

Python as a Hobby

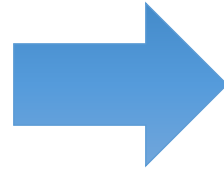
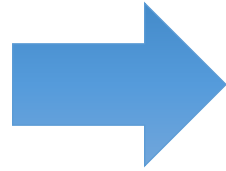
Condition

지금까지 배운걸 살펴보자

- 프로그램?
- 데이터
 - 숫자
 - 문자열
- 데이터를 담는 형태
 - 변수
 - 리스트
 - 튜플
 - 딕셔너리

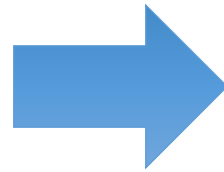
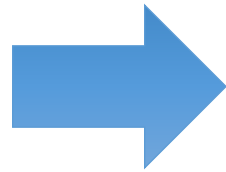
Program

- 컴퓨터에게 무언가를 시키는 것



Program

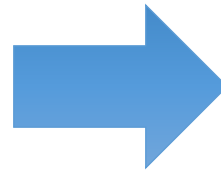
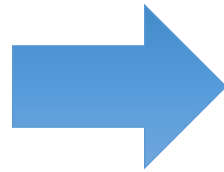
- 특정한 무언가를 시키는 것



“익태에게 5000 원 보내라”

Data in Program

- 특정한 무언가를 시키는 것

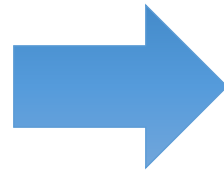
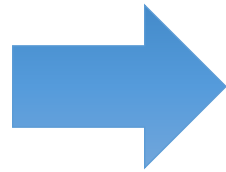


“익태에게 5000원 보내라”

문자열 데이터 숫자형 데이터 함수

Loop in Program

- 특정한 무언가를 순차적으로 시키는 것



“익태에게 5000 원 보내라”

“아름이에게 5000 원 보내라”

“승주에게 5000 원 보내라”

⋮

Loop in Program

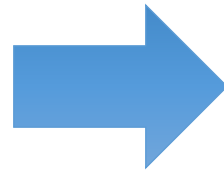
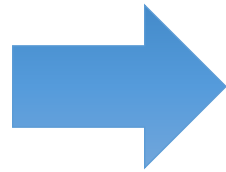
- 특정한 무언가를 순차적으로 시키는 것



“익태에게 5000원 보내라”
“아름이에게 5000원 보내라”
“승주에게 5000원 보내라”
⋮

Loop in Program

- 특정한 무언가를 반복해서 순차적으로 시키는 것



매달 1 일에

“익태에게 5000 원 보내라”

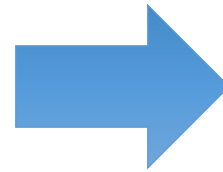
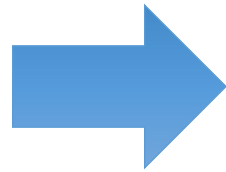
“아름이에게 5000 원 보내라”

“승주에게 5000 원 보내라”



Condition in Program

- 특정한 무언가를 반복해서 순차적으로 시키는 것



조건문

if
elif

매달 1 일에

“익태에게 5000 원 보내라”
“아름이에게 5000 원 보내라”
“승주에게 5000 원 보내라”

반복문

for
while

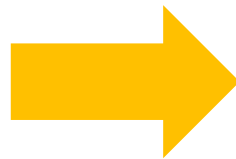


조건문 (Conditional statement)

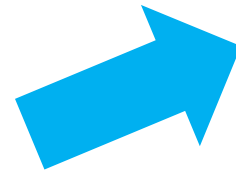
- 무언가를 판단/결정/선택 할 때 사용

조건문 (Conditional statement)

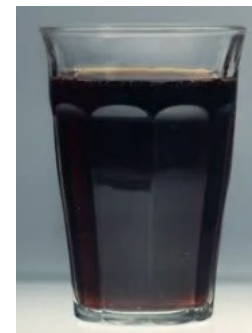
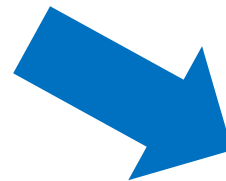
- 무언가를 판단/결정/선택 할 때 사용



얼음이 있느냐?
얼음이 없느냐?



맛집

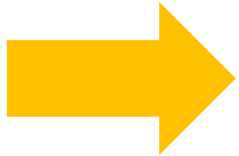


맛집 아님

조건문 (Conditional statement)

- 무언가를 판단/결정/선택 할 때 사용

소개남



답음



안답음



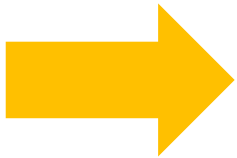
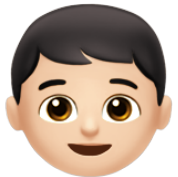
조건문 (Conditional statement)

- 프로그램은 순차적으로 진행

조건문 (Conditional statement)

- 프로그램은 순차적으로 진행

소개남



답음



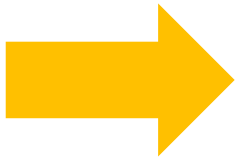
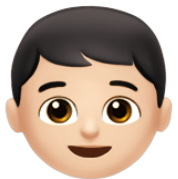
안답음



조건문 (Conditional statement)

- 프로그램은 순차적으로 진행

소개남



답음



연하



연상



안답음



조건문 (Conditional statement)

- 분기점을 만들어서 프로그램의 진행 방향을 변경

조건문 (Conditional statement)

- 분기점을 만들어서 프로그램의 진행 방향을 변경

1. 소개팅이 들어왔음

2. 소개팅남의 사진을 받았음

3. 소개팅남이 이상이를 닮았음

4. gogo

1. 소개팅이 들어왔음

2. 소개팅남의 사진을 받았음

3. 소개팅남이 이상이를 안닮음

4. byebye

조건문 (Conditional statement)

- 분기점을 만들어서 프로그램의 진행 방향을 변경

1. 소개팅이 들어왔음

2. 소개팅남의 사진을 받았음

3. 소개팅남이 이상이를 닮았음

4. gogo

1. 소개팅이 들어왔음

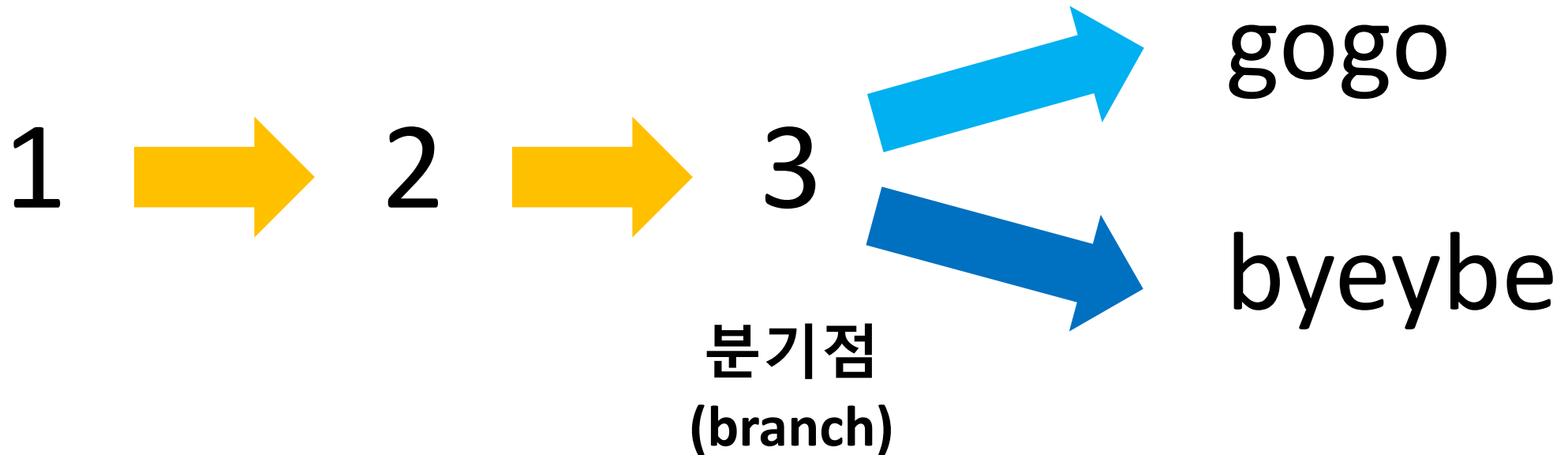
2. 소개팅남의 사진을 받았음

3. 소개팅남이 이상이를 안닮음

4. byebye

조건문 (Conditional statement)

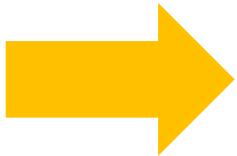
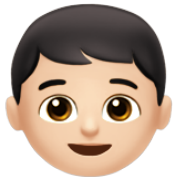
- 프로그램은 순차적으로 진행



조건문 (Conditional statement)

- 사용자의 반응(입력값)에 상호작용하기 위해서는 판단을 해야함

소개남



답음



연하



연상



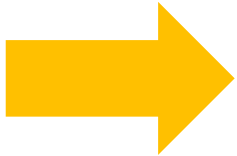
안답음



조건문 (Conditional statement)

- 사용자의 반응(입력값)에 상호작용하기 위해서는 판단을 해야함

소개녀



답음



연하



연상



안답음



조건문 (Conditional statement)

- 무언가를 **판단/결정/선택** 할 때 사용
- 프로그램은 순차적으로 진행
- 분기점을 만들어서 프로그램의 진행 방향을 변경
- 사용자의 반응(입력값)에 상호작용하기 위해서는 판단을 해야함

조건문을 어떻게 쓰느냐

- if
- if ~ else
- if ~ elif ~ else

if

- 판단 / 결정 / 선택이 필요한 곳에 조건문(if)을 넣음

if 조건:

실행할 문장1

실행할 문장2

...

if

- 조건의 결과는 참(true) 또는 거짓(false)으로 판단함
- if 조건을 쓰고 바로 뒤에 ':'을 붙여야 함

if 조건:

실행할 문장1

실행할 문장2

...

비교연산자

- 조건이란

- 조건에 만족한다
- 조건에 만족하지 않는다
- 두가지 밖에 없음

- 비교연산이란

- 어떤 두 가지를 비교
- 비교연산자로 이 연산이 참인지 거짓인지 판단할 수 있음

비교연산자

- 조건문에서 사용하는 비교연산자

<, >, ==, !=, >=, <=

비교연산자

대상1 연산자 대상2

대상1 < 대상2

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x >= y$	x가 y보다 크거나 같다
$x <= y$	x가 y보다 작거나 같다

비교연산자

- 해보자

x = 3

y = 2

x > y

True

taekim = 32

arming = 31

taekim > arming

taekim == arming

taekim < arming

비교연산자

- 또다른 비교연산자

in, not in

비교연산자

- 또다른 비교연산자

무언가 in 리스트

$x \text{ in } s$

비교연산자

- 또다른 비교연산자

soon = ['itk', 'arm', 'sjh']

'khc' in soon

비교연산자

- 또다른 비교연산자

soon = ['itk', 'arm', 'sjh']

'khc' **not in** soon

비교연산자

- 또다른 비교연산자

```
soon = ['itk', 'arm', 'sjh']
```

```
if 'khc' not in soon:
```

```
    soon.append('khc')
```

if

- 들여쓰기로 if 블록을 구분
- 같은 if문 안에 있다면 동일한 개수의 들여쓰기를 사용

if 조건:

스페이스 4칸
또는 8칸
또는 탭

■ 실행할 문장1
■ 실행할 문장2
■ ...

들여쓰기(indentation)

- 파이썬에서는 들여쓰기로 코드의 구역을 구분함
- 들여쓰기를 하는 방법
 - 스페이스 4칸
 - 스페이스 8칸
 - 탭(tab)

들여쓰기(indentation)

- 주의사항1

- 스페이스와 탭을 섞어서 사용할 수 없음
- 스페이스 4칸과 스페이스 8칸도 같이 사용할 수 없음
- 한가지만 사용해야함
- 탭 또는 스페이스4칸 중에 하나를 쓰자

들여쓰기(indentation)

- 주의사항2

- 동일한 구역이라면 (= 동일한 if문 블록 안에 있다면) 동일한 개수의 들여쓰기를 해야함

if 조건문:

수행할 문장1

수행할 문장2

┌─┐ 수행할 문장3

if 조건문:

수행할 문장1

수행할 문장2

┌─┐ ┌─┐ 수행할 문장3

if 조건문:

┌─┐ 수행할 문장1

수행할 문장2

┌─┐ 수행할 문장3

if

- 들여쓰기로 if 블록을 구분

if 조건:

```
■ 실행할 문장1  
■ 실행할 문장2  
■ ...
```

여기 if에 속한 애들

if

if 조건:

실행할 문장1

실행할 문장2

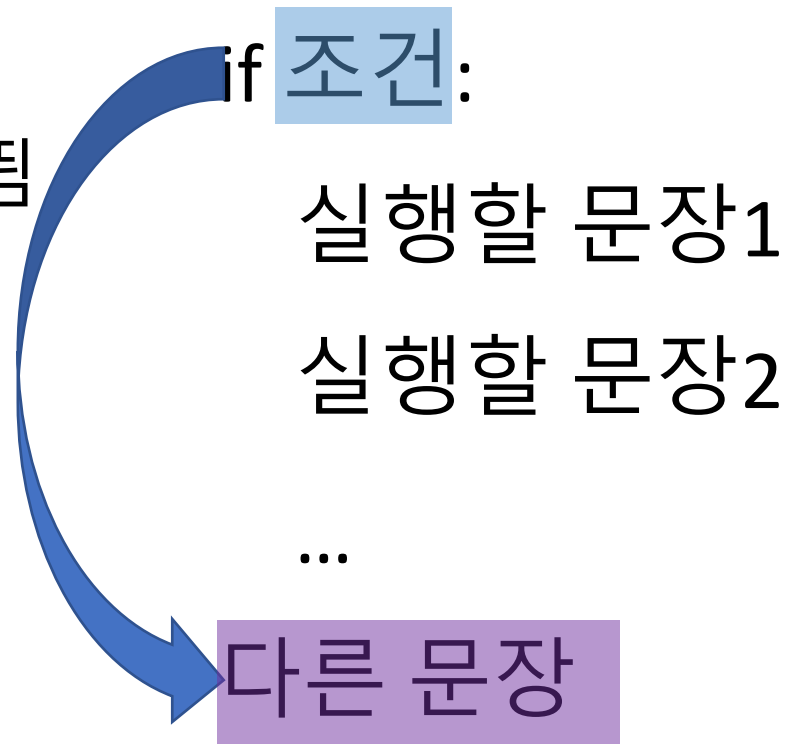
...



조건이 True 이면
아래 문장을 실행

if

조건이 False 이면
아래 문장을 건너뛴다



if

```
sogaeting_nam = '이상이'

if sogaeting_nam == '이상이':
    print("let's do this")

print("outside of if statement")
```

if 조건:

실행할 문장1
실행할 문장2
...

다른 문장



조건이 True 이면
아래 문장을 실행

if

- 해보자
 - 아래 두 변수를 사용하여 taekim이 더 크면 “taekim is older than me”
 - 더 작으면 “arming is older than me”
 - 를 출력해보자

taekim = 32

arming = 31

if

- 해보자

- soon 리스트의 원소 개수를 세고 만약 5명 이하면
- 'cjh'을 추가하는 코드를 작성해보자
- 리스트의 함수를 활용해보자

```
soon = ['itk', 'arm', 'sjh', 'khc']
```

if ~ else

- 조건이 false일 경우 실행할 부분

```
1  sogaeeting_nam = '이상이'
2
3  if sogaeeting_nam == '이상이':
4      print("let's do this")
5  else:
6      print("show me another man")
7
8  print("outside of if statement")
```

if ~ else

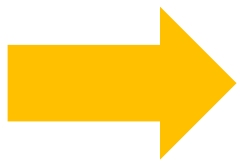
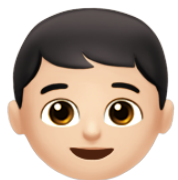
- 조건이 false일 경우 실행할 부분
- Else가 있는 것과 없는 것의 차이는 무엇일까?

if ~ else

- 조건이 false일 경우 실행할 부분
- Else가 있는 것과 없는 것의 차이는 무엇일까?
 - Else 안에 새로운 코드 실행가능
 - 또다른 조건문을 추가할 수 있음
 - 조건이 false일 때만 해야하는 일들을 실행
 - 에러처리

if ~ else

소개남



답음



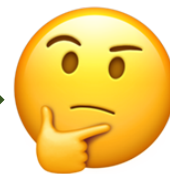
안답음



연하



연상

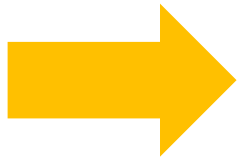
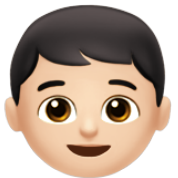


if ~ else

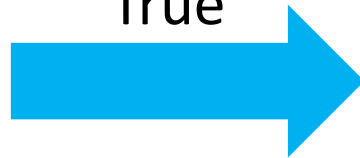
조건1

조건2

sogaeting_nam



True



False



True



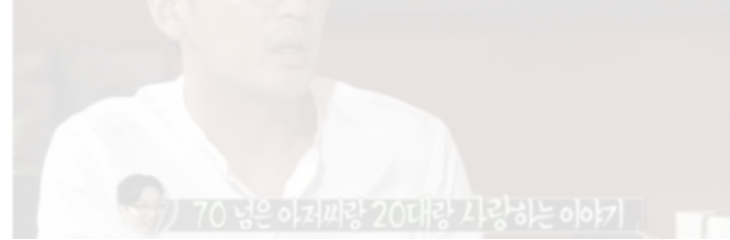
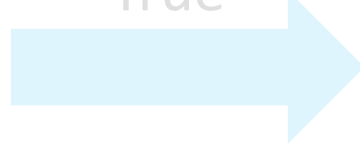
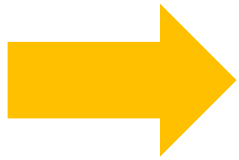
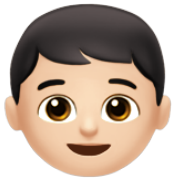
False



if ~ else

```
sogaeting_nam = '이상이'  
sogaeting_nam_age = 40
```

sogaeting_nam



조건2

True



False



if ~ else

조건2

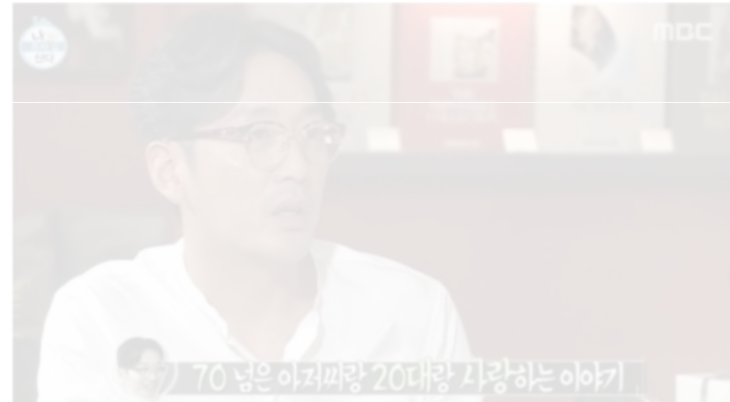
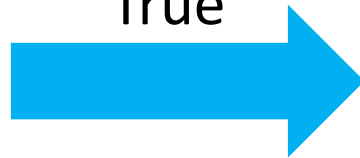
조건1

```
if sogaeeting_nam == '이상이':
```

sogaeting_nam



True



True



False



False



```
else:
```

```
| print("byebye")
```

if ~ else

조건1

sogaeting_nam



True

False

```
if sogaeeting_nam == '이상이':  
    if sogaeeting_nam_age <= 31:
```



True

False



```
else:  
    print("byebye")
```

if ~ else

조건2

```
sogaeting_nam = '이상이'
sogaeting_nam_age = 40

if sogaeting_nam == '이상이':
    if sogaeting_nam_age <= 31:
        print("let's do this")
    else:
        print("byebye")
else:
    print("byebye")
```

sogaeting_nam



True



False



if ~ elif ~ else

- 여러개의 조건을 판단할 때 사용
- 다중 if가 필요할 때 elif로 구현

```
If <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
...  
else:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...
```


if ~ elif ~ else

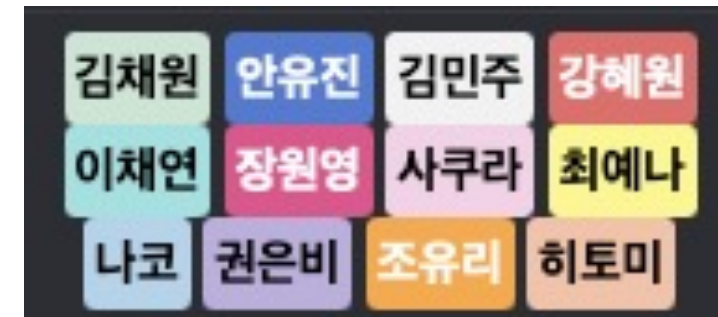
```
>>> pocket = ['paper', 'handphone']
>>> card = True
>>> if 'money' in pocket:
...     print("택시를 타고가라")
... else:
...     if card:
...         print("택시를 타고가라")
...     else:
...         print("걸어가라")
...
택시를 타고가라
>>>
```

```
>>> pocket = ['paper', 'cellphone']
>>> card = True
>>> if 'money' in pocket:
...     print("택시를 타고가라")
... elif card:
...     print("택시를 타고가라")
... else:
...     print("걸어가라")
...
택시를 타고가라
```

if ~ elif ~ else

- 해보자

- fan1은 아이즈원의 '김민주', '장원영'을 좋아한다
- fan2는 '장원영', '안유진'을 좋아한다
- fan3은 '권은비', '김채원'을 좋아한다
- 각 팬을 리스트로 만들고
- 리스트 안에는 팬이 좋아하는 멤버를 넣는다
- 팬 리스트 안에 있는 데이터에 따라 좋아하는 멤버이면 ':'
- 아니면 'T.T'를 출력하도록 코드를 작성해보자



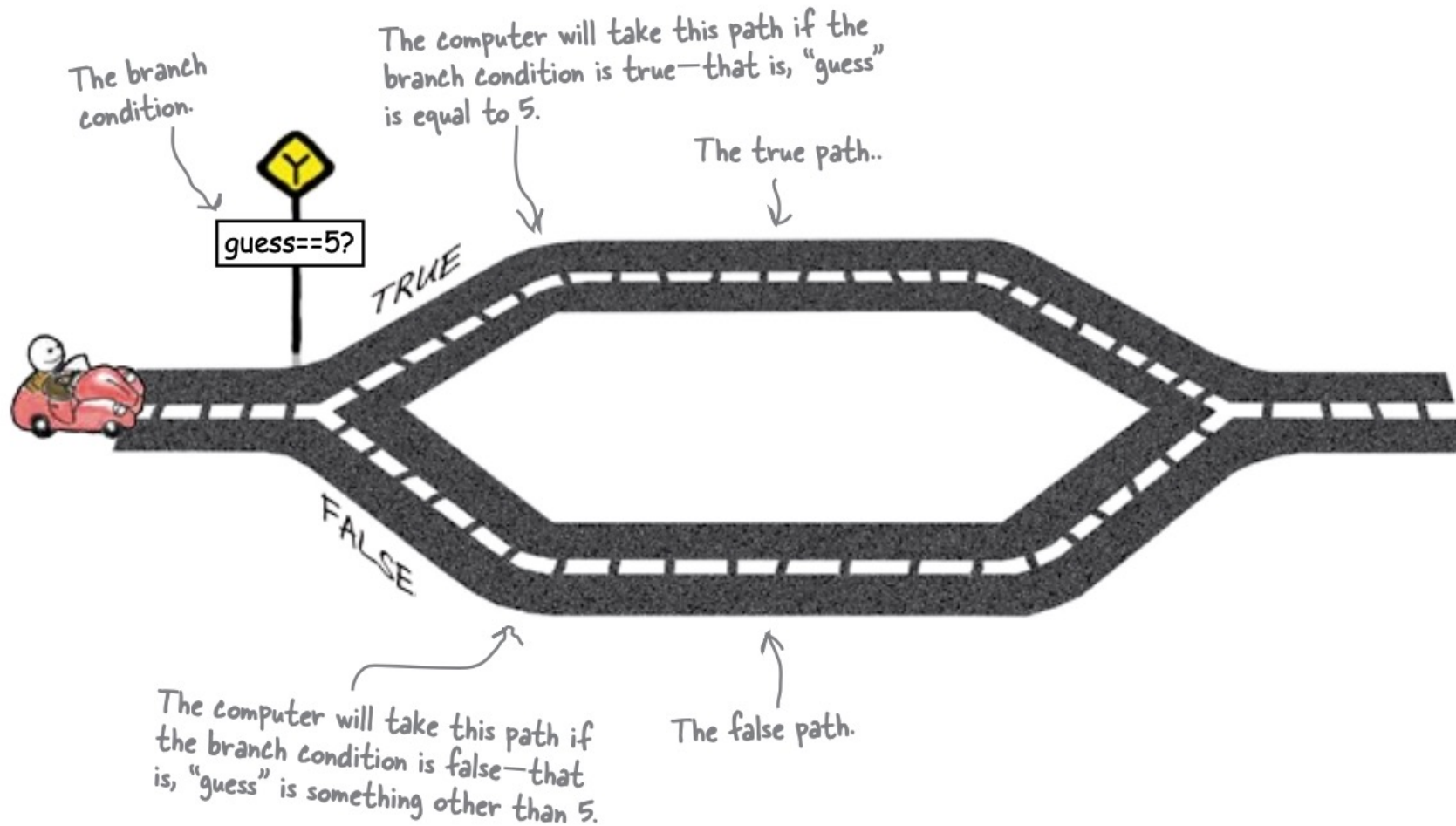
김채원	안유진	김민주	강혜원
이채연	장원영	사쿠라	최예나
나코	권은비	조유리	히토미

pass

- If문이 참이더라도 그냥 통과시키고 싶을 때 사용

```
>>> pocket = ['paper', 'money', 'cellphone']
>>> if 'money' in pocket:
...     pass
... else:
...     print("카드를 꺼내라")
...
```

코드의 흐름



코드의 흐름

