



흐름 제어

Python 기본 사항

발표자 이름

날짜

학습 내용

강의 핵심 내용

학습 내용:

- 프로그램에서 흐름 제어의 목적과 용도를 설명합니다.
- 조건 스테이트먼트를 사용해 평가된 기준에 따라 코드를 실행합니다.
- 루프를 사용해 기준을 충족할 때까지 코드를 반복 실행합니다.
- `Input()` 함수로 사용자가 프로그램을 실행할 때 사용할 입력에 대한 메시지를 표시합니다.



흐름 제어

흐름 제어는 명령형 프로그램의 개별 스테이트먼트 또는 지침, 함수 호출이 실행되거나 평가되는 순서입니다.

제어는
인터프리터가
코드를 실행하는
순서입니다.

다양한
스테이트먼트로
이 순서를 제어할
수 있습니다.

경우에 따라서는
조건이 충족되면
특정 액션이
발생되기를
바랍니다.
그때 조건
스테이트먼트를
사용합니다.

If: 키워드 **if**는 만약 이 조건이 충족되면 이 액션을 완료하라는 의미입니다.

Elif: 키워드 **elif**는 else if의 약어입니다. 조건이 true가 아니면 다른 액션을 완료하라는 의미입니다(한 블록에 여러 개의 **elif** 스테이트먼트를 포함할 수 있음).

Else: 키워드 **else**는 위 조건 중 하나도 충족되는 것이 없으면 이 액션을 완료하라는 의미입니다.

조건 스테이트먼트: 예제

일상에서 조건문이 사용되는 예입니다.

이 시간에 해당하면(if) 상점이 문을 엽니다. 그렇지 않으면(else) 상점이 문을 닫습니다.

연료 탱크의 연료가 특정 수준 이하로 떨어지면(if) 연료 부족 경고가 켜집니다.

직원이 배지를 갖고 있으면(if) 건물 출입을 허락합니다. 그렇지 않으면(else) 문이 잠깁니다.

어떤 사람의 나이가 18세보다 많으면(if, greater than) 성인으로 간주합니다. 18세보다 많지 않고 13~18세이면(else if) 십대로 간주합니다. 그렇지 않으면(else) 어린이로 간주합니다.

코드의 조건문

나이와 성인 구분에 대한 앞의 예를 Python에 코드로 작성해 보겠습니다. 조건문을 사용할 때에는 들여쓰기가 중요합니다. 관련된 조건문을 그룹화하거나 중첩해야 합니다.

```
if age > 18:  
    ...     print("adult")  
else:  
    ...     print("child")
```

어떤 나이를 평가하고 싶은지 Python에 알려줘야 합니다. **연령**이라는 변수를 만들고 변수에 숫자 값을 할당합니다. 조건 스테이트먼트를 작성하기 **전에** 이 단계를 완료해야 합니다.

코드의 조건문(계속)

코드가 이 예처럼 보일 것입니다.

```
age = 21

if age >18:
    print("adult")
else:
    print("child")
```

참고:

- 조건문 **다음**에 오는 콜론이 중요합니다.
- 여기에서 **print** 함수를 사용합니다. 이 코드를 실행하면 Python이 어떤 내용을 출력할 거라고 생각하십니까?

데모: 조건문



1. conditionals.py라는 새 파일을 만듭니다.
2. Python에서 'if', 'elif', 'else' 스테이트먼트를 사용해 조건문 집합을 만듭니다. 이는 다음과 같이 만들어야 합니다.

바나나가 5개 이상이면 'I have a bunch of bananas(나는 바나나가 많이 있다)'를 출력합니다.

바나나가 1~4개이면 'I have a small bunch of bananas(나는 바나나가 조금 있다)'를 출력합니다.

바나나가 없으면 'I have no bananas(나는 바나나가 없다)'를 출력합니다.

3. 위 조건을 사용하여 조건문에 print 스테이트먼트를 할당합니다.
4. 바나나의 개수를 나타내는 변수를 만듭니다.
5. 코드를 실행하여 결과를 관찰합니다.

루프

- 루프는 Python에 코드 블록을 반복적으로 실행하도록 지시하는 기법입니다.
- Python은 특정 횟수 **에 대해**(for) 또는 특정 조건이 true **일 때**(while) 루프를 실행합니다.
- Python에는 **for**와 **while**의 두 가지 루프 유형이 있습니다.

While 루프

- **While** 루프는 무제한 실행될 수 있으므로 루프를 중지할 조건을 포함해야 합니다. 중지할 조건을 포함하지 않으면 코드가 무한 루프를 생성합니다.
- 일반적으로 루프에는 **반복 카운터**를 사용합니다. 루프가 완료되면 카운터가 증가하거나 감소합니다. 카운터가 특정 횟수에 도달하면 루프가 중지됩니다.

예:

```
counter = 0

while counter <= 3:
    print("I love learning Python!")
    counter = counter + 1
```

데모: While 루프

1. **while.py**라는 새 파일을 만듭니다.
2. 위 슬라이드의 코드로 **while** 루프를 만듭니다.
3. 루프에 **print** 스테이트먼트를 할당합니다.
4. 코드를 실행하고 결과를 관찰합니다.



목록

```
>>> [1, 2, 3]
>>> ["alice", "bob"]
>>> [1, "alice", 2, "bob"]
>>> []
```

- 목록은 변경 가능한 데이터 유형입니다. 목록에는 여러 가지 데이터 유형(strings, ints, floats, 다른 목록)이 포함될 수 있습니다.
- 목록은 양쪽 끝에 대괄호([])로 표시합니다.
- 값은 대괄호 안에 들어가며 쉼표로 구분합니다.
- 목록 안에는 0개부터 무제한으로 항목의 개수를 포함할 수 있습니다.

데모: 목록

1. **lists.py**라는 파일을 만듭니다.
2. 목록을 만들고 목록에 세 개 이상의 항목을 포함합니다. 데이터 유형을 다양하게 사용하거나 여러 개의 목록을 만들어 보십시오.
3. 목록을 출력합니다.



For 루프

- **For** 루프는 **<thing>**의 각 요소에 대해 특정 태스크를 수행하도록 하는 형식입니다.
- 실제 사례를 들어 보면 다음과 같습니다.

레시피의 모든 달걀에
대해(for) 밀가루를 2컵
추가합니다.

트럭의 모든 패키지에
대해(for) 2kg을
추가합니다.

주문한 모든 햄버거에
대해(for) 인벤토리에서
1개를 뺍니다.

루프와 목록

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

- **For** 루프와 목록은 함께 사용하기 좋습니다.
- 목록의 모든 항목에 대해 Python이 해당 항목을 출력합니다.
 - **num**을 호출할 때마다 목록에서 **num**에 값(차례로 1, 2, 3)을 할당합니다.
 - 그러면 루프가 값을 출력합니다.
 - 값의 목록 전체를 지정하고 나면 루프가 중지됩니다.

데모: For 루프

1. **lists.py** 파일을 엽니다.
2. 목록을 출력하는 **for** 루프를 만듭니다.
3. 코드를 실행하고 결과를 관찰합니다.



사전

```
>>> {"key": "value"}  
>>> {0:100, 1:200, 2:999}  
>>> {0:{1:2}}  
>>> myDict = {}
```

- 사전에는 값에 연결된 변경 불가능한 키가 포함되어 있습니다. 키는 변경 불가능한 데이터 유형이어야 합니다.
- 사전은 서로 중첩될 수 있습니다.
- 빈 사전을 만들려면 안에 아무것도 없는 중괄호 한 쌍({})을 사용합니다.
- 키는 콜론으로 값을 구분합니다.
`{"key": "value"}`
- 키별로 사전의 값을 가져옵니다.
`myDict.get("key")` or `myDict["key"]`

데모: 사전

1. **dictionary.py**라는 파일을 만듭니다.
2. 사전을 하나 만들고 그 안에 네 개의 **키:값** 쌍을 만듭니다.
3. 사전의 세 번째 값을 출력합니다.
4. 사전 전체를 출력합니다.



Input

```
>>> response =  
input("what's your name? ")  
what's your name? Jorge  
>>> response  
'Jorge'
```

- **input()** 함수는 사용자에게 텍스트를 입력하도록 요청하고 값을 변수에 저장합니다.
- 선택 인수 중 하나는 문자열로 된 사용자용 프롬프트입니다.

데모: 'Hello, 이름'



1. **helloworld.py**를 엽니다.
2. 사용자의 입력을 가져옵니다.
3. 응답을 출력합니다.
4. **hello <response>**가 출력되도록 입력을 연결할 수 있습니까?
5. 프로그램을 다시 실행합니다.

학습 내용 확인



Python의 루프 두 가지는 무엇입니까?

값 목록에서 특정 요소를 어떻게 선택합니까?

elif는 무엇을 의미합니까?

요점



- if 조건 스테이트먼트는 조건을 평가한 후 true이면 코드 블록을 실행합니다.
 - elif와 else 스테이트먼트는 if 스테이트먼트 다음에 나옵니다.
- while 루프는 특정 조건이 false를 반환할 때까지 코드 블록을 실행합니다.
- 목록은 특정 순서가 지정된 값의 모음입니다. 값은 같은 유형이 아니어도 됩니다.
- 사전은 키-값 쌍의 모음으로, 키를 사용하여 값에 액세스할 수 있습니다.