

Stage 1 - 급여에 수습 적용하기

1주차 때 만든 급여 계산기에 수습이 적용되는지, 미적용되는지 판단하여 급여를 더욱 정확하게 계산하도록 합니다.

수습 적용하기

카페에서 알바를 해보신 적이 있나요? 처음에 알바를 하게 되면, 커피를 만드는 방법을 배워야하기 때문에 수습이 적용되어서 시급을 받게 되는데요. 조금은 서럽게 느껴지죠. 그래도 수습 기간을 잘 버티게 되면, 여러 음료를 만드는 법도 배우고 시급도 제대로 받을 수 있으니 얼마나 좋나요! 이번 시간에는 우리가 수습 기간에 놓인 알바생들을 위해 지난 시간에 만든 급여 계산기를 개선해 보려고 합니다. 수습이 적용되면 10%의 수습 기간 패널티를 부여하는 코드를 추가해 줄 거예요. 잘 따라오세요!

```

pay_calculator x
/Users/duhui/venv/python_day2/bin/python /Users/duhui/PycharmPro
급여계산기 프로그램입니다!
시급을 입력해주세요 : 8000
일일근무시간 : 10
한달근무일수 : 20

수습을 적용하나요?
1. 적용
2. 미적용
1
예상 월급은 : 1440000원입니다.

1440000원으로 무엇을 할 수 있을까요?
PC방 (1200원 기준) : 1200시간
점심 (7000원 기준) : 205끼
영화 (11000원 기준) : 130편
노래방 (20000원 기준) : 72시간

열심히 일하시는 당신에게...
비록 세상이 그대를 속일지라도
슬퍼하거나 노여워하지 말아라
슬픈 날은 참고 견디라
기쁜 날이 오고야 말리니.

Process finished with exit code 0

```

수습을 적용한 급여 계산기

조건문이란

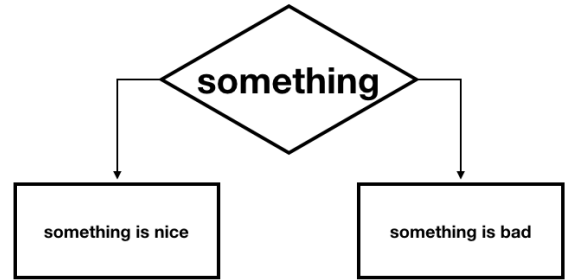
조건문이란 무엇인가?

프로그래밍적 사고력, 코딩적 사고력, 컴퓨팅적 사고력에서 가장 중요한 두가지가 있다면 조건문과 반복문을 뽑을 수 있습니다. 2주차 때는 조건문을 집중적으로 연습해 볼 계획이고요. 3주차 때는 반복문을 집중적으로 연습해 볼 계획입니다.

그렇다면 조건문이 뭘까요? 영어를 배울 때 조건문에 대해 배우죠. '만약 무엇이라면, 무엇이다.', 'if something is good, something is nice.' 이런식의 문장을 조건문이라고 하죠. 코딩에서도 마찬가지로입니다. 'if'를 사용하여 조건을 나누고 조건에 따라 다른 명령을 내리도록 합니다.

```
something = 'good'

if something is 'good':
    print("something is nice")
else:
    print("something is bad")
```



조건문 예시

조건문은 어떻게 사용하나요?

파이썬에서 조건문을 사용하는 방법에 대해 알아보시다.

조건문은 크게 if문과 else문으로 구성되어 있습니다. if문 뒤에는 참과 거짓을 판별할 수 있는 문장이 옵니다. if문 뒤에 오는 문장이 참이라면, if문 아래 오는 문장들이 수행이 되고, 거짓이라면 else문 아래에 오는 문장들이 수행됩니다.

콜론과 TAB

콜론은 if문에 대한 조건문장이 끝났음을 의미하고, 다음으로 그 조건에 해당되는 수행이 시작되는 의미입니다. 그 수행되는 코드들은 TAB으로 구분합니다.

```
something = 'good'

if something is 'good':
    print("something is nice")
else:
    print("something is bad")
```

콜론과 TAB의 사용

if문 뒤에 오는 문장 예시

example

```
1 True
2 # 언제나 True
```

example

```
1 False
2 # 언제나 False
```

example

```
1 a == 1
2 # a라는 변수가 1이면 True, 1이 아니면 False
```

example

```
1 a is 1
2 # a라는 변수가 1이면 True, 1이 아니면 False
```

example

```
1 a < 3
2 # a라는 변수가 3보다 작으면 True, 아니면 False
```

example

```
1 12 <= a < 19
2 # a라는 변수가 12보다 크거나 같고 19보다 작으면 True, 아니면 False
```

❶ '=='과 'is'는 언뜻보면 같은 행동을 하는 것 같습니다. 그러나 조금 차이가 있습니다. 'is'는 실제로 컴퓨터 메모리상에 올려진 위치가 같은지까지 비교하고, '=='은 컴퓨터 메모리에 올려진 위치가 달라도 값만 똑같으면 참이라고 판단합니다.

i 1주차 때 배웠던 자료형을 떠올려봅시다. 위의 예시 문장 모두가 불 형태의 자료형입니다. 참고적으로 모든 자료형들이 불 형태로 표현이 가능하다. 숫자의 경우 0을 제외한 모든 숫자가 True이고, 그 외의 자료형의 경우 비어있는 상태가 아니라면 모두 True를 나타냅니다.

급여 계산기 수습 적용

수습 적용 여부 입력받기

우리는 앞으로 조건문에 대한 정보를 입력받을 때, 숫자로 입력받을 예정입니다. 그러나 우리가 사용자로부터 받은 숫자는 실제 숫자형이 아니라 문자형으로 인식됩니다. 조건을 비교할 때 유의하여야 합니다.

숫자로 입력안내문 표시

```
calculators/pay_calculator.py
```

```
1 # 수습 적용 여부
2 practice = input('
3 수습을 적용하나요?
4 1. 적용
5 2. 미적용
6 ''')
```

숫자를 비교하기 위해 형변환

```
calculators/pay_calculator.py
```

```
1 # 수습 변수 추가
2 practice = int(practice)
```

수습을 적용한 급여 계산

드디어 조건문을 사용하여 급여를 계산해보겠습니다. 사용자로부터 입력받은 값이 '1'이라면 계산한 급여의 1/10을 수습 패널티로 적용하여 급여에서 제외시켜야 합니다. 그에 해당하는 코드는 아래와 같습니다.

calculators/pay_calculator.py

```
1  # 수습 적용
2  if practice == 1:
3      practice_price = profit // 10
4  else:
5      practice_price = 0
6
7  profit = profit - practice_price
```