

공학적 문제해결 기법

6주차 1차시

조건 걸기





6주차 1차시 : 조건걸기

I. 개요

예/아니오에 대한 질문과 응답

II. 학습 개요

1) 학습 목표

예/아니오에 대한 질문들은 조건문(condition)이라고 불리며, if문 안에 이러한 조건문과 응답을 알아본다. 또한 참 또는 거짓을 판단하는 연산자는 어떠한 것들이 있는지 학습한다.

2) 학습 목차(세부 목차)

- '예/아니오'에 대한 질문
- 참/거짓
- 여러 개의 조건걸기(IF - THEN - ELSE)



6주차 1차시 : 조건걸기

III. 학습 1 - '예/아니오'에 대한 질문

예/아니오에 대한 질문과 답

- ☒ 예/아니오에 대한 질문과 답 → 조건문
- ☒ if문 안에 조건문과 응답을 조합
- ☒ if문이 어떻게 사용되는지 알아보까요?



```
>>> age = 13
```

```
>>> if age > 20:
```

```
    print('You are too old!')
```

- ☒ 콜론(:) 다음 줄은 블록 안에 있어야 함
- ☒ 질문에 대한 대답이 '예' (조건이 내용이 참(true))라면 그 블록에 있는 명령이 실행



6주차 1차시 : 조건걸기

III. 학습 1 - '예/아니오'에 대한 질문

코드 블록(block)

☒ 프로그래밍구문들의집합

☒ if age > 20: 이 참일 경우 “You are too old!”라는 말보다 더 많은 것들을 출력하고 싶다면

➡

```
>>> age = 25
>>> if age > 20:
    __print("You are too old!")
    __print("Why are you here?")
```

☒ age > 20이라는 조건이 참일 경우에만 실행되는 두 개의 print문으로 구성

☒ 위빈칸인 공백(whitespace)은 중요한 의미가 있음

☒ 동일한 위치(왼쪽에서 동일한 개수의 공백으로 들여쓴 위치)에 있는 코드는 하나의 블록으로 그룹



6주차 1차시 : 조건걸기

III. 학습 1 - '예/아니오'에 대한 질문

코드 블록(block)

- ✓ 이전에 있던 공백보다 더 많은 공백을 가지고 새로운 코드 줄을 시작하는 것은 이전 블록에 속한 새로운 블록을 시작
- ✓ 블록에 있는 구문들을 그룹지어야 함
- ✓ 구문들은 서로 관련이 있기 때문에 모두 같이 차례차례 실행





6주차 1차시 : 조건걸기

III. 학습 1 - '예/아니오'에 대한 질문

코드 블록(block)

- ✓ 들여쓰기 간격을 바꾸면 새로운 블록을 생성

➡

```
>>> age = 25
>>> if age > 20:
    ____print('You are too old!')
    _____print('Why are you here? ')
```

- ✓ 네 칸의 공백으로 처음 시작했다면, 그 블록 내에서는 계속해서 네 칸의 공백을 사용해야 함
- ✓ 따라서 이 코드를 실행하면 문제가 있는 곳을 빨간색 블록으로 강조하면서
SyntaxError 메시지를 보여줌
- ✓ 공백을 일관되게 유지하면 코드를 훨씬 더 읽기 쉽게 해줌 가독성



6주차 1차시 : 조건걸기

Ⅲ. 학습 2 - 참 / 거짓

참(True)과 거짓(False)

- ☒ 조건문(condition)은 어떤 것을 비교하여 그 조건에 맞는지(참(True)) 맞지 않은지(거짓(False))를 알려줌
- ☒ age > 10은 조건문
- ☒ 조건문을 생성하기 위해서는 연산자(operator)라고 불리는 기호를 사용
- ☒ 조건문에 사용되는 관계연산자
- ☒ 대소를 비교하는 연산을 관계 연산이라 함
- ☒ 관계연산자표



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

참(True)과 거짓(False)

기 호	정 의
==	~와같다 (= 과 혼동하지말것!)
!=	~와같지않다.
>	~보다크다.
<	~보다작다.
>=	~보다크거나같다.
<=	~보다작거나같다.



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

관계연산

➡ `>>> 6 == 9`

`False`

`>>> 6 != 9`

`True`

`>>> 1 > 3`

`False`

`>>> 4 <= 5`

`True`

`>>> a=5`

`>>> b=10`

`>>> a < b`

`True`



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

관계연산

- ☑ 다음과 같은 복합적인 관계식도 지원

➡ `>>>0<a<b`
True

- ☑ 대부분의 다른 언어에서는 위 식을 다음과 같이 풀어서 써야함

➡ `>>>0<a and a<b`
True

- ☑ and 연산자 · 논리연산자 중 하나

- ☑ 논리연산자는 진위(참 또는 거짓)를 판별하는 논리적 연산에 사용되며, 그 결과 값은 참 또는 거짓으로 표현



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

관계연산

기 호	사 용 예
not	not x
and	x and y
or	x or y



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

논리연산

☒ 논리 연산자들은 여러 개의 조건을 결합하는 용도로 사용



```
>>> a=5
```

```
>>> b=10
```

```
>>> not a
```

```
False
```

```
>>> 0 < a and a < b
```

```
True
```

```
>>> 0 < a or a > b
```

```
True
```



6주차 1차시 : 조건걸기

III. 학습 2 - 참 / 거짓

몇 가지 예 살펴보기

☒ 실행결과가 어떻게 될까요?



```
>>> age = 10
```

```
>>> if age > 10 :
```

```
    print('You are too old for my jokes!')
```

```
>>> age = 20
```

```
>>> if age >= 10 :
```

```
    print('You are too old for my jokes!')
```

```
>>> age = 10
```

```
>>> if age == 10 :
```

```
    print('You are young!!')
```



6주차 1차시 : 조건걸기

Ⅲ. 학습 3 - 여러 개의 조건걸기(IF - THEN - ELSE)

IF - THEN - ELSE

- ☒ 어떤 조건이 True일 경우에 어떤 작업을 하도록 하는 것 외에도, 어떤 조건이 참이 아닐 경우 (False)에 어떤 작업을 하도록 사용할 수 있다.
- ☒ 이때 사용하는 기술이 if-then-else문
- ☒ 만약에 조건문이 참이면 이것을 실행하고, 그렇지 않으면 저것을 실행

```
>>> print("input your score?")
>>> score = 70
>>> if (score >= 60) :
    print("congratulations!!!")
else :
    print("fail")
```



6주차 1차시 : 조건걸기

Ⅲ. 학습 3 - 여러 개의 조건걸기(IF - THEN - ELSE)

IF - THEN - ELSE

 if블록내의문이 1개만 존재할때 다음과 같이 한줄에 붙여 쓸 수 있음

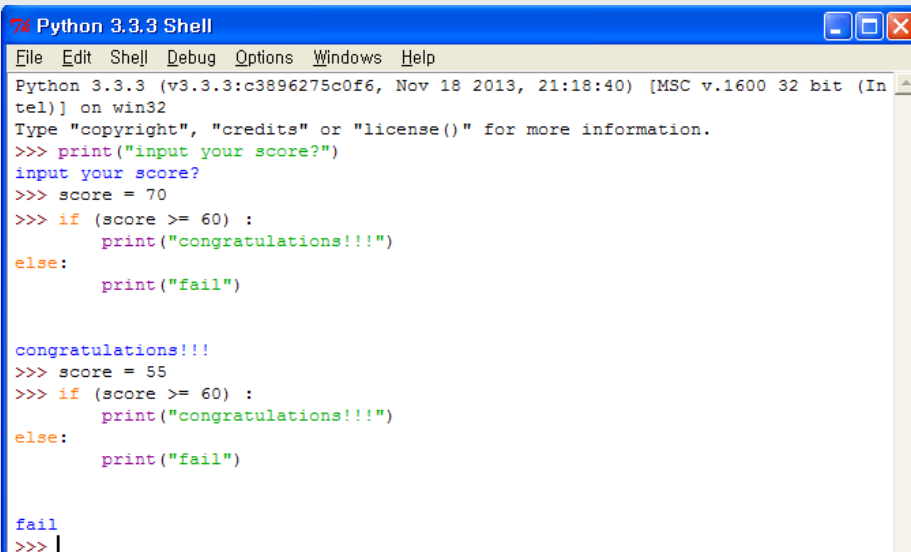


```
>>> print("input your score?")
```

```
>>> score = 55
```

```
>>> if (score >= 60) : print("congratulations!!!")
```

```
else : print("fail")
```



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("input your score?")
input your score?
>>> score = 70
>>> if (score >= 60) :
    print("congratulations!!!")
else:
    print("fail")

congratulations!!!
>>> score = 55
>>> if (score >= 60) :
    print("congratulations!!!")
else:
    print("fail")

fail
>>> |
```




『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』

