

UNIT 13

if 조건문으로 특정 조건일 때
코드 실행하기

13 if 조건문으로 특정 조건일 때 코드 실행하기

» if 조건문으로 특정 조건일 때 코드 실행하기

- 조건문을 사용하면 조건에 따라 다른 코드를 실행할 수 있음
- 세탁기에 빨래를 넣고 돌렸다면 다음과 같은 조건문을 만들어보자

```
if 세탁 완료 소리가 울리면:  
    빨래를 꺼내서 말린다.
```

- 다음과 같이 날씨에 따라 행동할 수도 있음

```
if 비가 온다면:  
    우산을 가지고 나간다.
```

```
if 날씨가 춥다면:  
    코트를 입고 나간다.
```

```
if 날씨가 덥다면:  
    반소매에 얇은 옷을 입고 나간다.
```

13.1 if 조건문 사용하기

» if 조건문 사용하기

- if 조건문은 if에 조건식을 지정하고 :(콜론)을 붙이며 다음 줄에 실행할 코드가 옵니다
- 이때 실행할 코드는 반드시 들여쓰기를 해야 함

```
if 조건식:  
    코드
```

- IDLE의 파이썬 셸에서 if 조건문을 사용해보자

```
>>> x = 10  
>>> if x == 10:  
...     print('10입니다.')  
...  
10입니다.
```

13.1 if 조건문 사용하기

» if 조건문 사용하기

- 만약 if 다음 줄에서 들여쓰기를 하지 않으면 들여쓰기 에러가 발생함
- 이 항상 이 부분을 주의하자

```
>>> x = 10
>>> if x == 10:
... print('10입니다.')
File "<stdin>", line 2
    print('10입니다.')
    ^
IndentationError: expected an indented block
```

- 참고로 IDLE의 파이썬 셸에서는 자동으로 들여쓰기가 되지만, 콘솔(터미널, 명령 프롬프트)에서 실행한 파이썬 셸에서는 자동으로 들여쓰기가 되지 않으므로 반드시 들여쓰기를 해줌

13.1 if 조건문 사용하기

» if 조건문의 기본 형태와 실행 흐름 알아보기

- 파이썬에서 if 조건문은 if 조건식: 형식으로 사용하며 그다음 줄에는 들여쓰기를 한 뒤 조건식이 만족할 때 실행할 코드를 넣음
- 이 조건식이 만족할 때 실행할 코드를 if 본문(if body)이라고 부름

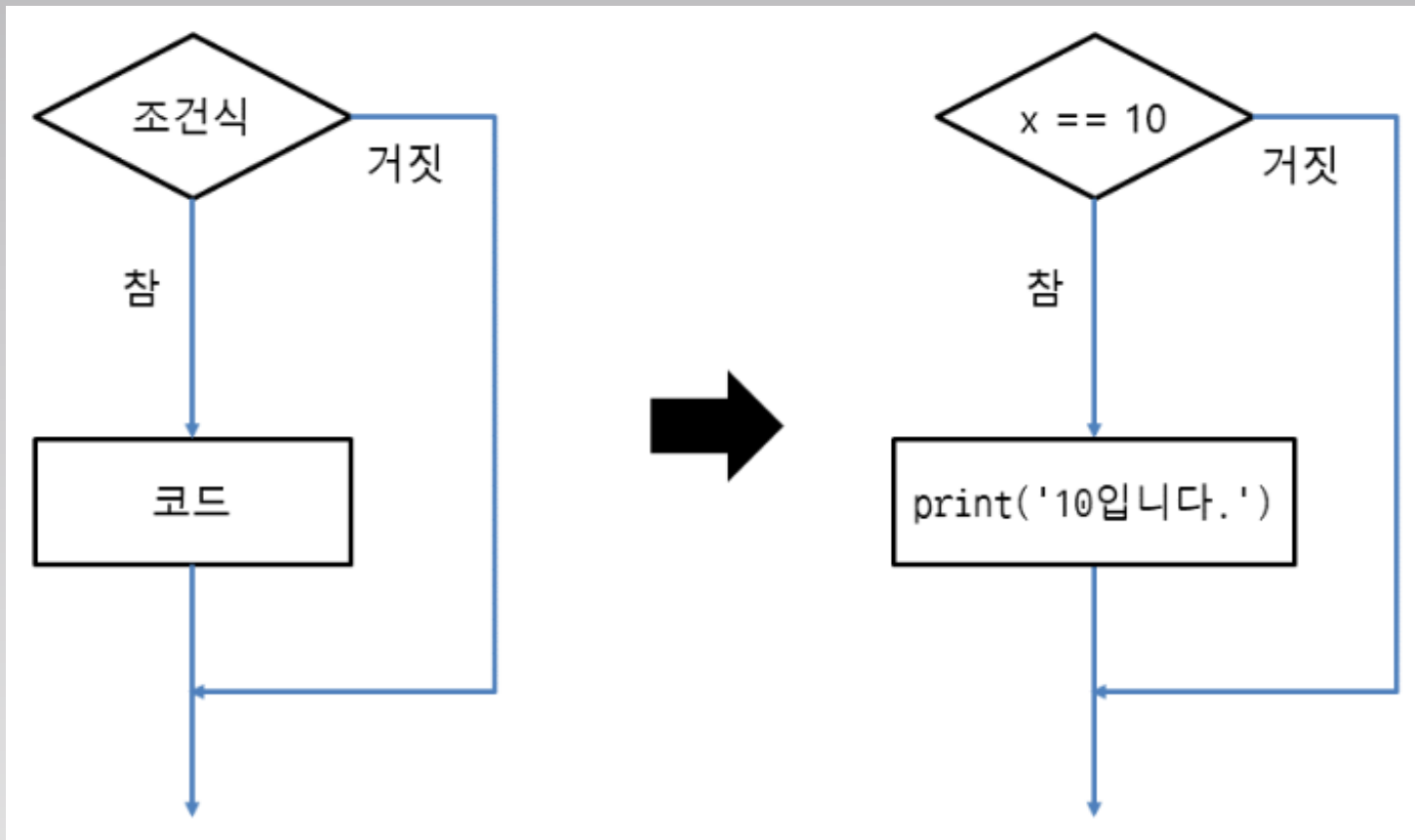
▼ 그림 if 조건문의 기본 형태

```
if x == 10:
    print('10입니다.')
```

The diagram illustrates the basic structure of an if statement in Python. It shows the code `if x == 10:` followed by an indented line `print('10입니다.')`. Annotations with arrows point to specific parts of the code: '조건식' (Condition) points to `x == 10`; '콜론' (Colon) points to the colon at the end of the first line; '조건식이 만족할 때 실행할 코드 (if 본문)' (Code to be executed when the condition is satisfied (if body)) points to the `print` statement; and '들여쓰기 4칸' (Indentation 4 spaces) points to the indentation of the `print` statement.

13.1 if 조건문 사용하기

▼ 그림 if 조건문의 실행 흐름



13.1 if 조건문 사용하기

» if 조건문을 사용할 때 주의할 점

- =은 할당, ==은 비교
- if에 =을 사용하면 문법 에러가 발생함

```
>>> if x = 10:  
    File "<stdin>", line 1  
        if x = 10:  
            ^  
SyntaxError: invalid syntax
```

13.1 if 조건문 사용하기

» if 조건문을 사용할 때 주의할 점

- 조건식 끝에 :을 빠뜨리는 실수도 자주하니 :도 확인해주자
- 조건식 끝에 :을 빠뜨렸을 때의 모습임

```
>>> if x == 10
      File "<stdin>", line 1
        if x == 10
            ^
SyntaxError: invalid syntax
```

- 문법 에러가 발생하면 콘솔에서는 잘못된 코드 아래에 ^가 표시되고, IDLE에서는 빨간색으로 표시되므로 자신이 실수한 부분을 쉽게 찾을 수 있음

13.1 if 조건문 사용하기

» if 조건문에서 코드를 생략하기

```
>>> x = 10
>>> if x == 10:
...     pass
...
>>>
```

- if 다음 줄에 pass라는 특별한 키워드를 넣었음
- pass는 아무 일도 하지 않고 그냥 넘어간다는 뜻
- pass만 넣고 나중에 할 일은 주석으로 남겨놓는 방식임

```
if x == 10:
    pass    # TODO: x가 10일 때 처리가 필요함
```

13.2 if 조건문과 들여쓰기

» if 조건문과 들여쓰기

- 파이썬은 들여쓰기도 문법으로 정해져 있으며 if 조건문도 들여쓰기가 중요함

if_indent_error.py

```
x = 10

if x == 10:
    print('x에 들어있는 숫자는')
    print('10입니다.')    # unexpected indent 에러 발생
```

13.2 if 조건문과 들여쓰기

» if 조건문과 들여쓰기

- 앞의 예제는 두 번째 print 부분에서 unexpected indent 에러가 발생함
- 올바른 코드로 고쳐 보면 다음과 같이 두 번째 print도 들여쓰기 4칸으로 만들어줌

if_indent.py

```
x = 10

if x == 10:
    print('x에 들어있는 숫자는')
    print('10입니다.')
```

실행 결과

```
x에 들어있는 숫자는
10입니다.
```

13.2 if 조건문과 들여쓰기

» if 조건문과 들여쓰기

- 첫 번째 print만 들여쓰기를 하고, 두 번째 print는 들여쓰기를 하지 않으면 의도치 않은 동작이 됨

if_wrong_indent_true.py

```
x = 10

if x == 10:
    print('x에 들어있는 숫자는')
print('10입니다.')
```

실행 결과

```
x에 들어있는 숫자는
10입니다.
```

13.2 if 조건문과 들여쓰기

» if 조건문과 들여쓰기

- print 두 개가 모두 실행됨
- 다음과 같이 x가 5라면 어떻게 될까?

if_wrong_indent_false.py

```
x = 5          # x에 5를 할당

if x == 10:    # x가 5라서 조건식을 만족하지 않음
    print('x에 들어있는 숫자는')
print ('10입니다.')
```

실행 결과

10입니다.

13.2 if 조건문과 들여쓰기

» if 조건문과 들여쓰기

- print를 한 줄 띄워보면 왜 그런지 알 수 있음

```
x = 5

if x == 10:    # x가 5라서 조건식이 만족하지 않음
    print('x에 들어있는 숫자는')

print('10입니다.')    # 위의 if와는 상관없는 코드
```

13.3 중첩 if 조건문 사용하기

» 중첩 if 조건문 사용하기

- 변수의 값이 10 이상이면 '10 이상입니다.'를 출력한 뒤 15이면 '15입니다.', 20이면 '20입니다.'를 출력함

if_if.py

```
x = 15

if x >= 10:
    print('10 이상입니다.')

    if x == 15:
        print('15입니다.')

    if x == 20:
        print('20입니다.')
```

실행 결과

```
10 이상입니다.
15입니다.
```

13.3 중첩 if 조건문 사용하기

» 중첩 if 조건문 사용하기

- x가 10 이상일 때 '10 이상입니다.'를 출력하는 코드임

```
if x >= 10:  
    print('10 이상입니다.')
```

- if의 조건식에 따라 코드를 실행해야 하므로 print는 들여쓰기를 했음
- if 안쪽의 if를 보면 들여쓰기가 되어 있음

```
if x >= 10:  
    print('10 이상입니다.')  
    if x == 15:  
        print('15입니다.')  
    if x == 20:  
        print('20입니다.')
```


13.3 중첩 if 조건문 사용하기

» 중첩 if 조건문 사용하기

- 들여쓰기가 된 if x == 15:와 if x == 20:은 처음에 나온 if x >= 10:에 속한 코드임
- if x >= 10:의 조건식이 만족해야만 실행되는 코드임
- 안쪽의 if에 속한 print는 들여쓰기를 한 번 더 해줌

```
if x >= 10:
    print('10 이상입니다.')

    if x == 15:
        print('15입니다.')

    if x == 20:
        print('20입니다.')
```

13.4 사용자가 입력한 값에 if 조건문 사용하기

» 사용자가 입력한 값에 if 조건문 사용하기

if_input.py

```
x = int(input())          # 입력받은 값을 변수에 저장

if x == 10:               # x가 10이면
    print('10입니다.')    # '10입니다.'를 출력

if x == 20:               # x가 20이면
    print('20입니다.')    # '20입니다.'를 출력
```

- 스크립트 파일을 실행한 뒤 10을 입력하고 엔터 키를 누르자

실행 결과

```
10 (입력)
10입니다.
```

13.4 사용자가 입력한 값에 if 조건문 사용하기

» 사용자가 입력한 값에 if 조건문 사용하기

- 입력 값이 x에 저장되고, if로 x가 10인지 비교함
- 10을 입력했으므로 '10입니다.'가 출력됨
- 20과 다른 값을 입력하여 if가 동작하는 모습을 살펴보자
- if 조건문은 조건식이 만족했을 때 코드를 실행한다는 점이 중요함
- if 조건문은 들여쓰기에 따라 문법 에러가 발생하거나, 의도치 않은 동작이 나올 수 있으므로 들여쓰기 규칙을 정확히 익혀 두자