

## Chapter3. 조건문

### 3-1블 자료형과 if 조건문

#### 불 만들기: 비교 연산자

```
ex)
print(10==100)
>>False
print(10 != 100)
>>True
이는 문자열에서도 적용할 수 있습니다
print("가방"=="가방")
>>True
print("가방" < "하마")
>>True
```

#### 불 연산하기 : 논리 연산자

-파이썬에는 세 개의 논리 연산자가 있습니다  
-not, and, or

```
not 연산자 : 참과 거짓을 반대로 바꿀때 사용합니다
ex)
print(not True)
>>False
```

and 연산자 : 양변값이 모두 참일 때만 True를 결과로 냅니다  
or 연산자 : 둘중 하나만 참이어도 True가 나옵니다

#### if 조건문이란?

-조건에 따라 코드를 실행하거나, 실행하지 않게 만들고 싶을 때 사용하는 구문입니다.

기본적인구조는 다음과 같습니다

```
"""
if 불 값이 나오는 표현식 :
    불 값이 참일때 실행할 문장
"""
```

```
만약 if 뒤에 불 값이 거짓인 경우는 아무것도 실행되지 않습니다
ex)
number = 123
if number > 0:
    print("양수입니다")
>>양수입니다
```

#### 날짜/시간 활용하기

```
ex)
import datetime
#현재 날짜/시간을 구합니다
now = datetime.datetime.now()

if now.hour < 12:
    print("현재 시각은 {}로 오전입니다".format(now.hour))
if now.hour >=12:
    print("현재 시각은 {}로 오후입니다".format(now.hour))
>>>현재 시각은 19로 오후입니다
```

#### 컴퓨터의 조건

여기서는 if문의 불값이 나오는 조건문을 어떻게 만들것인가에 대해 생각해 보는 시간을 가져보겠습니다  
홀수와 짝수를 구분해봅시다

```
ex)
number = input("정수 입력>")
number = int(number)
```

```
if number % 2 == 0 :
    print("짝수입니다")
```

```
if number % 2 == 1 :
    print("홀수입니다")
>>정수입력>52
>>짝수입니다
```

### 3-2 if~else와 elif 구문

#### else 조건문의 활용

-if 조건문의 조건이 거짓일 때 실행되는 부분입니다  
ex)

```
number = int(input("정수 입력>"))

if number % 2 == 0:
    print("짝수 입니다")
else :
    print("홀수 입니다")
>>정수 입력> 13
>>홀수 입니다
```

## elif 구문

-세 개 이상의 조건을 연결해서 사용할때 elif구문을 사용합니다. 그리고 elif구문은 if조건문과 else구문 사이에 입력합니다.

ex)

```
import datetime
```

```
now = datetime.datetime.now()
```

```
month = now.month
```

```
if 3<= month <=5 :
```

```
    print("현재는 봄입니다.")
```

```
elif 6<= month <=8 :
```

```
    print("현재는 여름입니다.")
```

```
elif 9<= month <=11 :
```

```
    print("현재는 가을입니다.")
```

```
else :
```

```
    print("현재는 겨울입니다.")
```

## False로 변환되는 값

-if 조건문의 매개변수에 불이 아닌 다른 값이 올때는 자동으로 이를 불로 변환해서 처리합니다.

False로 변환되는 값은 None, 0, 0.0,

빈컨테이너입니다.이 외에는 모두 True로 변환됩니다.

ex)

```
if 0 :
```

```
    print("0은 True로 변환됩니다.")
```

```
else :
```

```
    print("0은 False로 변환됩니다")
```

```
>>0은 False로 변환됩니다
```

## pass 키워드

-프로그래밍을 하다보면 일단 프로그래밍의 전체 골격을 잡아 놓고 내부에서 처리할 내용은 차근차근 생각하며 만들겠다는 의도로 다음과 같이 코딩하는 경우가 많습니다.

ex)

```
number = int(input("정수 입력>"))
```

```
if number > 0 :
```

```
    pass
```

```
else :
```

```
    pass
```

만약 if나 else 이후에 아무 내용도 쓰지않으면,

**indentationError**가 발생합니다.