빅데이터혁신 공유 사업 – 경기과학기술대학교 외 7개 대학

COS PRO 파이썬 2급

류현수

숭실대학교 겸임교수
공룡컴 평생교육원 이사

2023. 07. 10 ~ 07. 13

(10:00~13:00 , 4일, 총 12시간)

자격 시험일 : 15일(토)

COS PRO 자격증 소개

- Professional Coding Specialist
- 높은 수준의 프로그래밍 능력 요구
- Python, C, C++, Java 자격증
- CBT(Computer Based Test)로 평가 방식
- 시험 종료 후 바로 결과 확인
- 정기 시험

(Professional Coding Specialist)

http://www.ybmit.com

COS Pro 자격증 종류

Python, C, C++, Java

등급	시간	합격 기준	평가 항목	수준
1급	90분	600점 이상 (1000점 만점)	10개 문항 (완성 3, 부분 7)	개발자 채용 시 출제되는 문제와 유사한 수준
2급	50분		10개 문항 (완성 2, 부분 8)	프로그래밍 언어의 기초적인 개념을 충실히 배운 다음 이를 활용해 간단한 문제를 해결할 수 있는 수준
3급	50분		10개 문항 (완성 5, 부분 5)	프로그래밍 언어의 기초 개념을 이해하는 수준

https://www.ybmit.com/cos_pro/cos_pro_info.jsp

COS PRO 파이썬 2급 샘플 문제

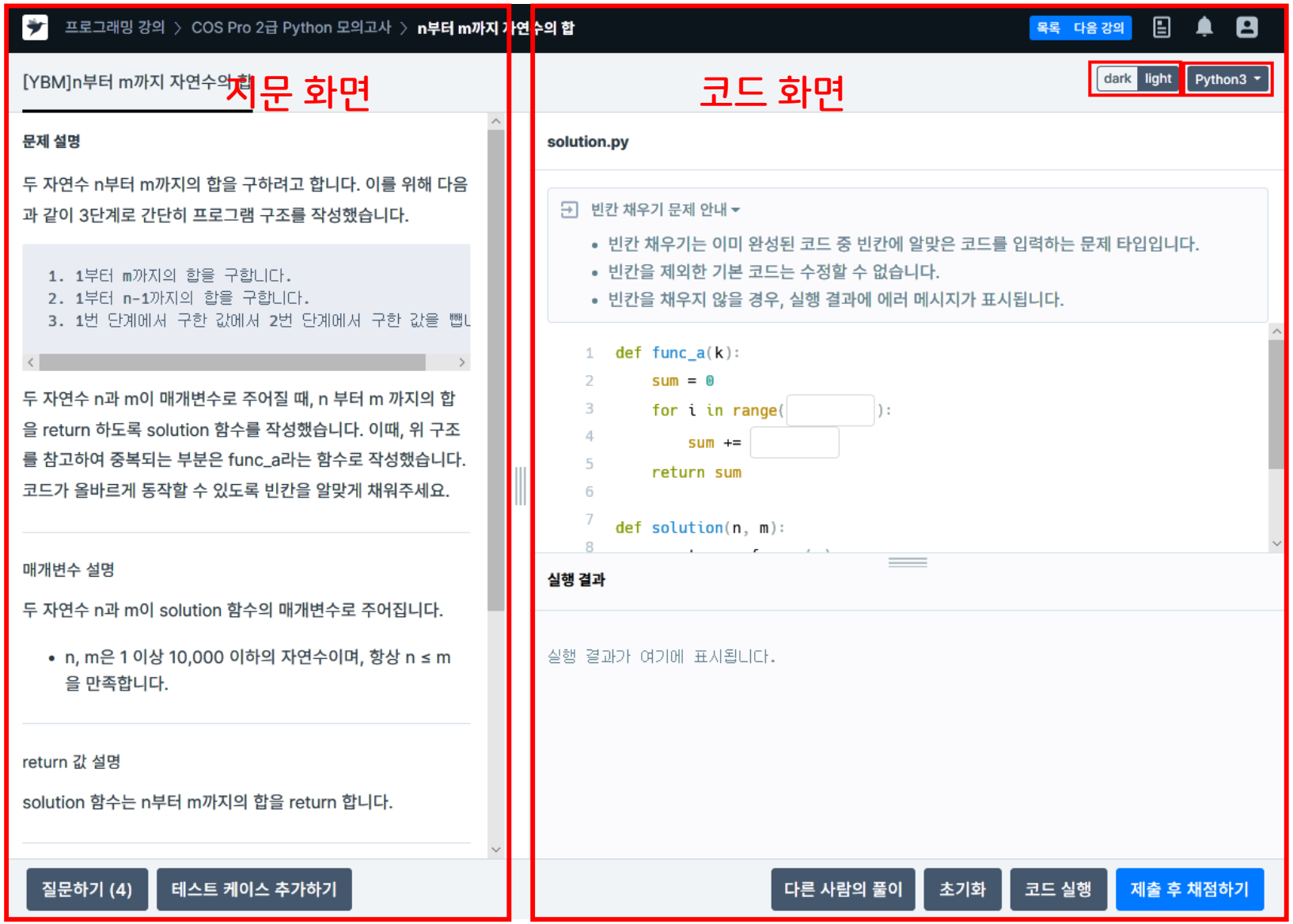
- ybmit 사이트에서 샘플 문제 제공

  https://www.ybmit.com/cos_pro/cos_pro_r_test.jsp

급수	언어	다운로드		
		지문	문제	정답
COS Pro 2급	C	지문	문제파일	정답파일
	C++	지문	문제파일	정답파일
	Java	지문	문제파일	정답파일
	Python	지문	문제파일	정답파일

문제 지문 문제 코드 정답 코드

COS Pro 체험하기 http://www.ybmit.com ➔ COS Pro ➔ COS Pro 체험하기



[YBM]n부터 m까지 자연수의 합

dark light Python3

코드가 올바르게 동작할 수 있도록 빈칸을 알맞게 채워주세요.

매개변수 설명

두 자연수 n과 m이 solution 함수의 매개변수로 주어집니다.

- n, m은 1 이상 10,000 이하의 자연수이며, 항상 n ≤ m 을 만족합니다.

return 값 설명

solution 함수는 n부터 m까지의 합을 return 합니다.

예시

n	m	result
5	10	45
6	6	6

예시 설명

예시 #1

5부터 10까지 자연수의 합은 45입니다.

예시 #2

6부터 6까지 자연수의 합은 6입니다.

solution.py

빈칸 채우기 문제 안내

- 빈칸 채우기는 이미 완성된 코드 중 빈칸에 알맞은 코드를 입력하는 문제 타입입니다.
- 빈칸을 제외한 기본 코드는 수정할 수 없습니다.
- 빈칸을 채우지 않을 경우, 실행 결과에 에러 메시지가 표시됩니다.

```
1  def func_a(k):
2      sum = 0
3      for i in range( k+1 ):
4          sum += i
5      return sum
6
7  def solution(n, m):
8
```

실행 결과

```
입력값 〉 5, 5
기댓값 〉 6
실행 결과 〉 테스트를 통과하였습니다.
```

테스트 결과 (~˘▾˘)~

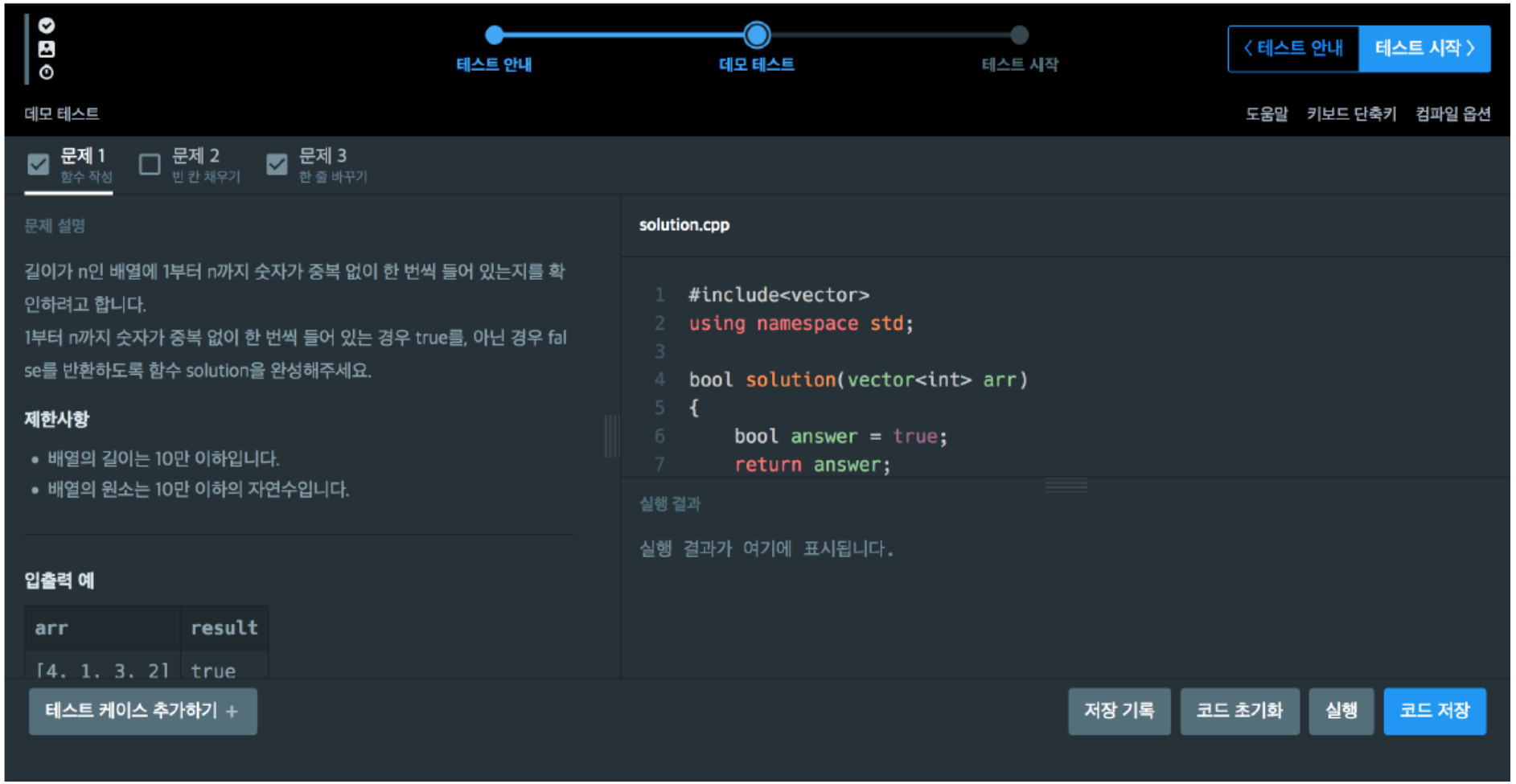
2개 중 2개 성공

샘플 테스트 케이스를 통과했다는 의미로, 작성한 코드가 문제의 정답은 아닐 수 있습니다.
(샘플 테스트 케이스는 [테스트 케이스 추가하기] 버튼을 통해 확인하실 수 있습니다.)

질문하기 (4) | 테스트 케이스 추가하기 | 다른 사람의 풀이 | 초기화 | 코드 실행 | 제출 후 채점하기

COS Pro 실제 시험 화면 구성

https://www.ybmit.com/cos_pro/cos_pro_info.jsp



저장 기록

저장한 코드를 확인하는 기능입니다.

8 / 8 〈 이전 종료 〉

코드 초기화
: 작성한 코드를 지우기

테스트 실행

작성한 코드가 정상 동작하는지 확인하는 기능입니다. print 문을 통해 solution 함수가 적절한 값을 return 하는지 체크합니다.

4 / 8 〈 이전 다음 〉

코드 저장

코드를 저장하는 기능입니다.

5 / 8 〈 이전 다음 〉

※ COS Pro 수검프로그램 예시.pdf 참조

결과 화면

시험이 끝나면 시험 결과를 알 수 있으며, 설문조사에 참여할 수 있습니다.

합/불합격 여부와 취득 점수는 채점이 완료 후 공개됩니다.
채점 완료까지는 약 5분 정도의 시간이 소요될 수 있습니다.

합격 여부와 점수를 공개합니다.

COS PRO 설문조사에 참여할 수 있습니다.

문제 풀이 및 시험 대비 주의사항

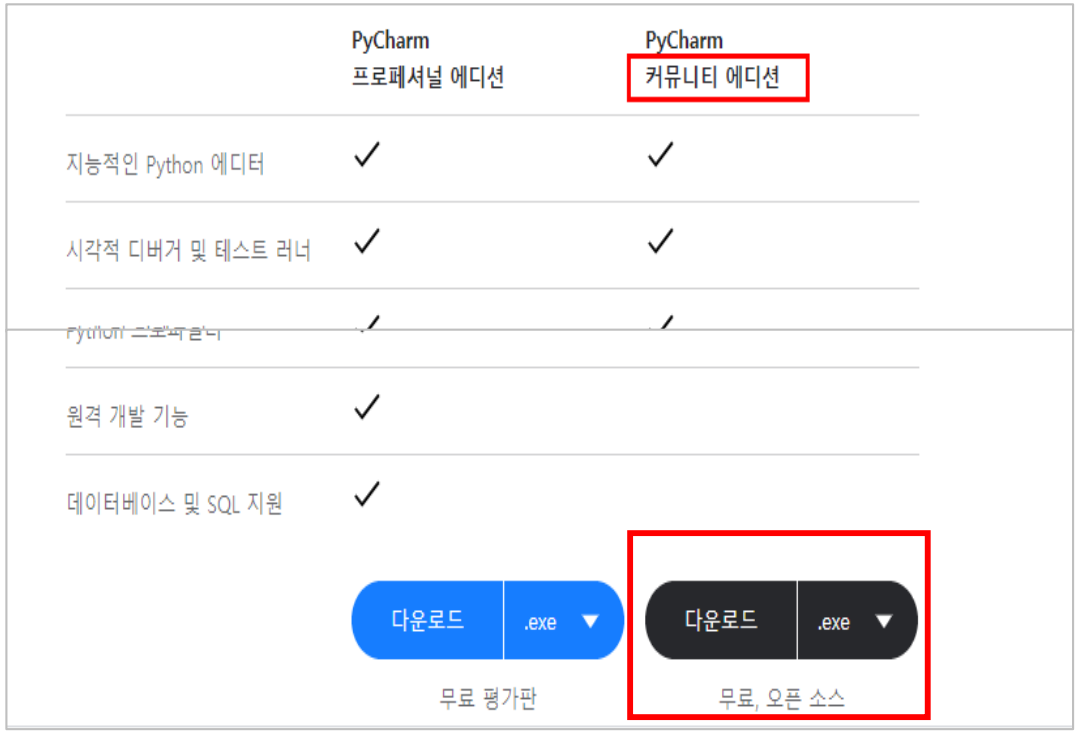
- 인강에서 샘플 문제(pdf) & 문제 코드 & 정답 코드 & 추가 제안 코드 제공
- 시험 보기 전 추가 샘플 문제가 새롭게 올라와 있는지 확인하기(ybmit.com)
- 시험 화면 등 변경 사항이 있는지 확인하기(ybmit.com)
- 주어진 시간을 확인하면서 난이도가 쉽고 전체 코딩이 아닌 문제부터 작성
- 각 번호별 배점이 다를 수 있음
- 신분증 필수 지참(대학교 학생증은 불가) https://www.ybmit.com/test_rec/test_guide.jsp
- 시험 시작 시간보다 일찍 도착

파이참(PyCharm) - 설치하기

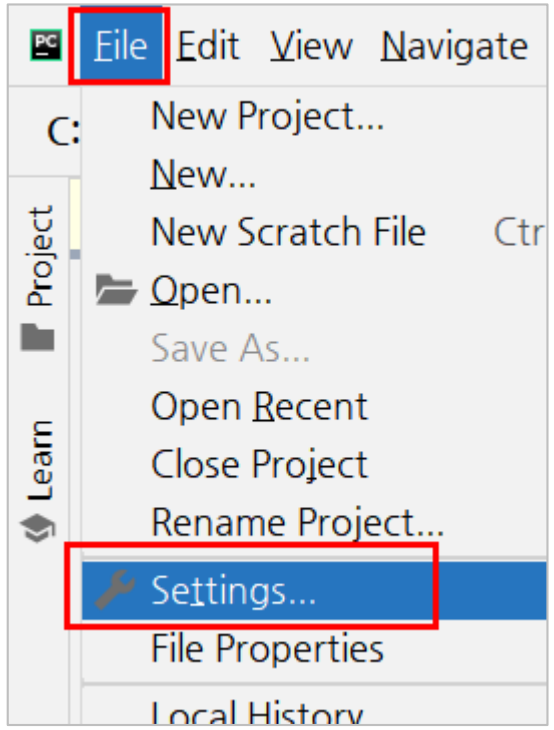
- 검색 엔진에서 “파이참“ 검색

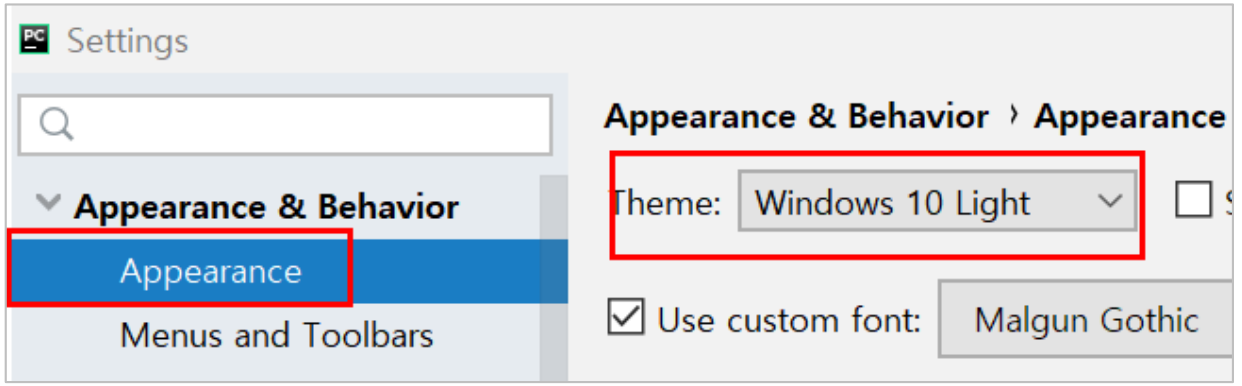
https://www.jetbrains.com/ko-kr/pycharm/

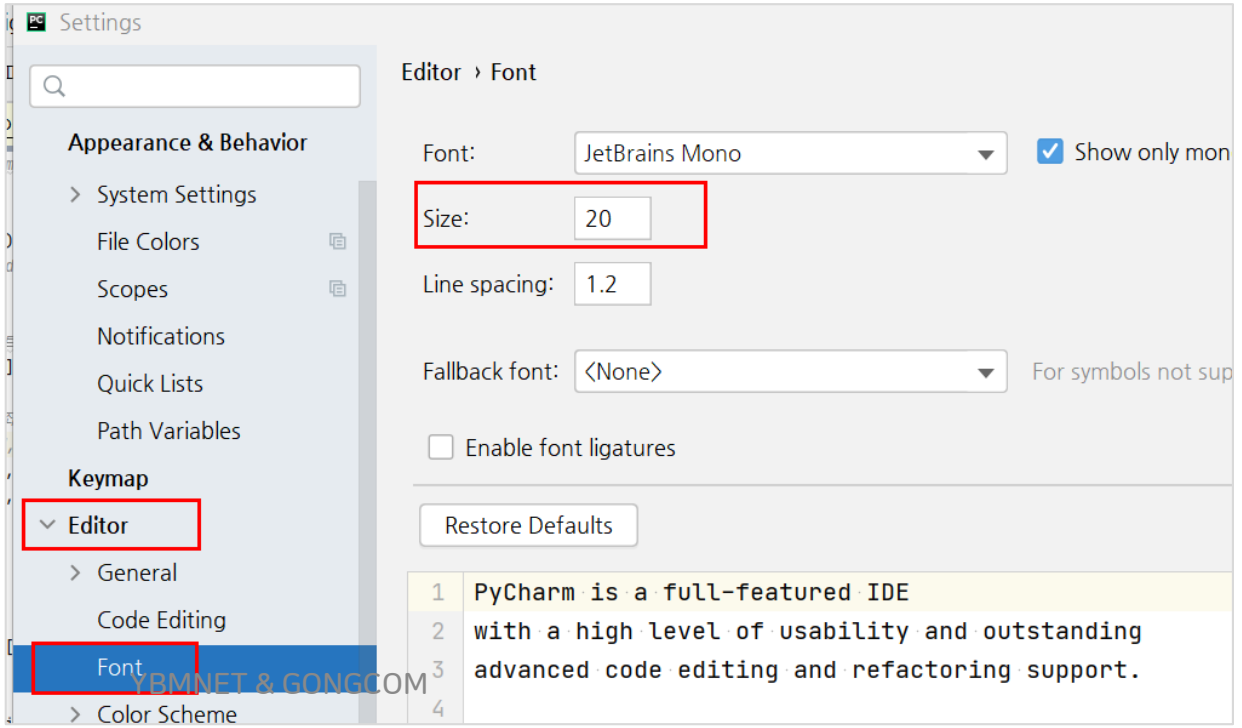
community 버전 설치



파이참(PyCharm) - 파이참 모양, 폰트 변경하기





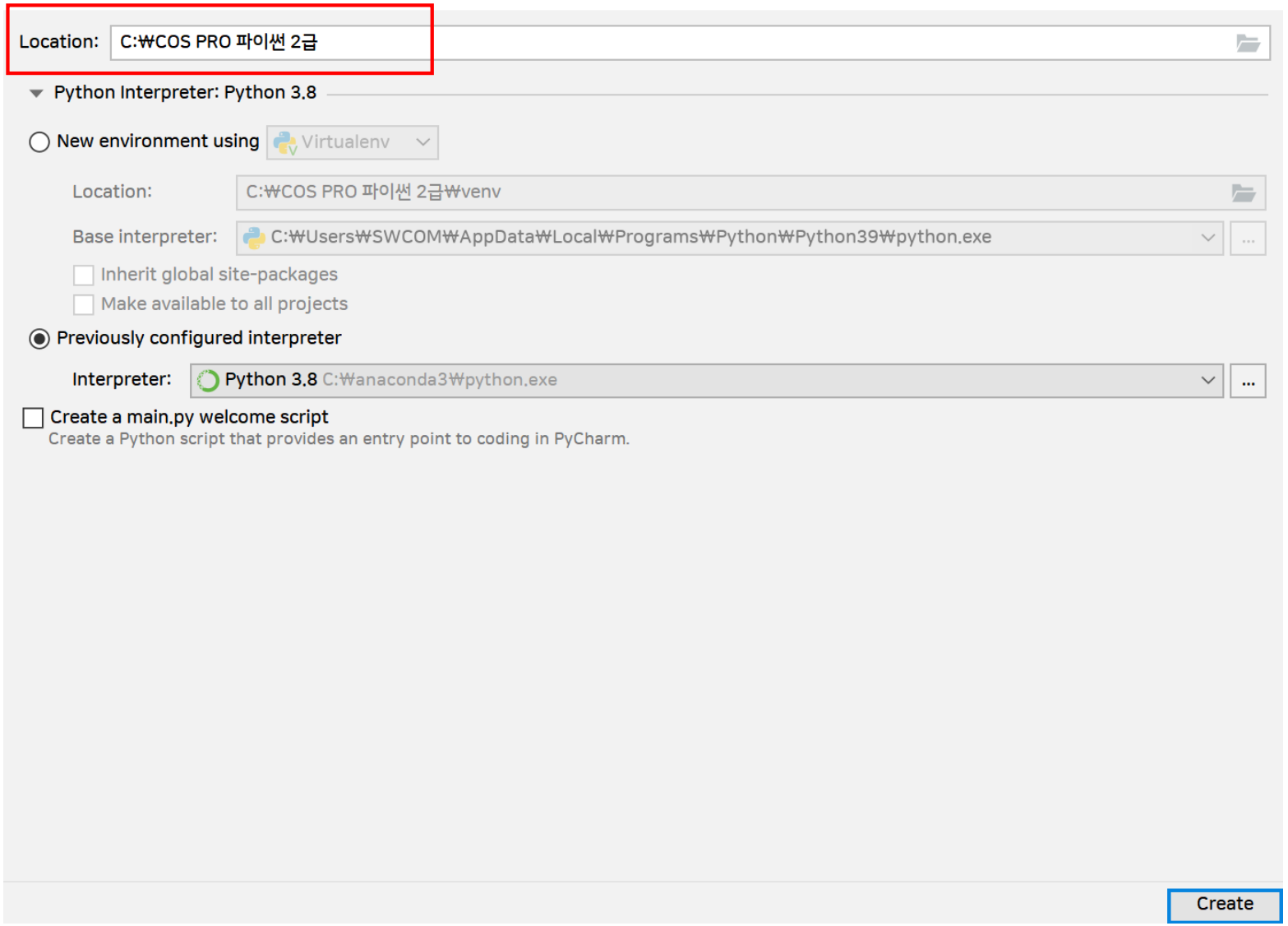


1. ybmit.com의 샘플 문제 다운로드

   ➔ c:에 압축 풀기

2. 파이참에서 [File]-[New Project]

   ➔ 압축 풀은 폴더 선택하기



리스트 항목 값 변경

```
x = [1,2,3,4]
print(x)

x[2] = 5
print(x)

x[0] = 0
print(x)
```

```
[1, 2, 3, 4]
[1, 2, 5, 4]
[0, 2, 5, 4]
```

리스트 요소 출력

a=[1,2,3]

```
for i in range(3):
    print(a[i])
```

```
for i in range(len(a)):
    print(a[i])
```

```
for k in [1,2,3]:
    print(k)
```

```
for k in a:
    print(k)
```

if '키워드' in 리스트

```
a=['orange','apple','banana']

if 'apple' in a:
    print('사과가 있습니다')
```

딕셔너리 만들기

- 키(key): 문자열, 정수형, 실수형, 튜플 등 ~~(리스트)~~
- 값(value): 문자열, 정수형, 실수형, 튜플, 리스트 등 모든 데이터 타입 가능

```
a={}

b = {1:"용이", 2:"코스",3:"파이썬"}

c={"1반":28,"2반":30}
```

- 리스트는 딕셔너리의 키로 사용할 수 없음

딕셔너리 항목 접근과 변경

키(Key)	1반	2반	3반	4반
값(Value)	27	28	27	30

```
m={"1반":27,"2반":28, "3반":27,"4반":30}
print(m["1반"])

for i in m:
   print(i, ":" , m[i])

m["3반"]=50

print(m)
```

리스트 메소드

append(값)	해당 값을 리스트에 추가
insert(인덱스,값)	해당 인덱스 번호에 값을 추가
extend(리스트)	리스트 뒤에 리스트를 붙임

```
a=list()

a.append(10)
print(a)                    [10]

a.insert(1,30)
print(a)                    [10, 30]

a.extend([50,60])
print(a)                    [10, 30, 50, 60]
```

index(값)	값에 해당하는 인덱스 번호 리턴
count(값)	해당 값의 개수를 리턴
pop(인덱스)	해당 인덱스 데이터를 삭제
remove(값)	해당 값을 삭제
clear()	리스트 전체 삭제

```
print(a.index(30))          1

print(a.count(10))          1

a.remove(10)
print(a)                    [30, 50, 60]

a.pop(2)
print(a)                    [30, 50]

a.clear()
print(a)                    []
```

자주 사용하는 함수들

min(리스트)	리스트 항목값 중 가장 작은 값을 리턴
max(리스트)	리스트 항목값 중 가장 큰 값
sum(리스트)	리스트 항목값의 합
len(리스트)	리스트 항목의 개수

```
a=[10,20,30]
print(min(a),max(a),sum(a),len(a))
```

```
10 30 60 3
```

map(함수,리스트)	리스트 항목값에 함수를 적용

리스트 항목값에 일괄적으로 함수를 적용하여 그 결과를 돌려줌
돌려 받는 값은 리스트가 아니므로 리스트로 변환해야 함

```
b=['11','12','13']
b2=map(int,b)
print(b2)
b2=list(b2)
print(b2)
```

```
<map object at 0x0000020B9B3A83D0>
[11, 12, 13]
```

리스트 컴프리헨션 (표현식)

[값 for 변수 in range(반복 범위)]

list(값 for 변수 in range(반복 범위))

```
a=[0 for _ in range(5)]
print(a)

a=[0 for i in range(5)]
print(a)

a=[0]*5
print(a)
```

```
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
```

[식 for 변수 in range(반복 범위)]

list(변수 for 변수 in range(반복 범위))

```
a=[i for i in range(5)]
print(a)
```

```
a=[]
for i in range(5):
    a.append(i)
```

```
a=[str(i+10) for i in range(5)]
print(a)

a=list(i for i in range(3,10,2))
print(a)
```

```
[0, 1, 2, 3, 4]
['10', '11', '12', '13', '14']
[3, 5, 7, 9]
```

딕셔너리 관련 함수, 메소드

dict()	딕셔너리로 리턴

```
m2={"A":27,"B":28,"C":27}
print(m2)

m3= dict(A=2,B=28,C=27)
print(m3)

m4= dict(A반=2,B반=28,C반=27)
print(m4)
```

```
{'A': 27, 'B': 28, 'C': 27}
{'A': 2, 'B': 28, 'C': 27}
{'A반': 2, 'B반': 28, 'C반': 27}
```

https://docs.python.org/3/library/stdtypes.html#dict

keys()	딕셔너리의 키들을 리턴
values()	딕셔너리의 값들을 리턴
items()	딕셔너리의 키와 값의 쌍을 리턴

```
m={"1반":27,"2반":28,"3반":27}

print(m.keys())

print(m.values())

print(m.items())
```

```
dict_keys(['1반', '2반', '3반'])
dict_values([27, 28, 40])
dict_items([('1반', 27), ('2반', 28), ('3반', 40)])
```

```
chg_list1=list(m.values())
print(chg_list1)

chg_list2=list(m.keys())
print(chg_list2)

chg_list3=list(m.items())
print(chg_list3)
```

```
[27, 28, 40]
['1반', '2반', '3반']
[('1반', 27), ('2반', 28), ('3반', 40)]
```

소트 관련 정리

a=[9,3,1,15]

```
# a.sort() : 오름차순. 리스트 순서 자체를 바꿈. 메소드
a.sort()
print("=== a.sort() 후 a ")
print(a,"₩n")
```

```
=== a.sort() 후 a
[1, 3, 9, 15]
```

```
# a.sort(reverse=True) : 내림차순. 리스트 순서 자체를 바꿈. 메소드
a.sort(reverse=True)
print("=== a.sort(reverse=True) 후 a ")
print(a,"₩n")
```

```
=== a.sort(reverse=True) 후 a
[15, 9, 3, 1]
```

```
# a.reverse() : 역순. 리스트 순서 자체를 바꿈. 메소드
a.reverse()
print("=== a.reverse() 후 a ")
print(a,"₩n")
```

```
=== a.reverse() 후 a
[15, 1, 3, 9]
```

```
# sorted(a) : 오름차순. 리스트 자체의 순서를 바꾸지 않음, 함수
print("=== sorted(a)")
print(sorted(a),"₩n")
print("=== sorted(a) 후 a " )
print(a,"₩n")
```

```
=== sorted(a)
[1, 3, 9, 15]

=== sorted(a) 후 a
[9, 3, 1, 15]
```

```
sorted(a, reverse=True)
```

```
# reversed(a) : 역순. 리스트 자체의 순서를 바꾸지 않음, 함수
print("=== reversed(a)")
print(reversed(a))
print(list(reversed(a)),"₩n")

print("=== reversed(a) 후 a " )
print(a,"₩n")
```

```
=== reversed(a)
<list_reverseiterator object at 0x033A4790>
[15, 1, 3, 9]

=== reversed(a) 후 a
[9, 3, 1, 15]
```

enumerate 함수

- (인덱스, 항목값) 튜플 형태로 변환

```
x=['a','b','c']

e=list(enumerate(x))
print(e)
```

```
[(0, 'a'), (1, 'b'), (2, 'c')]
```

```
for i,v in enumerate(x):
    print(i,v)
```

```
0 a
1 b
2 c
```

zip 함수

- 두 개 리스트를 하나로 묶음
- 인덱스가 같은 항목값을 쌍으로 튜플 형태로 반환

```
x1=[1,2,3]
x2=[4,5,6]

z=list(zip(x1,x2))
print(z)
```

```
[(1, 4), (2, 5), (3, 6)]
```

```
for a,b in zip(x1,x2):
    print(a,b)
```

```
1 4
2 5
3 6
```