 점:
  PCT다이어그램을 그릴 때 흡수 곡선과 방출 곡선의 식에서 변수의 값들이 달라진다.

관련 식

- 기울기를 고려한 반트호프 방정식:
- $\ln\frac{P}{Pref} = -\frac{\Delta H}{R \cdot T} + \frac{\Delta S}{R} + fs(C - Cmid)$
- 반트호프 방정식은 평형상태의 압력을 구할 때 사용하게 된다.

- 단일상에서의 농도식:
- $C = A \times P^{(\frac{\gamma}{2})} \times e^{-\frac{\gamma \cdot V \cdot P}{R \cdot T}} \times e^{-\frac{\gamma \cdot H}{R \cdot T}}$

- 전환 돔 곡선:
- $lnP = a_0 + a_1 C + a_2 C^2$ *(특정온도에서만)*
- 전환 돔 곡선에 전환점이 존재하며 곡선의 제일 윗부분은
- 평형상태가 나타나는 마지막 부분 즉 임계점이다.
- 뒤집어진 이차곡선 형태를 띈다.

방법 1: 전환 돔 곡선과 단일상을 비교하는 방법

- $LmNi_{4.9}Sn_{0.15}$의 PCT 다이어그램을 그리기

1. 전환점 찾기

전환 돔 곡선의 식과 단일상의 식에서 농도와 압력값이 같은 지역을 전환점으로 삼는다.

2. 알파상에서

알파상은 알파-평형 전환점 이전의 상태이며 압력값을 사용할 때에는 반트호프 식으로 구한 압력과 단일상으로 구한 압력 중 작은값을 사용한다.

3. 평형상태에서

평형상태는 알파-평형 전환점 이상 평형-베타 전환점 이하 구간이며 이 때에는 반트호프 식만으로 압력을 구한다.

4. 베타상태에서

베타상은 평형-베타 전환점 이후의 상태이며 압력값을 사용할 때에는 반트호프 식으로 구한 압력과 단일상으로 구한 압력 중 큰 값을 사용한다.

방법 2: 반트호프 식과 단일상을 비교하는 방법

- $LmNi_{4.9}Sn_{0.15}$의 PCT 다이어그램을 그리기

1. 전환점 찾기
반트호프 식과 단일상의 식에서 농도와 압력값이 같은 지역을 전환점으로 삼는다.

2. 알파상에서
알파상은 알파-평형 전환점 이전의 상태이며 압력값을 사용할 때에는 반트호프 식으로 구한 압력과 단일상으로 구한 압력 중 작은값을 사용한다.

3. 평형상태에서
평형상태는 알파-평형 전환점 이상 평형-베타 전환점 이하 구간이며 이 때에는 반트호프 식만으로 압력을 구한다.

4. 베타상태에서
베타상은 평형-베타 전환점 이후의 상태이며 압력값을 사용할 때에는 반트호프 식으로 구한 압력과 단일상으로 구한 압력 중 큰 값을 사용한다.

방법 1 VS 방법 2

방법 1 전환 돔 곡선 vs 단일 상 농도 식	방법 2 반트호프 식 vs 단일 상 농도 식
식이 복잡하지 않아서 변형 없이 코드를 짤 수 있다.	반트호프 식을 변형시키기 어려워 근사값을 이용한 방식을 사용해야한다.
특정 온도에서만 값을 얻을 수 있어 전체적인 개형을 보기가 힘들다.	모든 온도에서 값을 얻을 수 있다.

- 저는 방법 2번을 사용하여 $LmNi_{4.9}Sn_{0.15}$의 흡수과정에서의 PCT 다이어그램을 온도에 따라 그리는 파이썬 코드를 만들 것이며 실험을 통해 얻을 수 있는 값은 Y. T. Ge의 논문인 Characterisation of pressure-concentration-temperature profiles for metal hydride hydrogen storage alloys with model development에서 참고 하였습니다.

참고한 값

MH alloy	Process	$\triangle H$ J/(mol.H2)	$\triangle S$ J/(mol.K)	fs -	C_{mid} wt%	T_{crit} °C	P_{crit} bar
$LmNi_{4.91}Sn_{0.15}$	Absorption	26 956.183	102.754	0.575	0.712	129.493	74.193
	Desorption	30 505.142	111.568	0.669	0.712		
$Ti_{0.99}Zr_{0.01}V_{0.43}Fe_{0.09}Cr_{0.05}Mn_{1.5}$	Absorption	19 427.157	95.021	0.442	0.963	84.280	133.165
	Desorption	27 189.307	116.738	0.256	0.963		
$LaNi_{4.25}Al_{0.75}$	Absorption	38 466.374	99.418	0.963	0.511	220.601	13.295
	Desorption	41 423.871	105.408	0.940	0.511		
$Zr_{0.9}Ti_{0.1}Cr_{0.6}Fe_{1.4}$	Absorption	24 822.081	93.360	0.216	0.635	189.604	118.800
	Desorption	29 006.603	102.402	0.251	0.635		

MH alloy	Process	Region	A	γ	V	H
$LmNi_{4.91}Sn_{0.15}$	Absorption	α	5.229E−04	1.103	−1.046E−02	−11 573.73
		β	3.572E−02	0.546	−1.220E−05	−13 619.08
	Desorption	α	6.370E−04	1.594	−7.409E−02	−7383.45
		β	9.249E−02	0.410	3.699E−02	−13 433.95
$Ti_{0.99}Zr_{0.01}V_{0.43}Fe_{0.09}Cr_{0.05}Mn_{1.5}$	Absorption	α	3.041E−05	2.170	−7.370E−03	−6907.53
		β	1.432E−03	1.181	−6.150E−03	−9875.12
	Desorption	α	5.002E−04	1.203	−7.410E−02	−9762.77
		β	3.238E−01	0.217	3.760E−02	−14 080.33
$LaNi_{4.25}Al_{0.75}$	Absorption	α	4.820E−04	1.304	−1.046E−02	−13 917.41
		β	2.759E−01	0.221	−1.220E−05	−16 606.48
	Desorption	α	5.245E−04	1.362	−7.409E−02	−13 537.63
		β	2.918E−01	0.208	3.764E−02	−17 134.42
$Zr_{0.9}Ti_{0.1}Cr_{0.6}Fe_{1.4}$	Absorption	α	5.082E−04	1.255	−1.046E−02	−9606.81
		β	3.504E−01	0.248	−1.220E−05	−10 192.29
	Desorption	α	5.811E−04	1.420	−7.409E−02	−8463.50
		β	2.319E−01	0.232	3.762E−02	−16 065.83

Y. T. Ge. (2023). Characterisation of pressure-concentration-temperature profiles for metal hydride hydrogen storage alloys with model development. 10.

```python
import numpy as np
import matplotlib.pyplot as plt
```

- Numpy: 과학 계산과 수치 연산을 위한 핵심 라이브러리
- Matplotlib.pyplot: 파이썬에서 가장 널리 사용되는 그래프 그리기 라이브러리

```python
def draw_pct_diagram_for_input_temp(temperature_list=None):
```

- PCT 다이어그램 그리는 함수 형성
- temperature_list=None 아무것도 입력되지 않은 값부터 시작 가능

```python
Delta_H = 26956.183
Delta_S = 102.754
f_s = 0.575

A_alpha = 5.229E-4
gamma_alpha = 1.103
V_alpha = -1.05E-2
H_alpha = -11573.73

A_beta = 3.572E-2
gamma_beta = 0.546
V_beta = -1.22E-5
H_beta = -13619.08

R = 8.314
```

- LmNi4.91Sn0.15의 흡수 과정에서 나오는 값들을 미리 넣어 놓는다.

```
C_range_min, C_range_max, C_step = 0, 2.2, 0.005
P_min, P_max = 0.01, 1000

C_range = np.arange(C_range_min, C_range_max + C_step, C_step)
P guess range = np.linspace(P min, P max, 5000)
```

- np.arrange(시작, 끝, 간격) 일정 간격으로 배열을 만들어 준다.
- C_range_max에 Cstep을 더해준 이유는 2.2가 되어버리면 2.195값까지 밖에 나오지 않기 때문

- np.linspace(시작, 끝, 나눌개수) 시작 값과 끝 값까지를 5000개로 나눈 배열 형성

-> 나중에 각 식에서 값은 압력 값이 나올 때가 있는데 그 부분을 찾기 위해 최대한 많은 개수로 나눠줌

```
fig, ax = plt.subplots(figsize=(10,7))
colors = ["b", "g", "r", "c", "m", "y", "k"]
```

- 10 x 7의 공간 형성
- 색은 총 7가지 색으로 파, 초, 빨, 청록, 자홍, 노랑, 검정색을 의미 -> 나중에 각 그래프의 선 색을 의미

```
if temperature_list is None:
    print("온도를 입력하세요(°C). 입력 없이 바로 엔터 누르면 종료합니다.")
```

- temperature_list는 처음에 선언한 비어있을 때도 사용가능하다고 한 것

```
temps_to_plot = []
```

- 온도를 넣을 공간 만들기

```
    while True:
        temp_input = input("온도 (°C): ")
        if temp_input.strip() == '':
            break
        try:
            temp_val = float(temp_input)
            temps_to_plot.append(temp_val)
            if len(temps_to_plot) >= 7:
                print("7개까지 입력 가능합니다.")
                break
        except:
            print("숫자를 입력해 주세요.")

else:
    temps_to_plot = temperature_list
```

- temperature_list가 비어있으면 온도 입력 메시지와 입력한 온도를 저장할 공간을 만든다.
- 그리고 온도를 입력하는 부분을 만든다.
- 만약 빈칸을 입력하게 되면 그대로 멈추어 그 이전 값으로 계산하도록 한다.
- 입력된 온도값은 상수로 바꾸며 temps_to_plot으로 추가한다.
- 만약 7개 이상을 입력시 “7개까지 입력 가능합니다.”라는 문자와 함께 끝낸다.
- 그리고 숫자를 입력하지 않으면 “숫자를 입력해 주세요.”라는 메시지가 나오게 한다.
- temperature_list가 비어있지 않으면 즉 사전에 값이 주어지면 기존 리스트를 그대로 사용할 수 있도록 온도 리스트 전체를 temps_to-plot에 복사한다.

- while true -> 조건이 참인 동안 반복
- if ~ else -> 조건문 작성
- .strip -> 앞 뒤 공백을 제거 해주는 코드
- try ~ except ->코드 실행하고 try 안 쪽 코드의 조건을 만족하면 정상진행 그렇지 않으면 except 안에 코드 실행
- .append -> 값을 추가해주는 코드

```
for idx, T_C in enumerate(temps_to_plot):
    T = T_C + 273.15
    P_ref = 1.0
```

- 온도를 K로 바꿔주는 함수와 기준 압력은 1로 유지
- emumerate: 파이썬에서 리스트를 반복할 때 각 원소의 값과 인덱스 번호를 동시에 뽑아주는 내장함수

```
def P_plateau(C):
    return P_ref * np.exp((-Delta_H/(R*T)) + (Delta_S/R) + f_s * (C))

def C_single_phase(P, A, gamma, V, H):
    return A * (P ** (gamma/2)) * np.exp(-gamma * V * P / (R * T)) * np.exp(-gamma * H / (R * T))
```

- 평형상태의 반트호프 식과 단일상에서의 농도를 구하는 식을 정의함

```
def find_crossing_C(A, gamma, V, H):
    min_diff = float('inf')
    crossing_C = None
    for C in C_range:
        Peq = P_plateau(C)
        diff = np.abs(C_single_phase(P_guess_range, A, gamma, V, H) - C)
        P_phase = P_guess_range[np.argmin(diff)]
        diff_P = np.abs(P_phase - Peq)
        diff_C = np.abs(C_single_phase(P_phase, A, gamma, V, H) - C)
        total_diff = diff_P + diff_C
        if total_diff < min_diff:
            min_diff = total_diff
            crossing_C = C
    return crossing_C
```

- C값을 이전에 0부터 2.2까지 0.005단위로 정했었고 P값은 0부터 1000까지 5000개 단위로 나눴었음 이 5000개의 P값을 단일상의 식에 대입하여 농도를 구하고 0.005단위의 농도와 비교하여 (5000개의 압력마다 구한 농도와 0.005단위의 농도의 차이를 구하고)가장 농도차이가 작은 값에서의 압력 값을 가져와 단일상 압력을 구함
- 이제 구한 값들로 평형상태와의 차이를 각각 구해주고 (단일상의 압력인 P_phase와 평형 압력의 차이와 단일상 식에서 단일상 압력 P_phase를 넣은 농도와 평형상에서의 농도의 차이)
- 농도의 차이와 압력의 차이의 합이 최솟값인 것을 찾아 전환점에서의 농도를 구함
- ->이렇게 근사값을 사용하는 이유는 단일 상의 식을 P로 바꾸기도 힘들고 평형상태의 식을 C에관한 식으로 바꾸기도 힘들기 때문
- min_diff = float('inf'): 무한대부터 아래로 내려오는 형식으로 비교해서 최종에는 0에 가까운 수가 나오도록 하는 것
- for: 값을 만족할때까지 반복함
- np.argmin: 최솟값을 반환
- np.abs: 절댓값 반환

```
C_alpha_eq = find_crossing_C(A_alpha, gamma_alpha, V_alpha, H_alpha)
C_beta_eq = find_crossing_C(A_beta, gamma_beta, V_beta, H_beta)
```

- 알파상과 베타상의 각 전환점의 농도 구함

```
P_final = []

for C in C_range:
    P_eq = P_plateau(C)

    diff_alpha = np.abs(C_single_phase(P_guess_range, A_alpha, gamma_alpha, V_alpha, H_alpha) - C)
    P_alpha = P_guess_range[np.argmin(diff_alpha)]

    diff_beta = np.abs(C_single_phase(P_guess_range, A_beta, gamma_beta, V_beta, H_beta) - C)
    P_beta = P_guess_range[np.argmin(diff_beta)]
```

- 알파와 베타값의 각각의 농도 차이의 최솟값에 따른 단일상의 압력을 구해줌
-> 식을 변형하여 압력을 구하긴 힘들기 때문

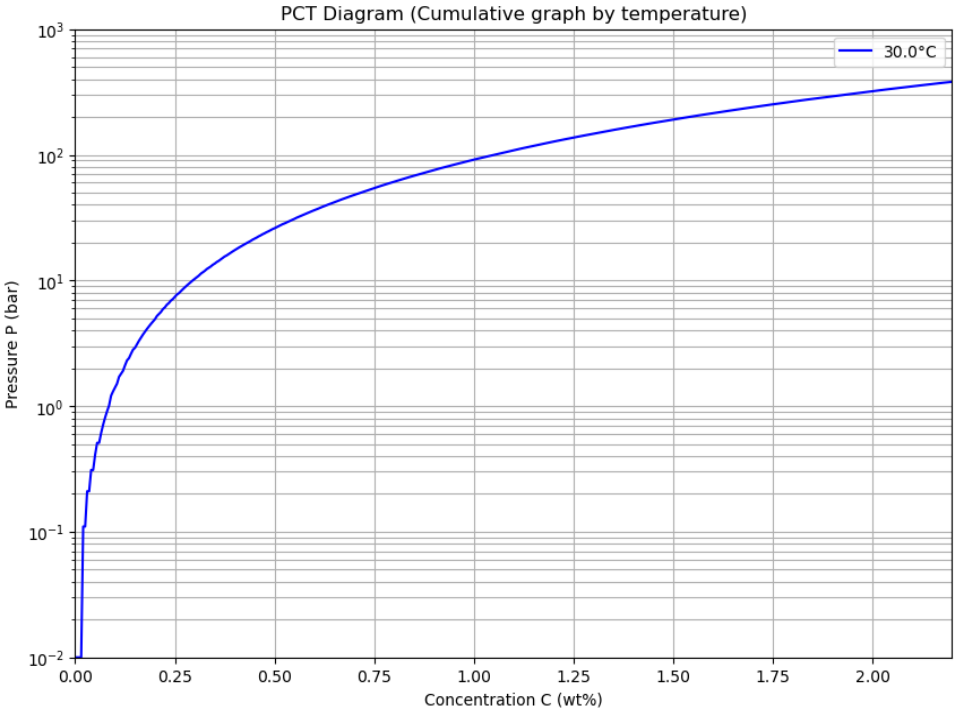
```
if C < C_alpha_eq:
    P_final.append(min(P_alpha, P_eq))
elif C_alpha_eq <= C <= C_beta_eq:
    P_final.append(P_eq)
else:
    P_final.append(max(P_beta, P_eq))
```

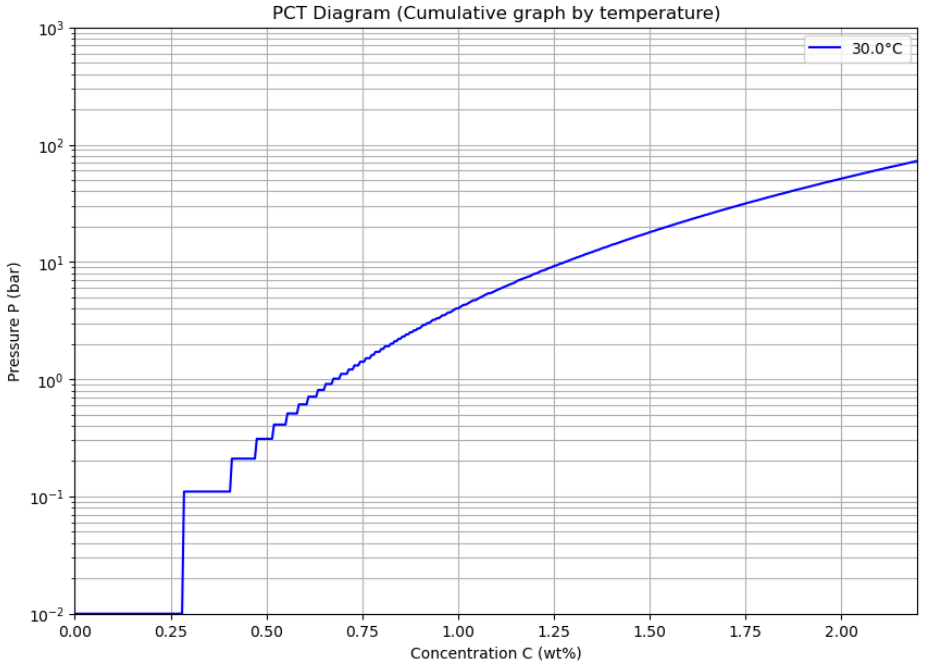
- 전환점보다 농도가 작으면 알파상으로 단일상 압력과 평형 식 압력 중 더 작은 식을 입력 평형 상태에서는 평형압력 식을 사용하고 평형-베타 전환점보다 농도가 크면 베타상으로 두 식 중 더 큰 식의 압력을 사용한다.

```
    color = colors[idx % len(colors)]
    ax.semilogy(C_range, P_final, label=f'{T_C}°C', color=color)

ax.set_xlim([C_range_min, C_range_max])
ax.set_ylim([P_min, P_max])
ax.set_xlabel('Concentration C (wt%)')
ax.set_ylabel('Pressure P (bar)')
ax.set_title('PCT Diagram (온도별 누적 그래프)')
ax.legend()
ax.grid(True, which="both")
plt.show()


draw_pct_diagram_for_input_temp()
```

- 그래프를 그리는 함수

- color = colors[idx % len(colors)]: 여러 온도(곡선)을 그릴 때, 색상을 반복적으로 자동 지정
- ax.semiology: 반로그 그래프로 각 온도별 곡선을 그림
- x축: C_range, y축: P_final
- 각 온도별로 색깔과 범례(label) 다르게 표시
- ax.set_xlim: x축 범위를 정함
- ax.set_ylim: y축 범위를 정함
- ax.set_xlabel: x축 단위 설명 즉 x축 이름 설정
- ax.set_ylabel: y축 단위 설명 즉 y축 이름 설정
- ax.set_title: 그래프 제목 지정
- ax.legend: 곡선별로 범례가 표기 되게 함
- ax.grid: x축, y축 모두에 그리드를 표시해 값 판독 쉽게 함
- plt.show(): 그래프를 보여주는 명령어






```
P_final = []

for C in C_range:
    P_eq = P_plateau(C)

    diff_alpha = np.abs(C_single_phase(P_guess_range, A_alpha, gamma_alpha, V_alpha, H_alpha) - C)
    P_alpha = P_guess_range[np.argmin(diff_alpha)]

    diff_beta = np.abs(C_single_phase(P_guess_range, A_beta, gamma_beta, V_beta, H_beta) - C)
    P_beta = P_guess_range[np.argmin(diff_beta)]
```

다음 코드의 이해

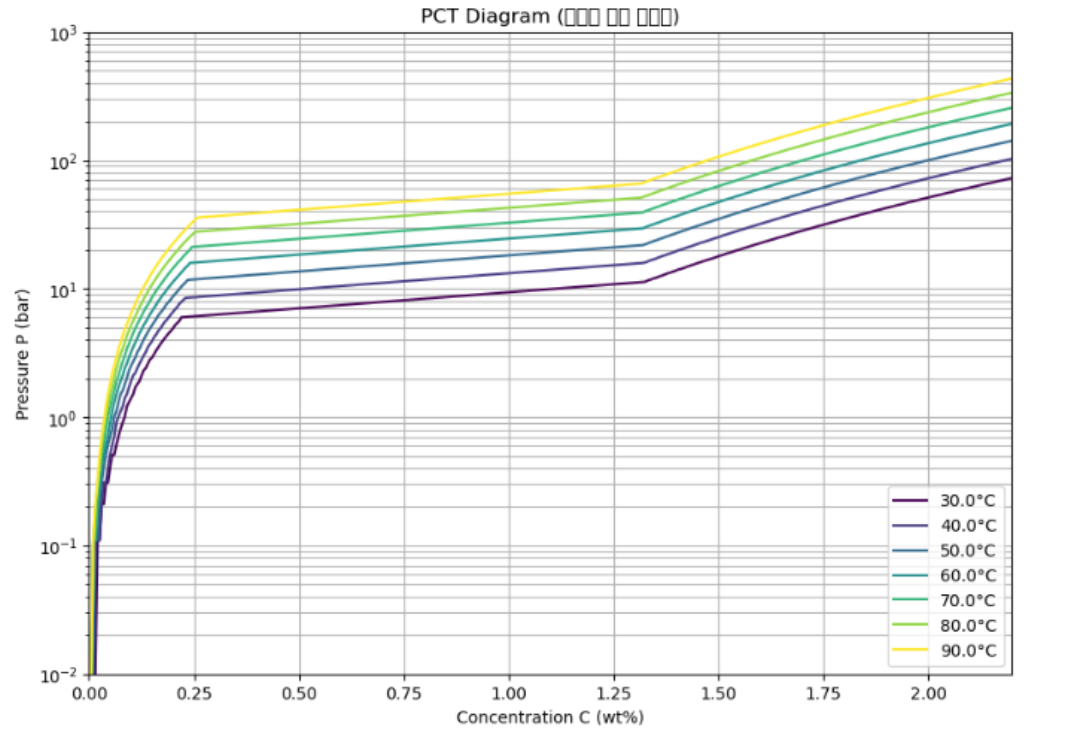
- 단일상 식에서 각 농도에 대한 압력을 구한 그래프: 30도에서
- 위쪽 그래프: 알파상의 상수를 대입하여 얻은 그래프로 초반 부분이 매끄러운 것을 알 수 있다.
- 아래쪽 그래프: 베타상의 상수를 대입하여 얻은 그래프로 베타상은 후반 부분을 다루는 값이 큰 부분이기 때문에 초반 부분이 우그러져 있음을 볼 수 있다.
- 다음과 같은 그래프를 그리는 방법(코드가 진행되는 과정):

1. 기준값으로 삼을 농도가 정해진다. Ex) 0.005wt% (농도 범위 내에서)
2. 단일상 식에 범위가 정해진 압력의 모든 값을 넣는다.
3. 모든 압력값을 넣어 구한 농도와 기준값으로 삼아진 농도를 비교한다(차이로)
4. 농도값의 차이가 가장 작은 부분의 압력을 기준값으로 삼은 농도의 압력으로 삼는다.
5. 이런 방식을 계속해서 그래프를 만든다.

추가하면 좋은 점(많은 온도값에 대하여 컬러맵으로 색깔 칠하기)

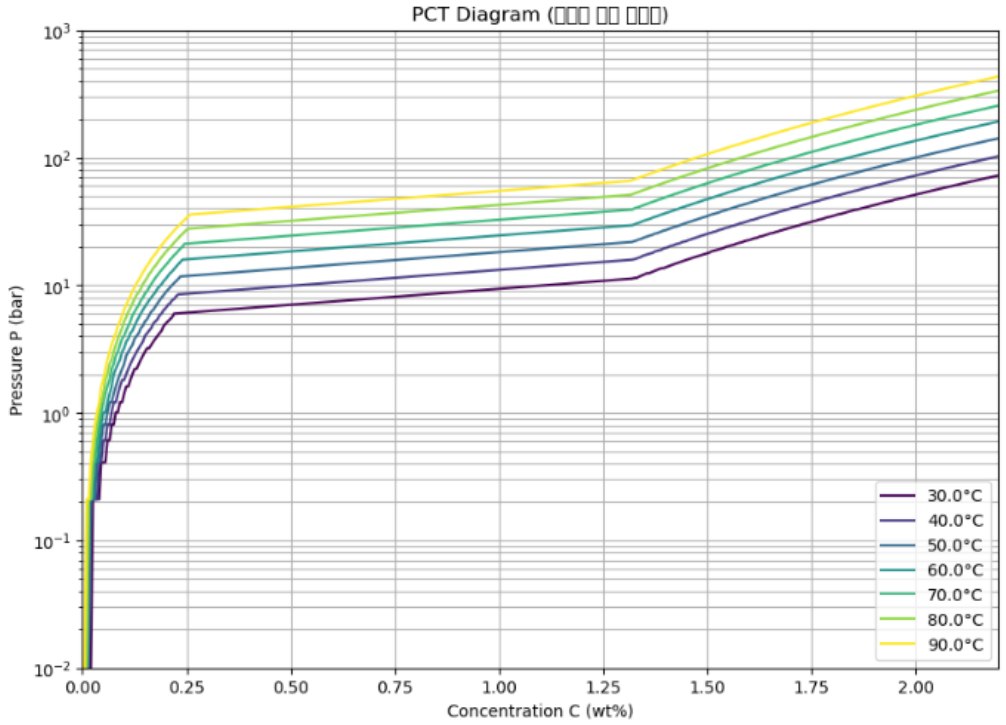
```
from matplotlib import colormaps
from matplotlib.colors import Normalize


cmap = colormaps['viridis']
norm = Normalize(vmin=0, vmax=len(temps_to_plot)-1)


color = cmap(norm(idx))
```

- 입력한 개수에 따라 색깔을 컬러맵의 순서대로 넣어주는 코드로 color를 7개로 설정한 것과 달리 컬러맵의 아주 많은 색깔을 사용할 수 있어 온도 입력 개수를 7개로 제한할 필요가 없다.

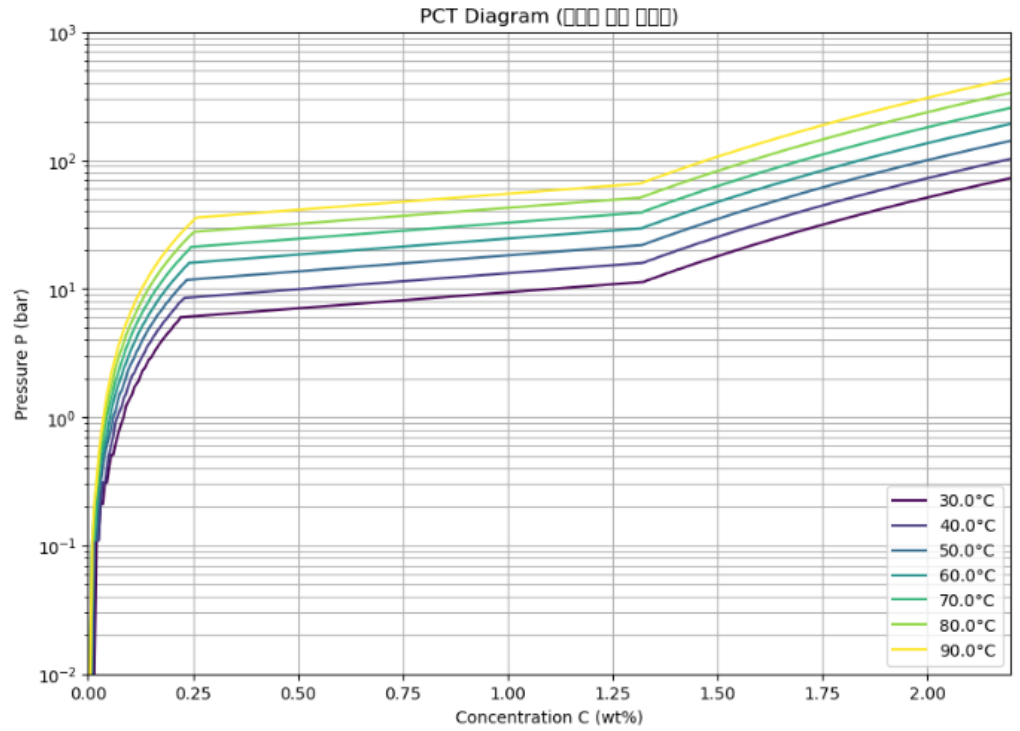


해결한 점

- 완성된 그래프를 얻었을 때 압력값은 5000개 단위로 나눴을 때에는 초반부에 우그러진 그래프의 형태가 보였다. 하지만 10000개의 단위로 바꿔보니 조금 더 정확한 값을 얻을 수 있었기 때문에 우그러짐이 덜하게 되었다. 더 많은 값을 입력하고 싶었지만 너무 많이 나누면 코드 진행이 너무 오래 걸리기 때문에 적당한 10000개로 나눈 값이 괜찮았다.



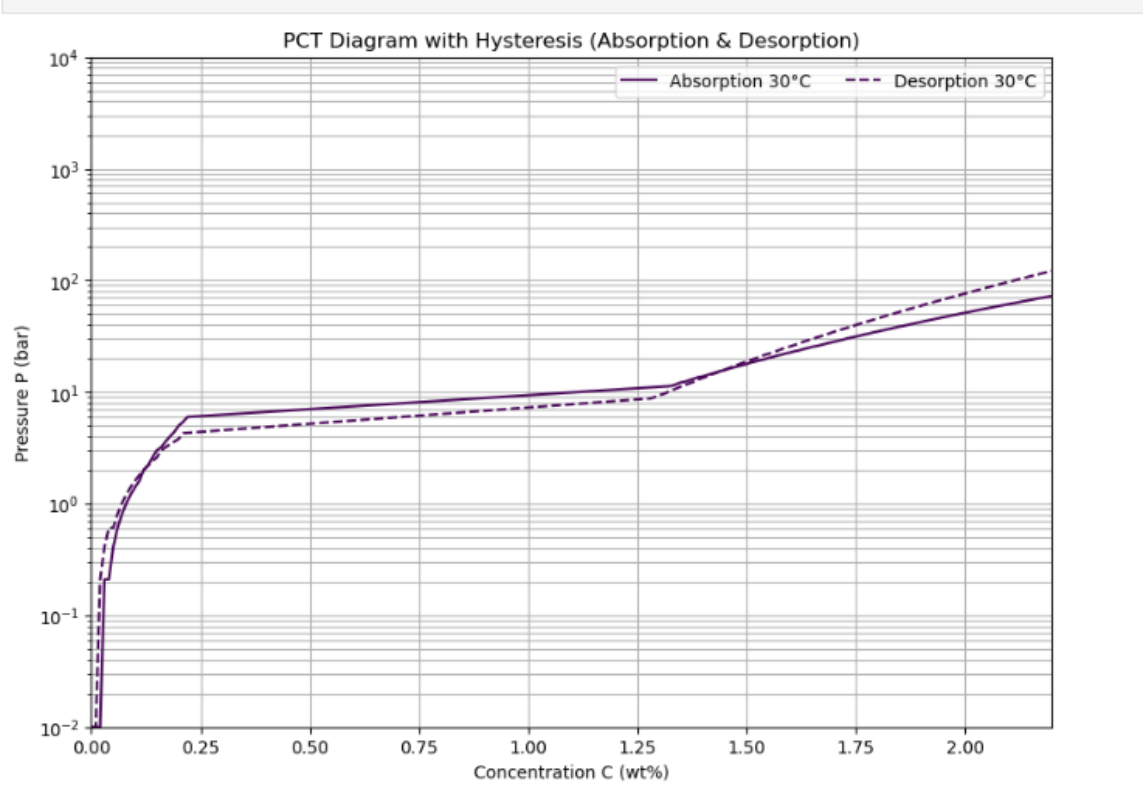
5000개로 압력값을 나눴을 때



10000개로 압력값을 나눴을 때

실패한 점

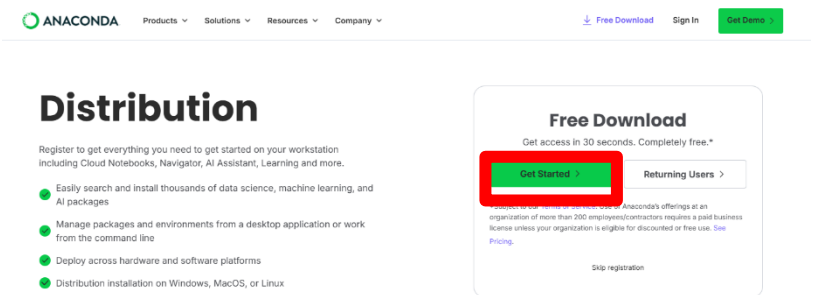
- 위와 같은 방식으로 주어진 값으로 방출 그래프도 같이 그려서 히스테리시스를 확인하고자 하였다. 하지만 히스테리시스의 형태가 얼추 나올 것 같이 만들어졌지만 완전히 정확한 모습이 아니었다. 이는 실험으로 얻은 결과이기 때문에 완벽히 정확한 모습은 아니며 또한 근사값을 이용한 방식이었기 때문에 아무리 쪼개도 정확한 값에 도달을 할 수 없었던 걸로 보인다.(온도를 바꿔봐도 비슷한 현상)

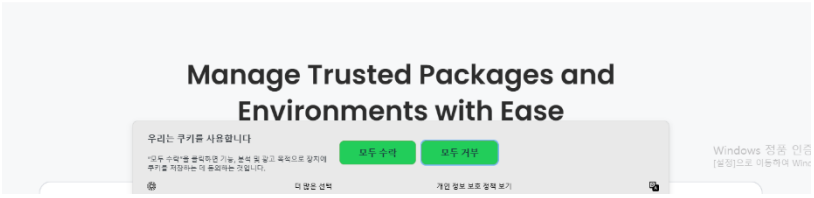


코드 실행(ANACONDA 설치)

1. 구글에 아나콘다 다운로드 입력

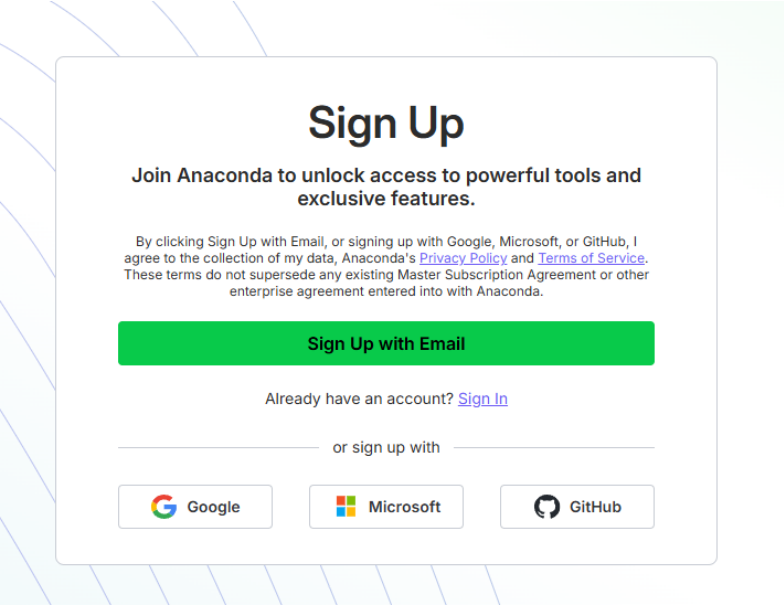
2. 제일 처음에 뜨는 창 터치 후 Get Started입력



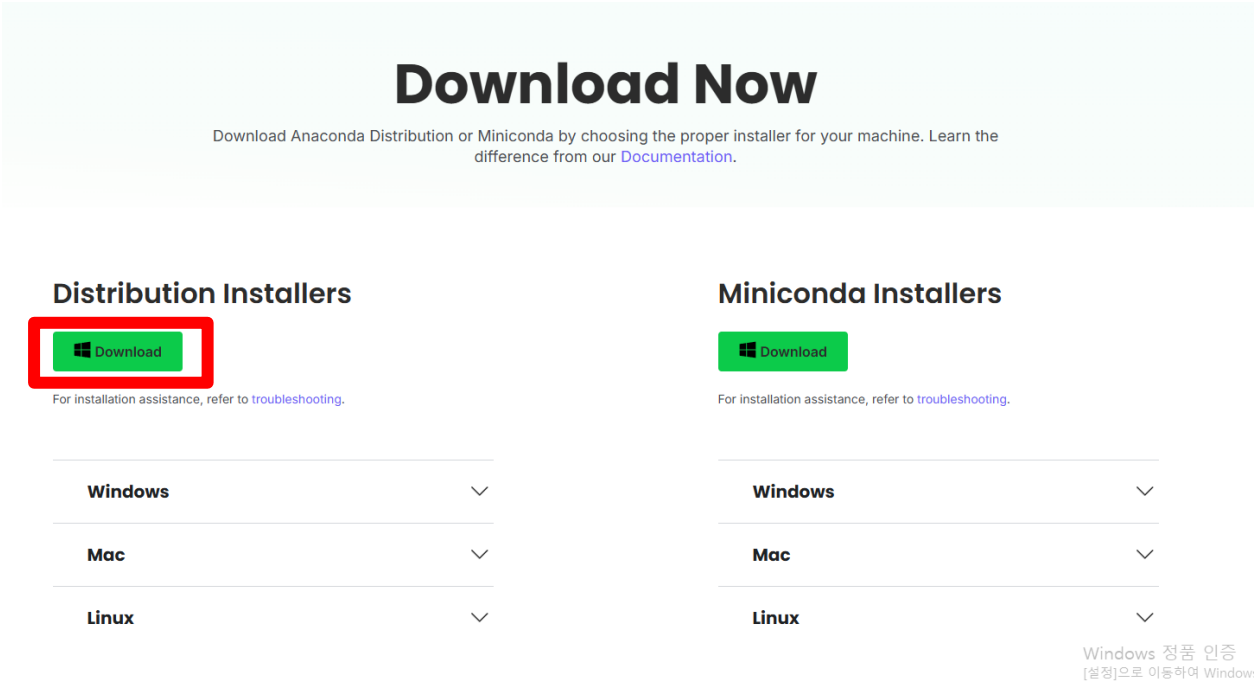


코드 실행(ANACONDA 설치)

3. 이후 로그인 창이 뜨는데 로그인 해주기



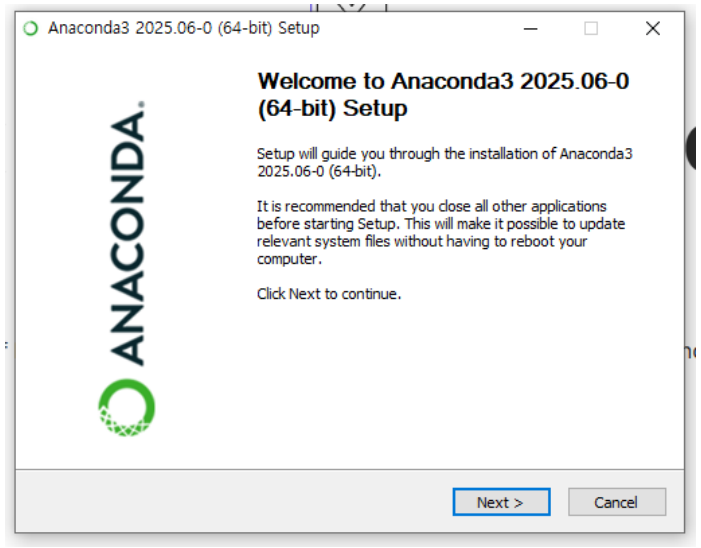
4. Distribution과 Miniconda 중 하나 선택하기 다운로드
프로그램 하나하나 설치하기 귀찮으면 Distribution으로 설치

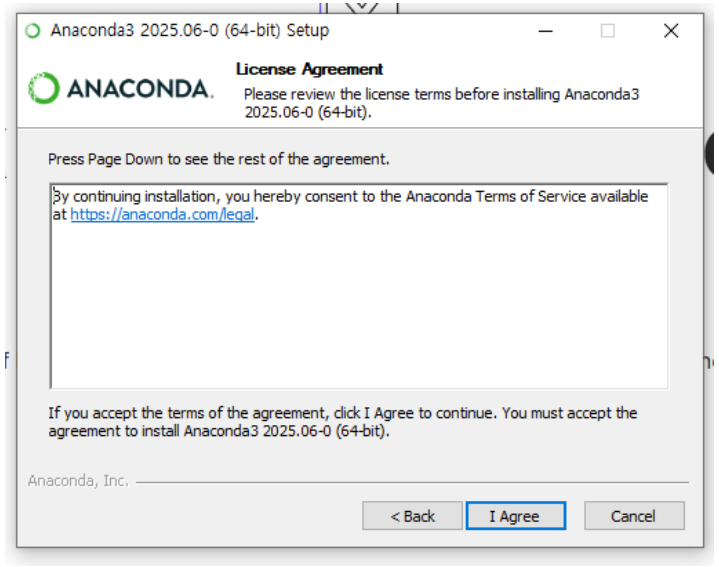


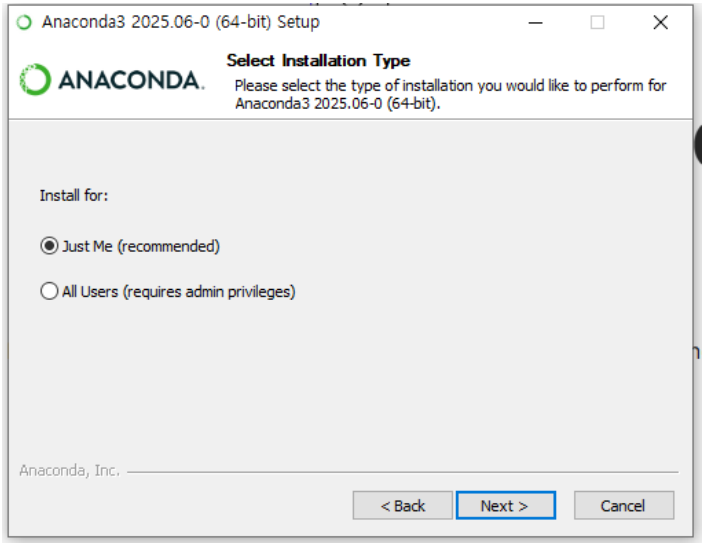
코드 실행(ANACONDA 설치)

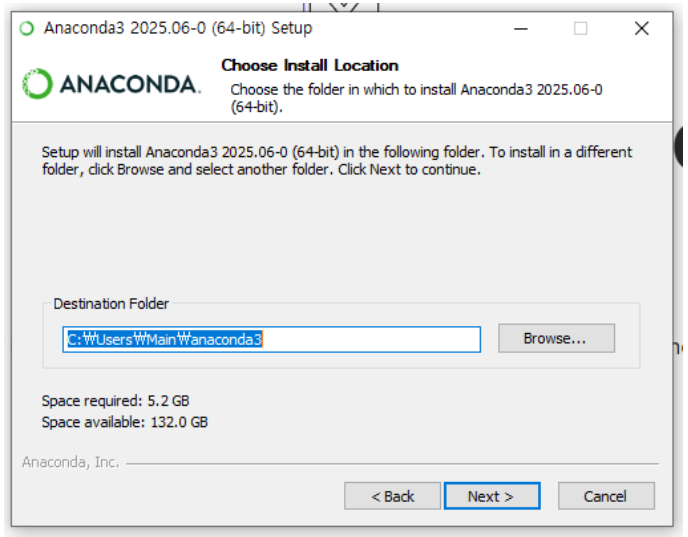
5. 다운로드 누른 후 실행해주면 다음과 같은 창들이 나오는데 모두 next를 누르면 됨

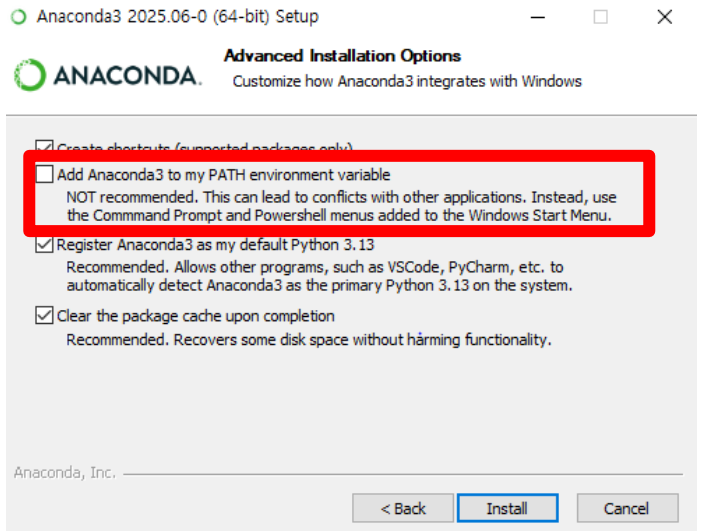
(5번째 사진에서 빨간색으로 표시한 부분은 체크 X 체크하면 다른 프로그램과 충돌 가능성이 생김 나머지는 본인의 기호에 맞게 체크하면 된다.)





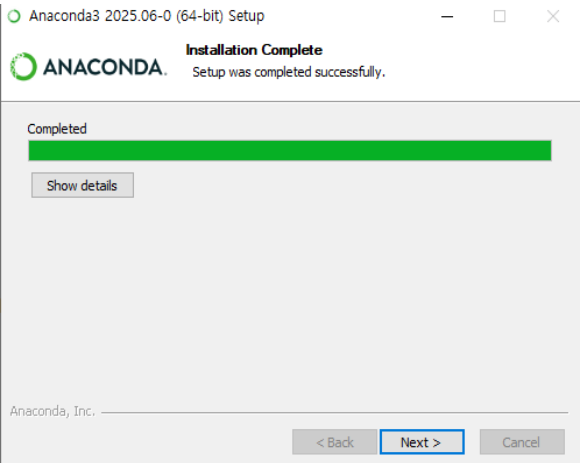


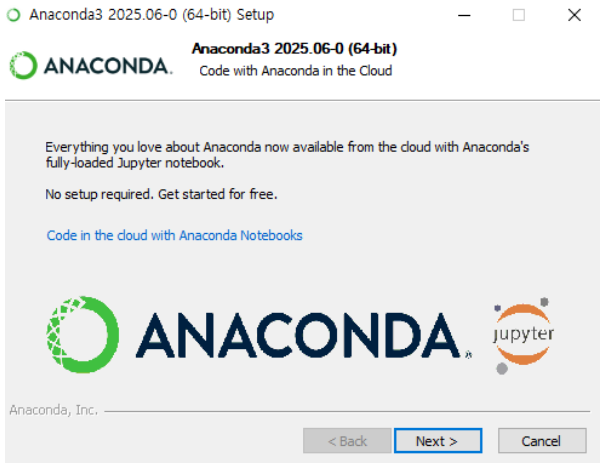


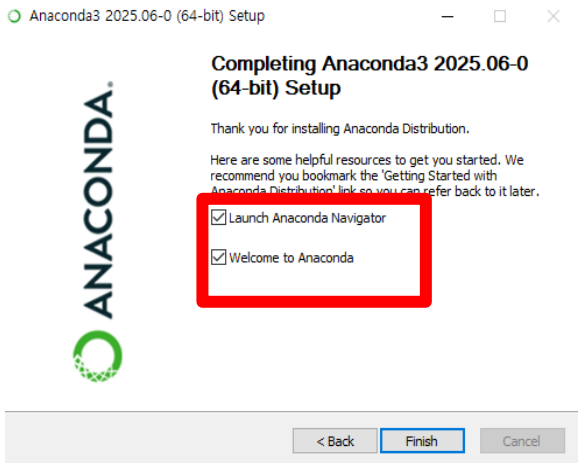


코드 실행(ANACONDA 설치)

6. 설치 완료되면 모두 넘겨주고 바로 Anaconda Navigator실행하려면 체크박스에 체크하면 됨
만약 바로 실행하고 싶지 않다면 체크하지 않고 윈도우 검색 창에 Anaconda Navigator 검색하고 실행

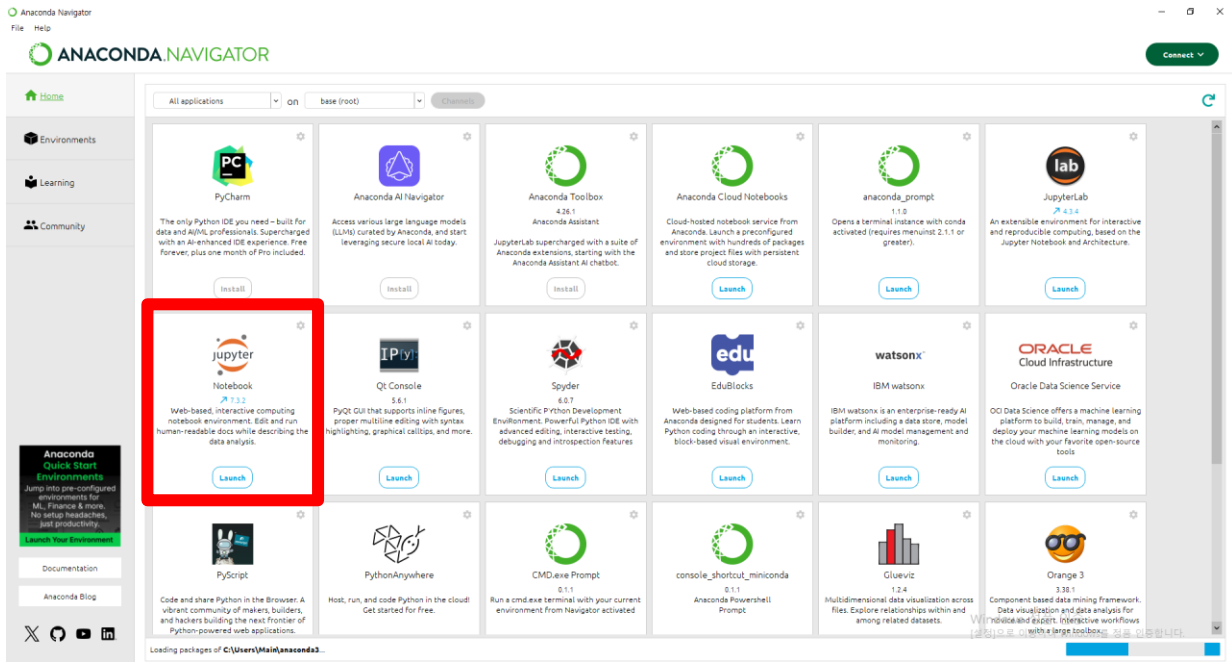






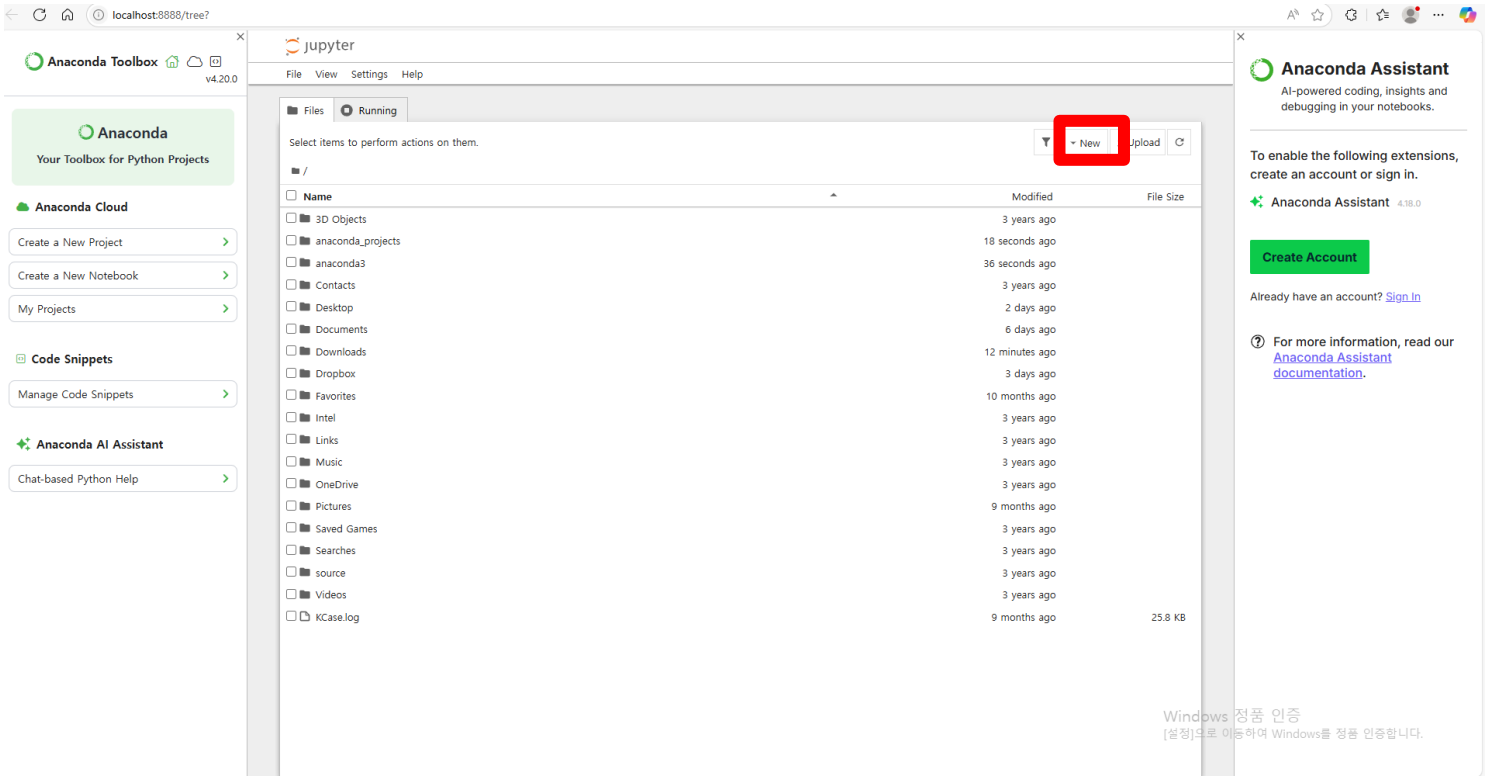
코드 실행(Jupyter Notebook)

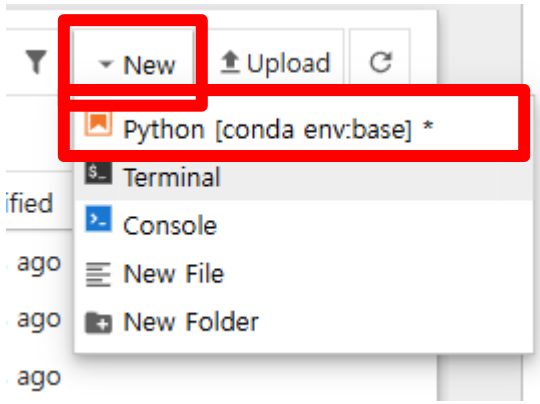
7. Anaconda Navigator에서 Jupyter Notebook 선택



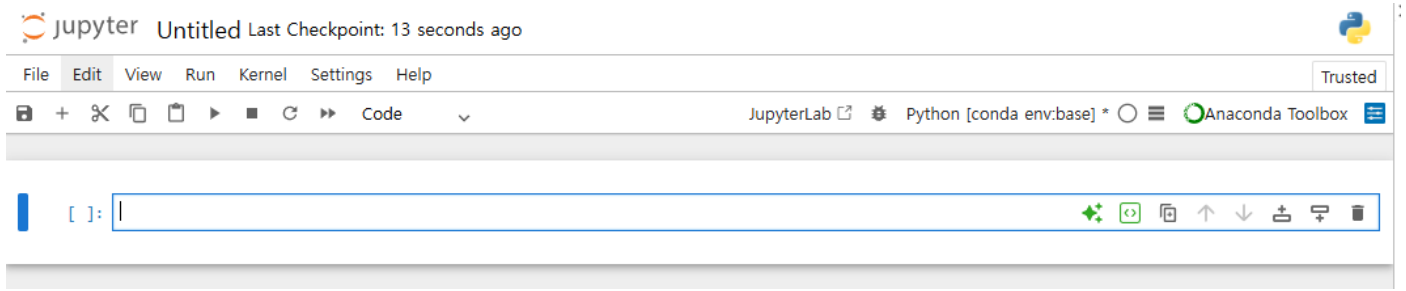
코드 실행(Jupyter Notebook)

8. Jupyter Notebook 실행되면 New 터치 후 Python 선택





코드 실행(주피터 노트북)

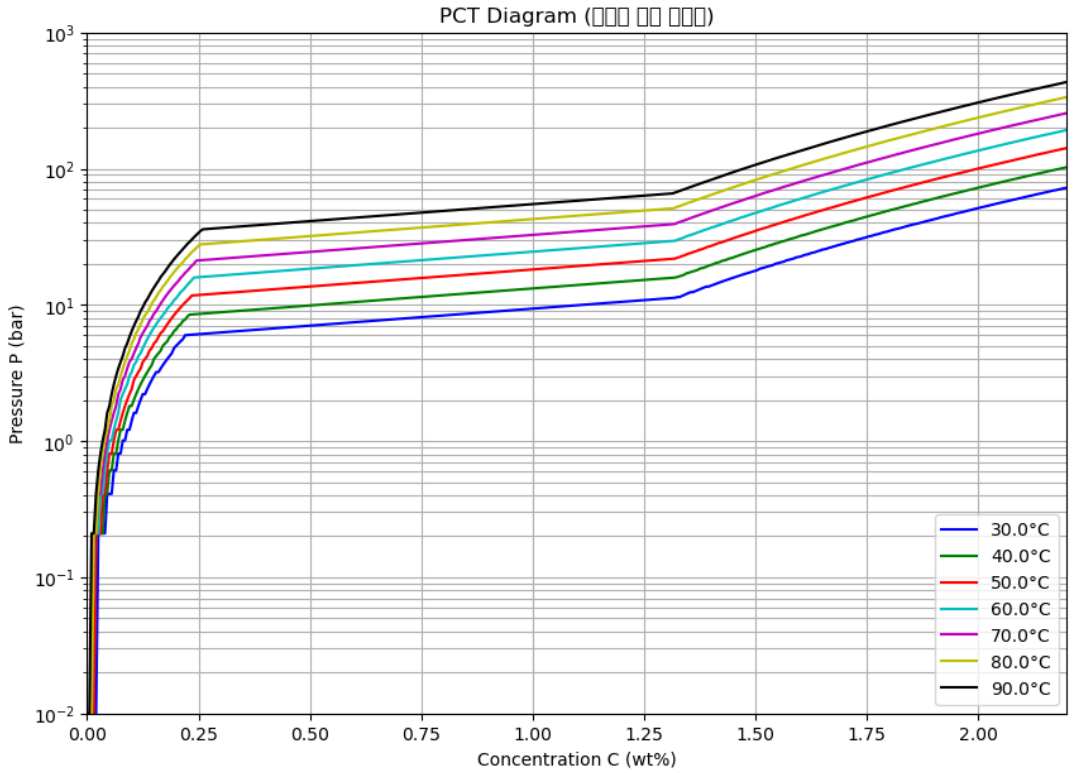



```
            P_eq = P_plateau(C)

            diff_alpha = np.abs(C_single_phase(P_guess_range, A_alpha, gamma_alpha, V_alpha, H_alpha) - C)
            P_alpha = P_guess_range[np.argmin(diff_alpha)]

            diff_beta = np.abs(C_single_phase(P_guess_range, A_beta, gamma_beta, V_beta, H_beta) - C)
            P_beta = P_guess_range[np.argmin(diff_beta)]

            #알파와 베타값의 각각의 농도에 따른 단일상의 압력을 구해줌 -> 식을 변형하여 압력을 구하긴 힘들기 때문

            if C < C_alpha_eq:
                # 알파상 구간: 알파상 압력과 평형 압력 중 작은 것 선택(임계점보다 작을때)
                P_final.append(min(P_alpha, P_eq))
            elif C_alpha_eq <= C <= C_beta_eq:
                # 평형 구간: 평형 압력 사용
                P_final.append(P_eq)
            else:
                # 베타상 구간: 베타상 압력과 평형 압력 중 큰 것 선택(임계점보다 클 때)
                P_final.append(max(P_beta, P_eq))

        color = colors[idx % len(colors)]
        ax.semilogy(C_range, P_final, label=f'{T_C}°C', color=color)

    ax.set_xlim([C_range_min, C_range_max])
    ax.set_ylim([P_min, P_max])
    ax.set_xlabel('Concentration C (wt%)')
    ax.set_ylabel('Pressure P (bar)')
    ax.set_title('PCT Diagram (온도별 누적 그래프)')
    ax.legend()
    ax.grid(True, which="both")
    plt.show()


draw_pct_diagram_for_input_temp()

코드해석
```

9. 다음 창이 뜨면 코드 입력 후
Shift + Enter를 눌러 코드를 실행해준다.

코드 실행(주피터 노트북)

```
온도를 입력하세요(°C). 입력 없이 바로 엔터 누르면 종료합니다.
온도 (°C): ↑↓ for history. Search history with c-↑/c-↓
```



10. 온도 입력창에 온도 입력하고 Enter 누르면 PCT 다이어그램을 얻을 수 있다.