

Control Statement

Control Statement

❖ Control Statement

- ✓ 프로그램은 위에서 아래로 한 문장씩 수행하는데 이러한 제어 흐름을 넘어가야 하는 경우도 있고 갈림길을 만나는 경우도 있으며 반복하기도 함
- ✓ 파이썬의 Control Statement은 if, for, while 등
- ✓ Expression(표현식)
 - 판단이 가능한 문장
- ✓ 파이썬 블록
 - 전체를 하나의 영역으로 간주하는 코드들의 모임으로 파이썬에서는 Control Statement의 영역, 함수, 클래스, 파일이 코드 블록이 될 수 있음
 - 작성 규칙
 - ◆ 내부 블록은 같은 거리만큼 들여쓰기(4칸을 권장)
 - ◆ 탭을 사용할 수 있지만 공백 과 탭을 함께 사용하는 것은 권장하지 않음
 - ◆ 들여쓰기 간격은 일정해야 함

Conditional Statement

❖ if

if expression :

code block

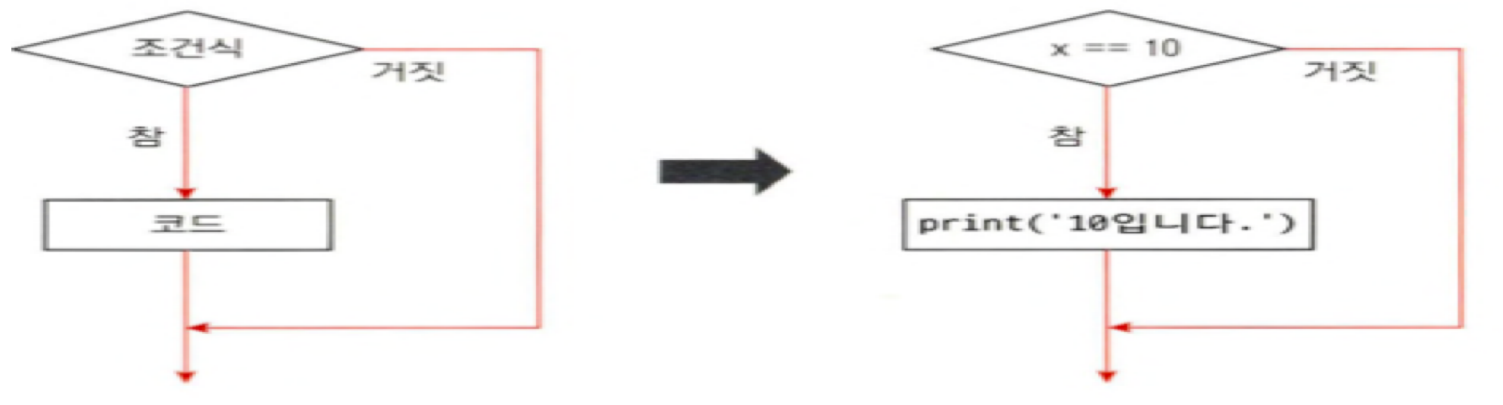
- ✓ 표현식(expression)의 결과가 True이면 if안의 code block이 실행되고 False이면 code block을 수행하지 않고 다음으로 진행
- ✓ 숫자 데이터는 0이면 False 0이 아니면 True로 간주
- ✓ 데이터의 모임은 데이터가 존재하면 True 그렇지 않으면 False

if 비가오면:

우산을 쓴다

외출을 한다.

=>비가오면 우산을 쓰고 외출을 하고 비가 오지 않으면 외출을 한다.



Conditional Statement

❖ if

```
var1 = True
if var1:
    print("1 - True 일 때 수행될 문장")
    print(var1)
```

#숫자 0은 False 로 간주

```
var2 = 0
if var2:
    print("2 - True 일 때 수행될 문장")
    print(var2)
```

```
var3 = [1]
if var3:
    print("3 - 데이터가 있으면 True")
    print(var3)
```

Conditional Statement

❖ if

```
var3 = []  
if var3:  
    print("4 - 데이터가 없으면 False")  
    print(var3)
```

```
var3 = None  
if var3:  
    print("3 - None 이어도 False")  
    print(var3)
```

```
print("프로그램 종료!")
```

1 - True 일 때 수행될 문장

True

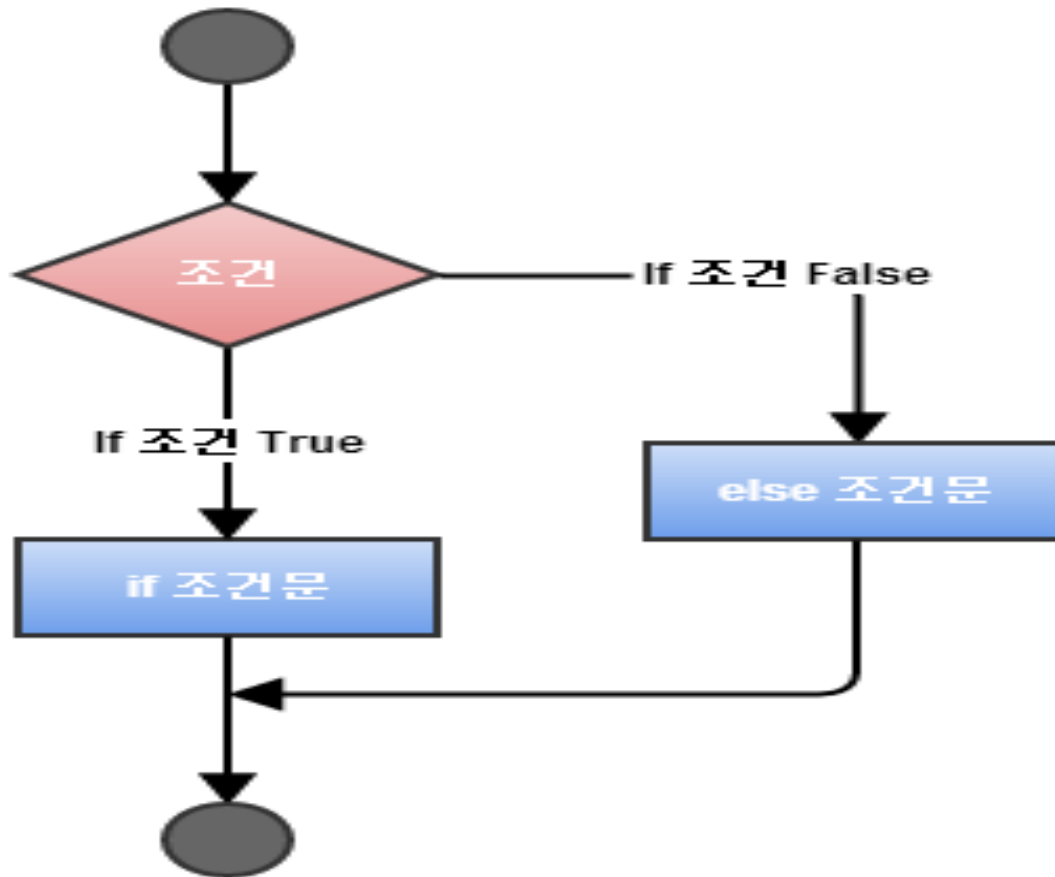
3 - 데이터가 있으면 True

[1]

프로그램 종료!

Conditional Statement

❖ if-else



Conditional Statement

❖ if-else

if expression :

code block

else:

code block

- ✓ expression이 True 일 때와 False일 때를 구분해서 수행
- ✓ True일 때는 if 블록의 내용을 수행하고 False일 때는 else 블록의 내용을 수행

if 아침 7시 이전에 도착하면:

스타벅스 커피

else:

할리스 커피

=>아침 7시 이전이면 스타벅스 커피 그렇지 않으면 할리스 커피

Conditional Statement

- ❖ 점수를 입력받아서 60점 이상이면 합격 아니면 불합격이라고 출력

```
score = int(input("점수를 입력하세요:"))  
if score >= 60:  
    print("합격입니다.")  
else:  
    print("불합격입니다.")
```

점수를 입력하세요:54
불합격입니다.

Conditional Statement

- ❖ 변수에 값 할당을 if, else로 축약 – 삼항 연산자 처럼 사용

```
x = 5
if x == 10:
    y = True
else:
    y = False
print(y)
```

```
x = 10
y = True if x == 10 else False
print(y)
```

False
True

Conditional Statement

- ❖ if - elif - else
if 와 함께 elif 절 사용

```
if 조건1:  
    코드블록  
...  
elif 조건2:  
    코드블록  
...  
elif 조건3:  
    코드블록  
...  
elif 조건4:  
    코드블록  
...  
else:  
    코드블록  
...
```

첫 번째 조건은 항상 if로 시작

두 번째 조건부터는 elif를 이용하며 여러 번 추가가 가능

마지막의 else는 생략 가능

Conditional Statement

- ❖ 1번을 누르면 아메리카노 2번을 누르면 카페라떼 3번을 누르면 에스프레소를 나머지를 누르면 제공하지 않는 메뉴라고 출력하기

```
menu = input("메뉴를 입력하세요:")  
if menu == '1':  
    print('아메리카노')  
elif menu == '2':  
    print('카페 라떼')  
elif menu == '3':  
    print('에스프레소')  
else:  
    print('잘못된 메뉴를 입력')
```

메뉴를 입력하세요:2
카페 라떼

Conditional Statement

- ❖ 점수를 입력받아서 수, 우, 미, 양, 가 를 출력해주는 코드

```
score = int(input("점수를 입력하세요:"))
```

```
if score >= 90 and score <= 100:
```

```
    print('수')
```

```
elif score >= 80 and score < 90:
```

```
    print('우')
```

```
elif score >= 70 and score < 80:
```

```
    print('미')
```

```
elif score >= 60 and score < 70:
```

```
    print('양')
```

```
elif score >= 0 and score < 60:
```

```
    print('가')
```

```
else:
```

```
    print('잘못 입력된 점수입니다.')
```

점수를 입력하세요:97

수

Conditional Statement

❖ switch

- ✓ 다른 언어의 switch가 존재하지 않기 때문에 dictionary를 이용해서 switch를 구현

```
switcher = {  
    0:'Sunday',  
    1:'Monday',  
    2:'Tuesday',  
    3:'Wednesday',  
    4:'Thursday',  
    5:'Friday',  
    6:'Saturday'  
}
```

```
print(switcher.get(3, 'Unknown Week'))
```

Wednesday

Conditional Statement

❖ switch

- ✓ 함수를 이용해서 switch를 구현

```
def switch(lang):  
    if lang == "JavaScript":  
        return "You can become a web developer."  
    elif lang == "PHP":  
        return "You can become a backend developer."  
    elif lang == "Python":  
        return "You can become a Data Scientist"  
    elif lang == "Solidity":  
        return "You can become a Blockchain developer."  
    elif lang == "Java":  
        return "You can become a mobile app developer"
```

```
print(switch("JavaScript"))  
print(switch("PHP"))  
print(switch("Java"))
```

You can become a web developer.
You can become a backend developer.
You can become a mobile app developer

Conditional Statement

❖ switch

- ✓ 3.10에서 추가된 문법

match term:

case pattern-1:

action-1

case pattern-2:

action-2

case pattern-3:

action-3

case _:

action-default

Conditional Statement

❖ switch

✓ 변경된 문법을 적용

```
lang = input("What's the programming language you want to learn? ")
```

```
match lang:
```

```
    case "JavaScript":
```

```
        print("You can become a web developer.")
```

```
    case "Python":
```

```
        print("You can become a Data Scientist")
```

```
    case "PHP":
```

```
        print("You can become a backend developer")
```

```
    case "Solidity":
```

```
        print("You can become a Blockchain developer")
```

```
    case "Java":
```

```
        print("You can become a mobile app developer")
```

```
    case _:
```

```
        print("The language doesn't matter, what matters is solving problems.")
```

Conditional Statement

- ❖ A 기업의 입사 시험은 필기시험 점수가 80점 이상이면서 코딩 시험을 통과해야 합격인데 필기시험 점수를 입력받아서 80점 이상이면 합격 그렇지 않으면 불합격이라고 출력하는 프로그램을 작성
- ❖ 표준 입력으로 국어, 영어, 수학, 과학 점수를 입력 받아서 합격 여부 출력하기
 - ✓ 입력받은 네 과목의 평균 점수가 80점 이상일 때 합격이라고 출력하고 그렇지 않으면 불합격이라고 출력하는 프로그램을 작성
 - ✓ 점수는 0점부터 100점 까지만 입력 받을 수 있으며 범위를 벗어났다면 잘못된 점수 를 출력하고 합격, 불합격 여부는 출력하지 않도록 작성
- ❖ 년도를 입력받아서 윤년 여부를 출력하는 프로그램을 작성
 - ✓ 윤년은 4의 배수이고 100의 배수가 아닌 경우 또는 400의 배수인 경우
 - ✓ 윤년이면 입력하신 해는 윤년입니다 로 출력하고 윤년이 아니면 입력하신 해는 윤년이 아닙니다 라고 출력하도록 작성

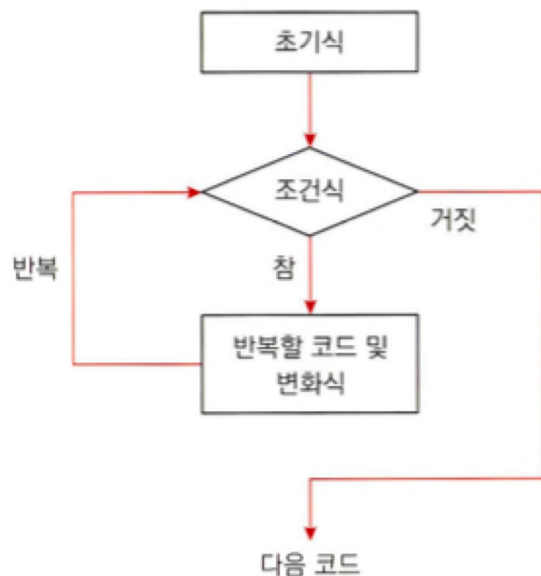
Conditional Statement

- ❖ 나이를 입력받아서 현재 요금에서 각 입력 데이터에 맞는 요금을 감소시킨 후 잔액을 출력하는 프로그램을 작성
 - ✓ 표준 입력으로 나이가 입력
 - ✓ 입력 값은 7 이상이 입력되어야 하는데 7이상이 아니면 잘못된 입력이라고 하고 잔액은 출력하지 않음
 - ✓ 현재 교통카드에 는 9,000원이 들어있음
 - ✓ 교통카드 시스템에서 시내버스 요금은 다음과 같음
 - 어린이(초등학생 7세 이상 12세 이하): 650원
 - 청소년(중 • 고등학생 13세 이상 18세 이하): 1,050원
 - 어른(일반 19세 이상): 1,250

Loop

- ❖ while – 반복문
while 표현식:
코드블록

=> 표현식이 거짓이 아닌 동안 코드 블록을 수행



```
limit = int(input('몇 번 반복할까요: '))
```

```
count = 0
while count < limit:
    count = count + 1
    print('%d회 반복.' % (limit))
```

```
print('반복이 종료되었습니다.')
```

몇 번 반복할까요: 3
3회 반복.
3회 반복.
3회 반복.
반복이 종료되었습니다.

Loop

❖ 반복문 사용

#excel1.xls 부터 excel3.xls 파일을 읽기

```
print('--반복문을 사용하지 않는 경우')  
print('excel1.xls 파일 읽기')  
print('excel2.xls 파일 읽기')  
print('excel3.xls 파일 읽기')
```

```
print('--반복문을 사용하는 경우')  
i = 1  
while i <= 3:  
    print('excel%d.xls 파일 읽기'%(i))  
    i = i+1
```

--반복문을 사용하지 않는 경우

excel1.xls 파일 읽기

excel2.xls 파일 읽기

excel3.xls 파일 읽기

--반복문을 사용하는 경우

excel1.xls 파일 읽기

excel2.xls 파일 읽기

excel3.xls 파일 읽기

Loop

❖ 구구단 2단 출력

```
i = 0
while i < 9:
    i = i + 1
    print('2 * %d = %d'%(i, 2*i))
```

2 * 1 = 2

2 * 2 = 4

2 * 3 = 6

2 * 4 = 8

2 * 5 = 10

2 * 6 = 12

2 * 7 = 14

2 * 8 = 16

2 * 9 = 18

Loop

- ❖ 하나의 단어를 입력받아서 입력받은 단어를 한겨레 신문사에서 검색해서 3개 페이지 기사를 가져오기 위한 주소 만들기
 - ✓ `http://search.hani.co.kr/Search?command=query&keyword=검색어&media=news&submedia=&sort=d&period=all&datefrom=2000.01.01&dateto=2019.06.24&pageseq=페이지번호`
 - ✓ 페이지 번호는 0부터 시작
#한겨레 주소 만들기
`keyword = input("검색어를 입력하세요 :")`
`pagenumber = 0`
`while pagenumber < 3:`
 `url = "http://search.hani.co.kr/Search?command=query&keyword="+ keyword +`
 `"&media=news&submedia=&sort=d&period=all&datefrom=2000.01.01&dateto=2019.06.`
 `24&pageseq=" + str(pagenumber)`
 `print(url)`
 `pagenumber = pagenumber + 1`
- 검색어를 입력하세요 :python
`http://search.hani.co.kr/Search?command=query&keyword=python&media=news&submedia=&sort=d&period=all&datefrom=2000.01.01&dateto=2019.06.24&pageseq=0`
`http://search.hani.co.kr/Search?command=query&keyword=python&media=news&submedia=&sort=d&period=all&datefrom=2000.01.01&dateto=2019.06.24&pageseq=1`
`http://search.hani.co.kr/Search?command=query&keyword=python&media=news&submedia=&sort=d&period=all&datefrom=2000.01.01&dateto=2019.06.24&pageseq=2`

Loop

❖ while

✓ while – else

```
while 표현식:  
    코드블록1  
else:  
    코드블록2
```

- 표현식이 거짓이 아니면 코드블록 1을 수행하고 중간에 종료되지 않고 반복문 수행을 종료한 경우에 코드블록 2를 수행

Loop

❖ 무한 반복

- ✓ while 문의 표현식이 절대 False가 될 수 없는 루프

```
while True:  
    코드블록
```

Loop

- ❖ 3번 반복 수행한 후 다른 동작 수행

```
i = 0
while i < 3:
    i = i + 1
    print('2 * {0} = {1}'.format(i, 2*i))
else:
    print("반복문을 정상적으로 종료하였습니다.")
```

2 * 1 = 2

2 * 2 = 4

2 * 3 = 6

반복문을 정상적으로 종료하였습니다.

Loop

❖ for

- ✓ for는 순서열을 순회하면서 각 데이터에 로직을 수행한 후 순서열의 끝에 도달하면 반복을 종료함.
- ✓ while과 마찬가지로 else 사용 가능
- ✓ `__iter__` 가 구현된 객체를 for의 순서열에 사용

```
for 반복변수 in 순서열:  
    코드블록
```

```
for i in (1, 2, 3):  
    print(i)
```

변수 i에는 매 반복마다 튜플 (1, 2, 3)의 요소들이 차례대로 복사
튜플의 길이는 3이므로 이 for문은 3번 반복 수행

```
1  
2  
3
```

Loop

- ❖ () 안의 내용을 순서대로 가져와서 출력

```
for i in (1, 2, 3) :  
    print(i)  
else:  
    print("반복문을 정상적으로 수행했습니다.")
```

1
2
3

반복문을 정상적으로 수행했습니다.

Sequence

❖ range

- ✓ range([시작값], 종료값, [간격])를 이용해서 일정한 패턴의 수열 생성
- ✓ 숫자를 하나만 대입하면 종료 값만 있는 것으로 간주하고 시작 값은 0 간격은 1로 설정
- ✓ 2개의 숫자를 대입하면 시작 값과 종료 값만 있는 것으로 간주하고 간격은 1로 설정
- ✓ 종료 값은 포함되지 않음

```
print(list(range(10)))#종료 값만 있는 경우
print(list(range(5, 10)))#시작 값과 종료 값만 있는 경우
print(list(range(10, 0, -1)))#3가지 값을 모두 대입한 경우
#10에서 20까지의 짝수를 출력하는 경우
for i in range(10,20,2):
    print(i, end="Wt")
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[5, 6, 7, 8, 9]

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

10

12

14

16

18

Control Statement Nesting

❖ Control Statement 중첩

- ✓ Control Statement 안에 다른 Control Statement을 사용 가능
- ✓ 바깥쪽 Control Statement에서 시작해서 안쪽 Control Statement이 전부 수행되고 나면 바깥쪽 Control Statement의 시작부분으로 이동해서 수행



Control Statement Nesting

❖ 구구단 전체 출력

```
print("★ 구구단을 출력★\n")
for x in range(1, 10):
    for y in range(2, 10):
        print(y,"X",x,"=",format(x*y,'2d'),end=" ")
    print("")
```

★ 구구단을 출력★

```
2 X 1 = 2 3 X 1 = 3 4 X 1 = 4 5 X 1 = 5 6 X 1 = 6 7 X 1 = 7 8 X 1 = 8 9 X 1 = 9
2 X 2 = 4 3 X 2 = 6 4 X 2 = 8 5 X 2 = 10 6 X 2 = 12 7 X 2 = 14 8 X 2 = 16 9 X 2 = 18
2 X 3 = 6 3 X 3 = 9 4 X 3 = 12 5 X 3 = 15 6 X 3 = 18 7 X 3 = 21 8 X 3 = 24 9 X 3 = 27
2 X 4 = 8 3 X 4 = 12 4 X 4 = 16 5 X 4 = 20 6 X 4 = 24 7 X 4 = 28 8 X 4 = 32 9 X 4 = 36
2 X 5 = 10 3 X 5 = 15 4 X 5 = 20 5 X 5 = 25 6 X 5 = 30 7 X 5 = 35 8 X 5 = 40 9 X 5 = 45
2 X 6 = 12 3 X 6 = 18 4 X 6 = 24 5 X 6 = 30 6 X 6 = 36 7 X 6 = 42 8 X 6 = 48 9 X 6 = 54
2 X 7 = 14 3 X 7 = 21 4 X 7 = 28 5 X 7 = 35 6 X 7 = 42 7 X 7 = 49 8 X 7 = 56 9 X 7 = 63
2 X 8 = 16 3 X 8 = 24 4 X 8 = 32 5 X 8 = 40 6 X 8 = 48 7 X 8 = 56 8 X 8 = 64 9 X 8 = 72
2 X 9 = 18 3 X 9 = 27 4 X 9 = 36 5 X 9 = 45 6 X 9 = 54 7 X 9 = 63 8 X 9 = 72 9 X 9 = 81
```

Control Statement Nesting

❖ *을 5개씩 5개 출력

```
for x in range(0, 5):  
    for y in range(0, 5):  
        print('{0}'.format('*'), end="")  
    print("")
```

Control Statement Nesting

❖ 삼각형 모양의 별 출력

```
for x in range(0, 5):  
    for y in range(0, x+1):  
        print('{0}'.format('*'), end='')  
    print("")
```

*

**

Control Statement Nesting

❖ 삼각형 모양의 별 출력

```
for x in range(0, 5):  
    for y in range(0, 5 - x):  
        print('{0}'.format(' '), end="")  
  
    for y in range(0, 2 * x + 1):  
        print('{0}'.format('*'), end="")  
    print("")
```

```
*  
***  
*****  
*****  
*****
```

기타

❖ continue

- ✓ 반복문이 실행하는 코드 블록의 나머지 부분을 실행하지 않고 다음 반복으로 건너가도록 흐름을 조정하는 예약어
- ✓ 반복문 안에서 if 와 같이 사용되는 것이 일반적

```
for i in range(10):  
    if i % 2 == 1:  
        continue
```

```
    print(i)
```

i가 홀수일 때 코드블록의 나머지 부분을 실행하지 않고 다음 반복으로 바로 건너감

continue가 실행되는 경우에는 이 코드는 실행되지 않음

```
>continue.py
```

```
0  
2  
4  
6  
8
```

기타

❖ break

- ✓ 반복문을 중단시키는 기능 수행
- ✓ 반복문 안에서 if와 같이 사용하는 것이 일반적

```
i = 0
while(True):
    i = i+1
    if i == 1000:
        print('i가 %d값이 되었습니다. 반복문을 중단합니다.'%(i))
        break
    print(i)
```

```
> break.py
0
1
2
...
997
998
999
i가 1000이 되었습니다. 반복문을 중단합니다.
```

기타

❖ for - break - else

```
for i in (1, 2, 3):  
    if(i%3 == 0):  
        break  
    print(i)  
else:  
    print("반복문을 정상적으로 종료한 경우에만 수행합니다.")  
print("무조건 수행합니다.")
```

1

2

무조건 수행합니다.

연습문제

- ❖ 1부터 100까지 짝수의 합 구하기
- ❖ 교통 카드 잔액 출력
 - ✓ 표준 입력으로 인원 수가 입력됨
 - ✓ 1인당 요금은 1,350원으로 교통카드를 사용할 때마다 요금을 차감하고 각 줄에 잔액을 출력하는 프로그램을 작성
 - ✓ 최초 금액은 10,000 원으로 설정하고 출력하지 않음
 - ✓ 잔액은 음수가 될 수 없으며 잔액이 부족하면 출력을 종료
- ❖ 2부터 1000까지 Prime(소수) 의 개수 구하기
 - ✓ Prime은 1과 자기 자신으로만 나누어지는 수
 - ✓ 2부터 자신의 절반이 되는 숫자까지 나누어 떨어지지 않으면 Prime
- ❖ 2부터 1000까지 완전수 출력 하기
 - ✓ 완전수는 자기 자신을 제외한 약수의 합이 자기 자신과 동일한 수
 - ✓ $6 : 1 + 2 + 3$

연습문제

- ❖ n번째 피보나치 수열의 값 출력하기
 - ✓ 피보나치 수열은 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ..
 - ✓ 첫번째 와 두번째 숫자는 1
 - ✓ 세번째 숫자부터는 이전 2개의 항의 합

