

# 왜 프로그램을 할까?

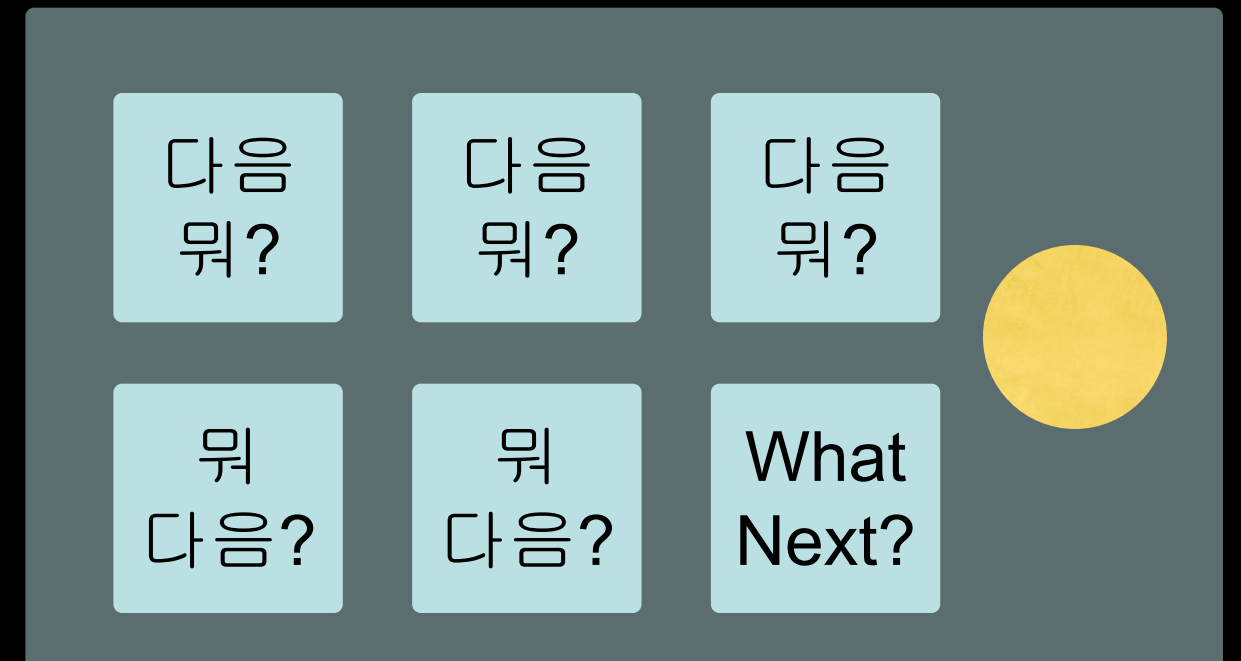
## 제1장

모두를 위한 파이썬  
[www.py4e.com](http://www.py4e.com)



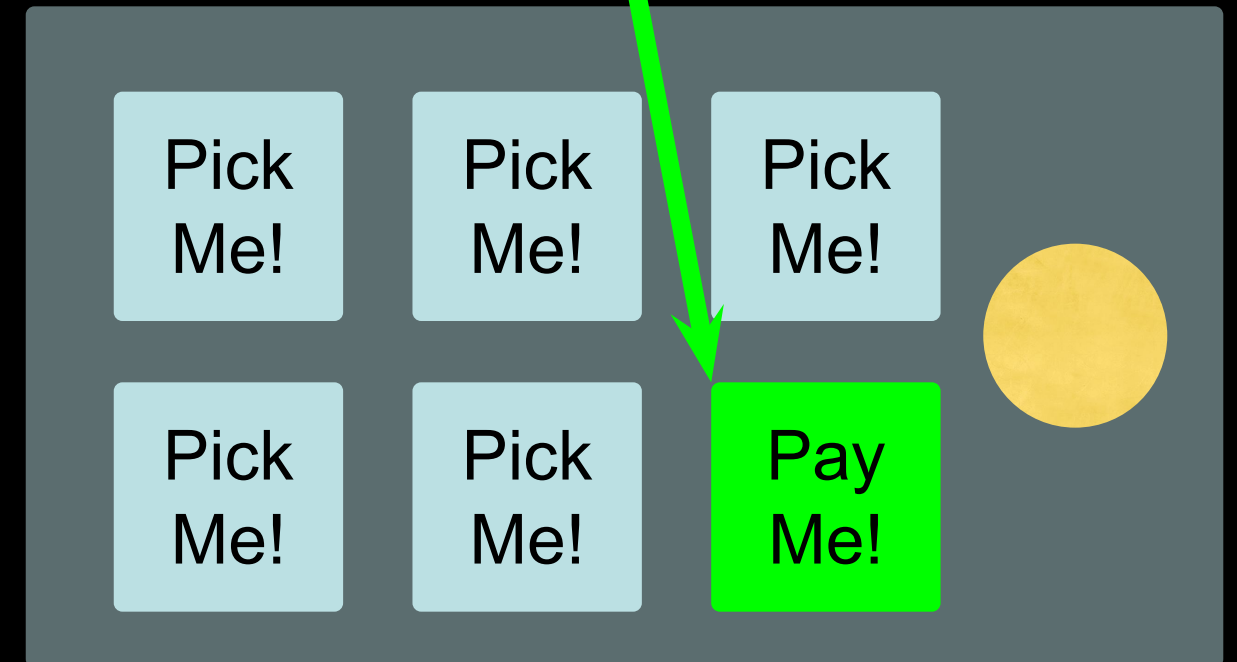
# 컴퓨터는...

- 컴퓨터는 사람을 위해 일하기 위한 목적으로 만들어짐
- 원하는 일을 시키려면 컴퓨터 언어를 알아야 함
- 여러 프로그램(명령)을 컴퓨터에 저장해 놓고 원하는 것을 때에 따라 골라 실행



# 프로그래머는...

- 아이폰 애플리케이션은 큰 시장
- 아이폰 애플리케이션은 30억이 넘는 다운로드
- 직장을 그만두고 아이폰 개발자가 되는 프로그래머가 있음
- 프로그래머는 프로그램 방법을 앎

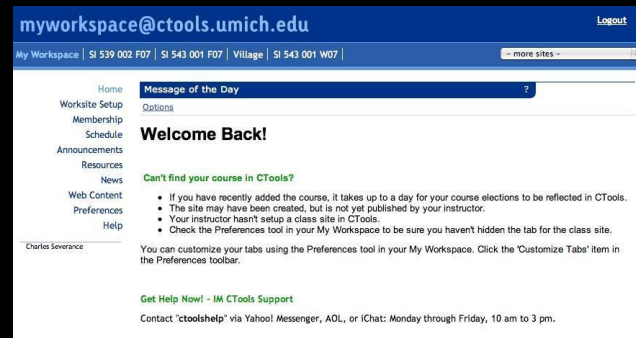


# 사용자 vs. 프로그래머

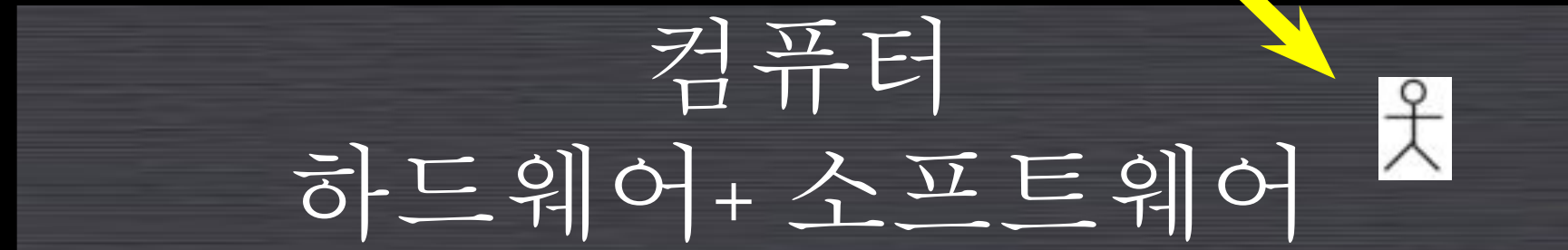
- 사용자는 컴퓨터를 도구로 사용한다 - 워드, 스프레드시트, 지도 등
- 프로그래머는 컴퓨터의 작동 방식과 언어를 학습
- 프로그래머는 도구를 사용해 새로운 도구를 만듦
- 프로그래머는 많은 사용자를 위한 도구나 특정 일을 자동화하기 위한 “도우미”를 개발

# 프로그래머가 되는 이유

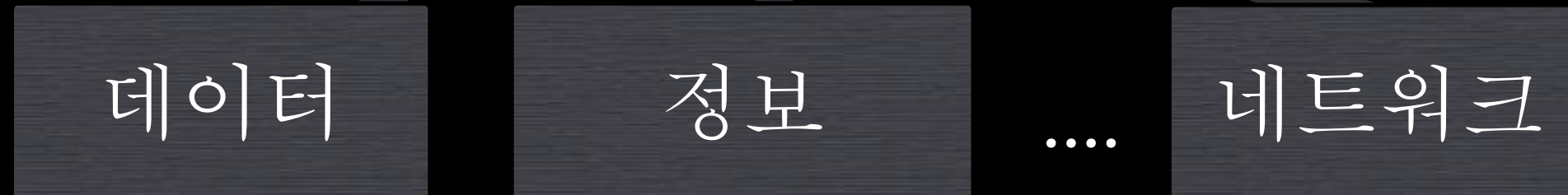
- 일을 하기위해 - 사용자나 프로그래머
  - 조사 자료 정리
- 다른 사람을 위해 - 프로그래밍 직업
  - Sakai 프로그램의 성능 문제 해결
  - 웹사이트에 방명록 추가



사용자



프로그래머



- 프로그래머는 사용자의 요구를 충족시킬 수 있는 소프트웨어를 개발
- 하드웨어와 소프트웨어를 다루며 데이터, 정보, 네트워크 문제를 해결

# 코드, 소프트웨어, 프로그램

- 명령문

- 컴퓨터의 작은 지능
- 사람의 시간과 에너지를 아끼기 위해서, 생각과 계산이 필요한 일들을 파이썬으로 코드화하고 실행

- 예술 - 사용자 입장

# 사람을 위한 프로그램



<https://www.youtube.com/watch?v=XiBYM6g8Tck>



# 사람을 위한 프로그램

while music is playing:

Left hand out and up

Right hand out and up

Flip Left hand

Flip Right hand

Left hand to right shoulder

Right hand to left shoulder

Left hand to back of head

Right hand to back of head

Left hand to right hip

Right hand to left hip

Left hand on left bottom

Right hand on right bottom

Wiggle

Wiggle

Jump



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

# 사람을 위한 프로그램

while music is playing:

Left hand out and up

Right hand out and up

Flip Left hand

Flip Right hand

Left hand to right shoulder

Right hand to left shoulder

Left hand to back of head

Right **ham** to back of head

Left hand to right **hit**

Right hand to left **hit**

Left hand on left bottom

Right hand on right bottom

Wiggle

Wiggle

Jump



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

# 사람을 위한 프로그램

while music is playing:

Left hand out and up

Right hand out and up

Flip Left hand

Flip Right hand

Left hand to right shoulder

Right hand to left shoulder

Left hand to back of head

Right **hand** to back of head

Left hand to right **hip**

Right hand to left **hip**

Left hand on left bottom

Right hand on right bottom

Wiggle

Wiggle

Jump



<https://www.youtube.com/watch?v=XiBYM6g8Tck>



# 사람을 위한 프로그램

the clown ran after the car and the car ran into the tent and  
the tent fell down on the clown and the car



Image: [https://www.flickr.com/photos/allan\\_harris/4908070612/](https://www.flickr.com/photos/allan_harris/4908070612/) Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)

# 파이썬을 위한 프로그램



Image: [https://www.flickr.com/photos/allan\\_harris/4908070612/](https://www.flickr.com/photos/allan_harris/4908070612/) Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)

```
name = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

python words.py  
Enter file: words.txt  
to 16

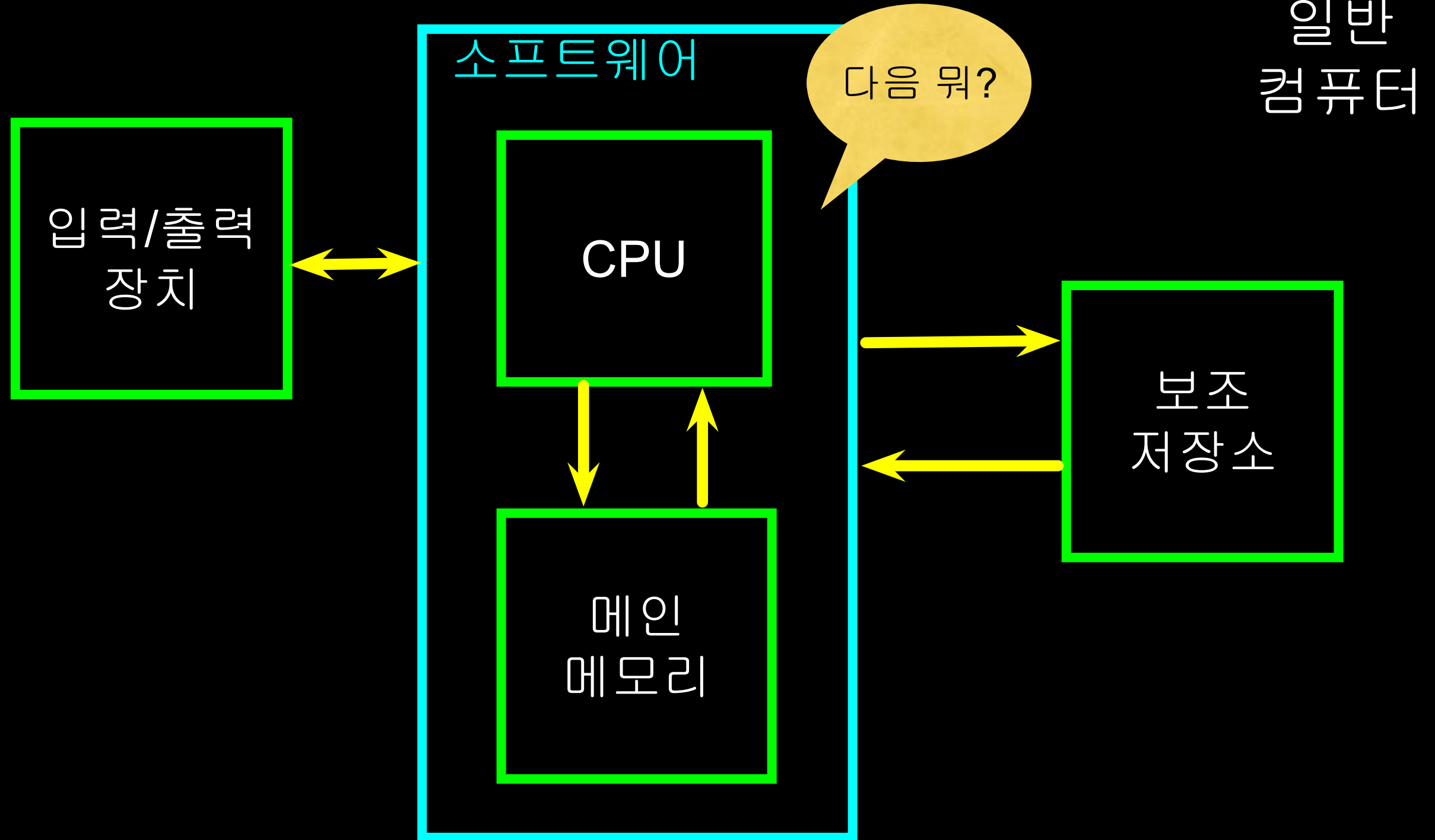
python words.py  
Enter file: clown.txt  
the 7

# 하드웨어 구조



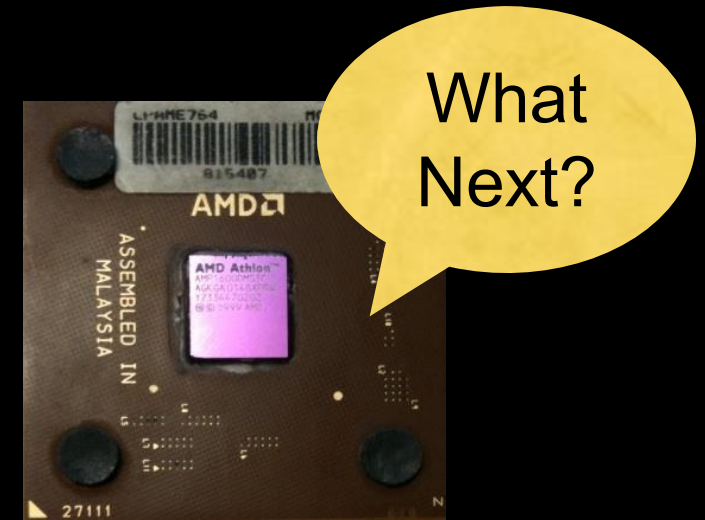
<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

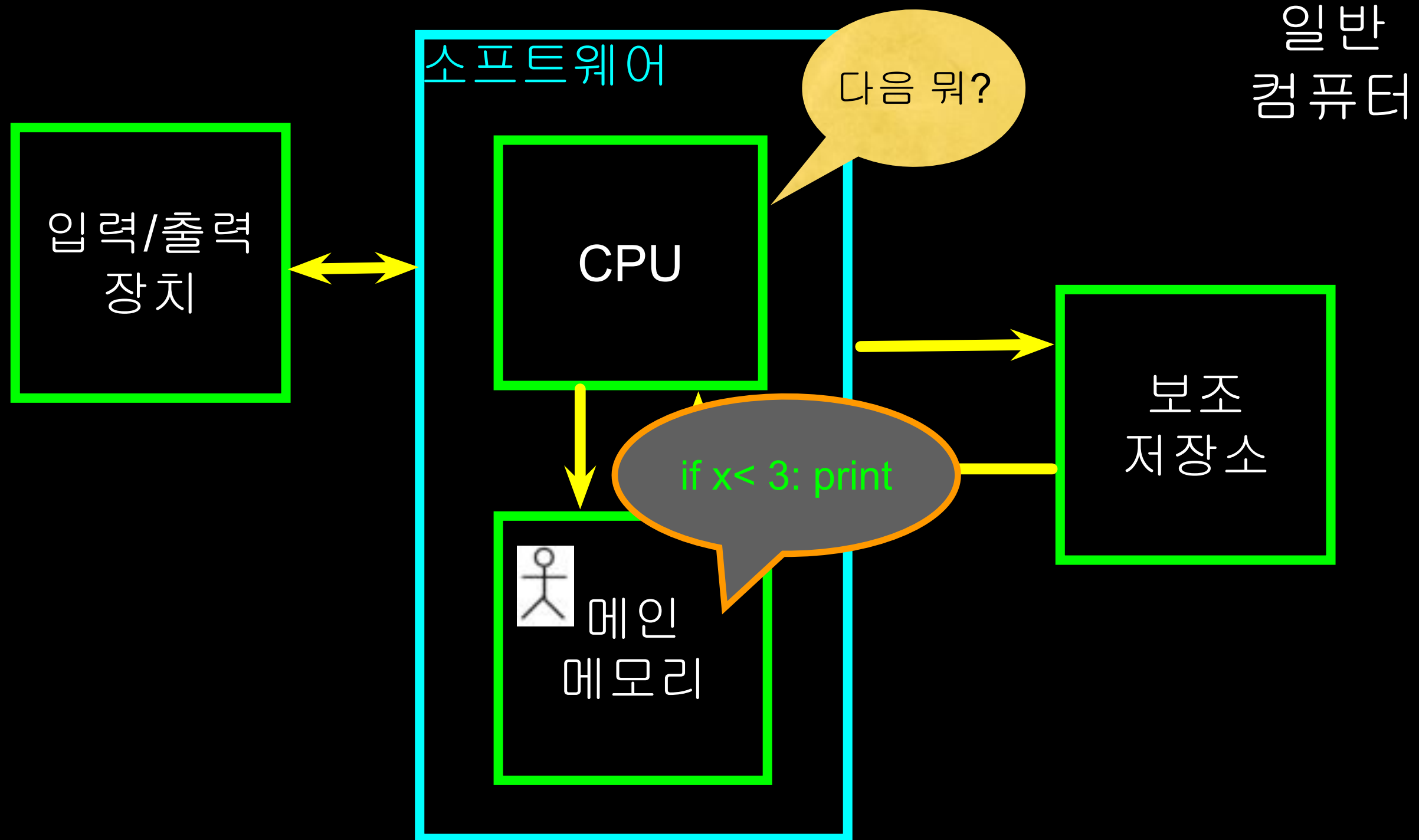


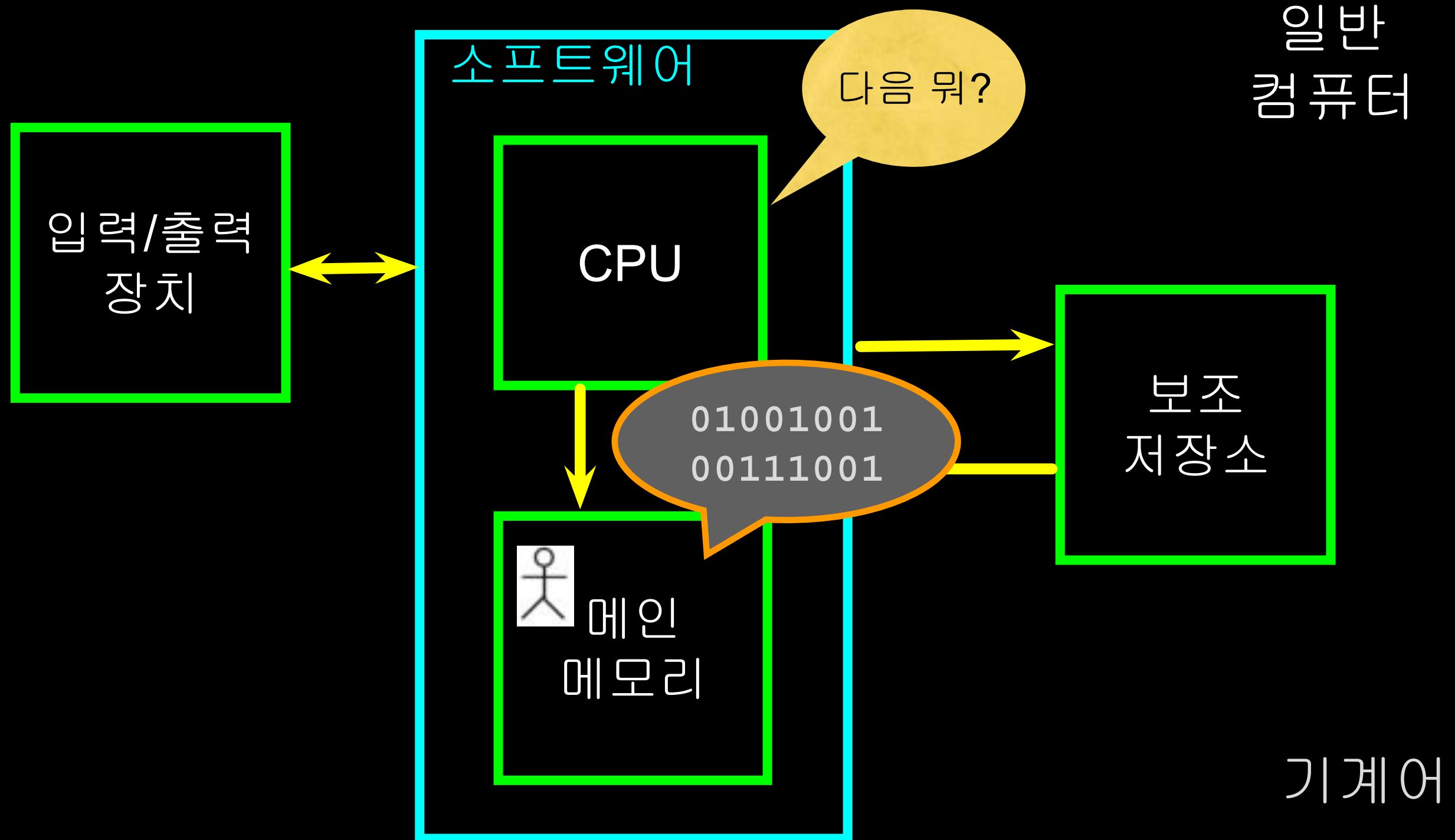


# 정의

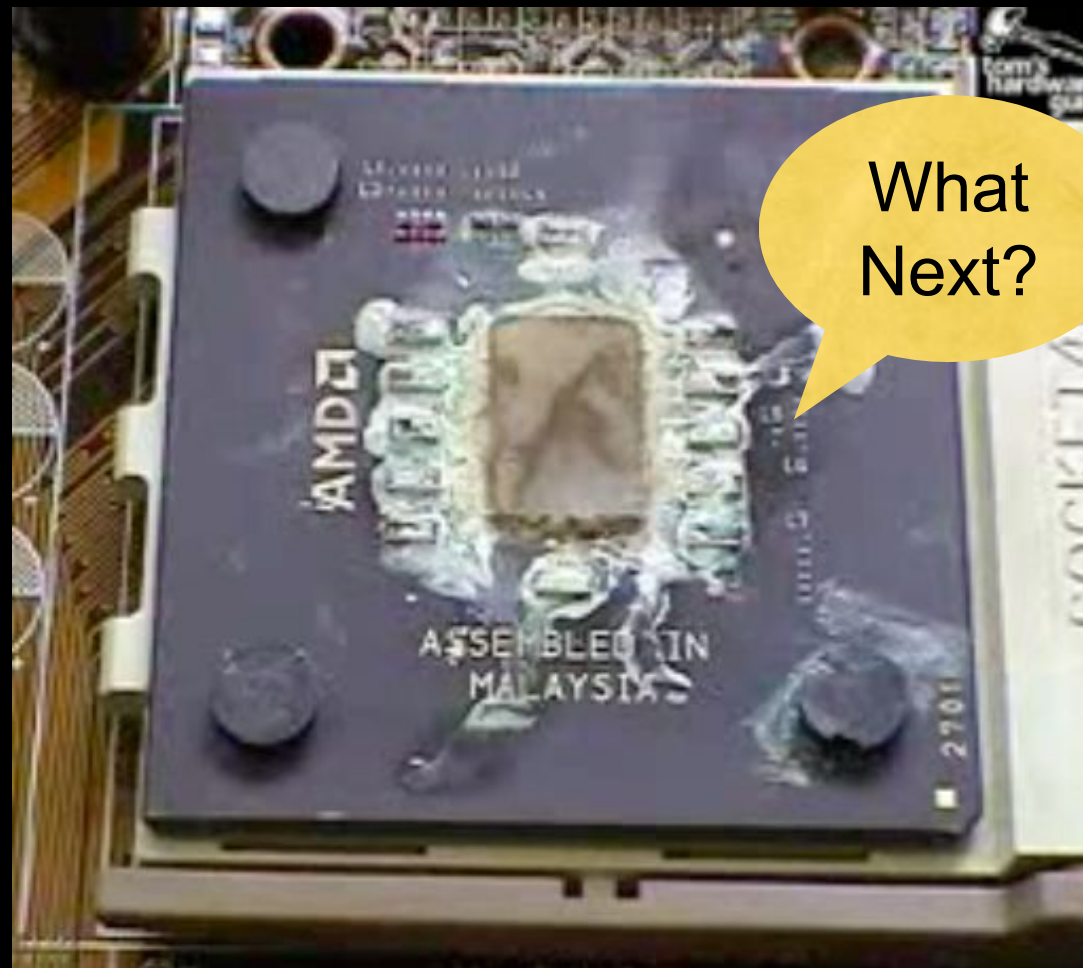
- **CPU:** 프로그램을 실행한다 - 다음에 무엇을 해야 할지 계속 묻는다. 컴퓨터의 뇌라고 보지 않는다 매우 빠르지만 똑똑하지 않다.
- **입력 장치:** 키보드, 마우스, 터치 화면
- **출력 장치:** 화면, 스피커, 프린터, DVD
- **메인 메모리:** 빠르고 작은 일시적 저장소 - 리부팅하면 지워진다
- **보조 저장소:** 느리고 작은 영구적 저장소 - 지워질 때까지 보존된다 - 디스크/ USB







# CPU



<http://www.youtube.com/watch?v=y39D4529FM4>

# 하드디스크



<http://www.youtube.com/watch?v=9eMWG3fwiEU>

파이썬 언어

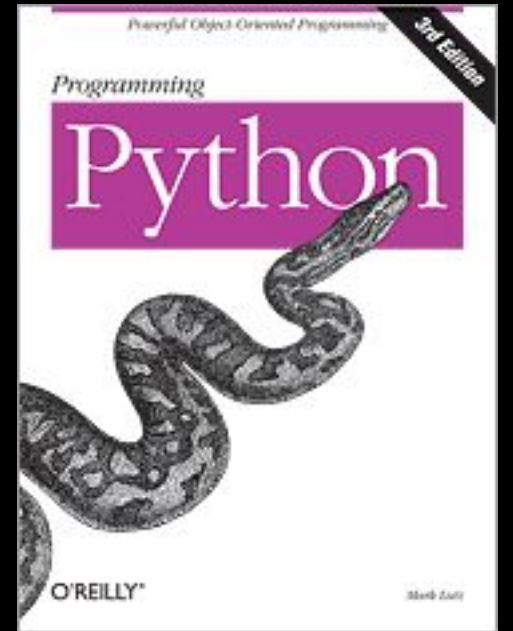
Parseltongue은 뱀의 언어. Parseltongue을  
하는 사람을 Parselmouth이라 부름.  
평범하지 않은 유전적 능력일 수 있다고 함.  
대부분의 Parselmouths은 Salazar Slytherin  
계통.



<http://harrypotter.wikia.com/wiki/Parseltongue>



Python은 파이썬 인터프리터의 언어.  
Python을 하는 사람을 Pythonista이라 부름.  
평범하지 않은 유전적 능력일 수 있다고 함.  
대부분의 Pythonistas는 Guido van Rossum가  
개발한 소프트웨어를 사용.



# 초급자: 문법 에러

- 파이썬 언어를 배워야지만 파이썬에게 명령을 내릴 수 있음. 처음에는 실수가 많은게 당연
- 실수를 하면 “문법 에러”가 발생. 감정적으로 받아들이지 마라
- 컴퓨터는 빠르지만 우리와 달리 배울 수 있는 능력이 없음. 따라서 우리가 파이썬을 배우는 것이 컴퓨터가 우리 언어를 배우는 것보다 빠름

파이썬과 대화하기


```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
```

```
"help", "copyright", "credits" or "license" for more information.
```

```
>>>
```



다음 뭐?

```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
```

```
"help", "copyright", "credits" or "license" for more information.
```

```
>>> x = 1
```

```
>>> print(x)
```

```
1
```

```
>>> x = x + 1
```

```
>>> print(x)
```

```
2
```

```
>>> exit()
```

파이썬이 올바르게 설치됐는지 확인하는 방법  
quit()로 세션을 끝낼 수 있음

어떻게 대화할까?

# 파이썬의 요소

- 단어 - 변수와 예약어 (챕터 2)
- 문장 구조 - 문법 (챕터 3-5)
- 이야기 구조 - 목적에 맞는 프로그램

```
name = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

파이썬에서  
파일에 있는 단어  
개수 세기

python words.py  
Enter file: words.txt  
to 16



# 예약어

예약어를 변수 이름/식별자로 사용할 수 없음

```
False  class  return  is  finally  
None  if  for  lambda  continue  
True  def  from  while  nonlocal  
and  del  global  not  with  
as  elif  try  or  yield  
assert  else  import  pass  
break  except  in  raise
```

# 문장/줄

x = 2	←	대입문
x = x + 2	←	대입문 + 표현식
print(x)	←	출력문

변수

연산자

상수

함수

프로그래밍 문단

# 파이썬 스크립트

- 대화식 파이썬은 3-4줄의 코드를 실험하기 좋음
- 하지만 대부분의 프로그램은 길기 때문에 파일에 코드를 적고 파이썬이 파일에 있는 명령을 실행
- 파이썬에게 스크립트를 주는 것과 같음
- 파일 이름에 “.py”를 붙여 파이썬을 썼다는 것을 표시

# 대화식 vs. 스크립트

- 대화식

- 파이썬에 한 줄을 입력하면 바로 반응

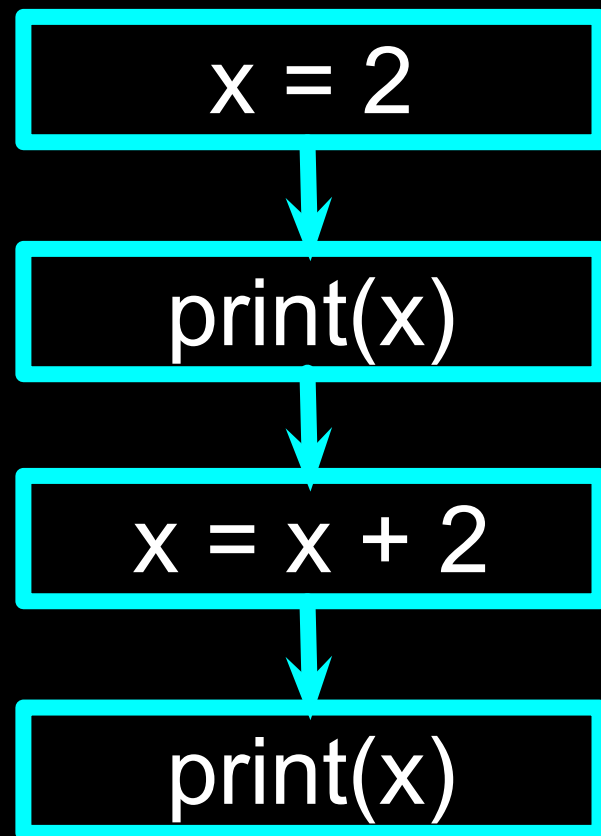
- 스크립트

- 텍스트 에디터를 사용해 파일에 명령을 적고 파이썬이 파일에 있는 명령을 실행

# 프로그램 단계/흐름

- 순차문 - 레시피나 설명문과 같이 진행 순서를 가짐
- 조건문 - 건너뛸 수 있음
- 반복문 - 반복돼 실행
- 프로그램에서 자주 반복될 부분을 따로 저장 (제4장)

# 순차문



프로그램:

```
x = 2  
print(x)  
x = x + 2  
print(x)
```

결과:

2  
4

프로그램은 순차적으로 진행  
프로그래머는 프로그램의 흐름을 결정

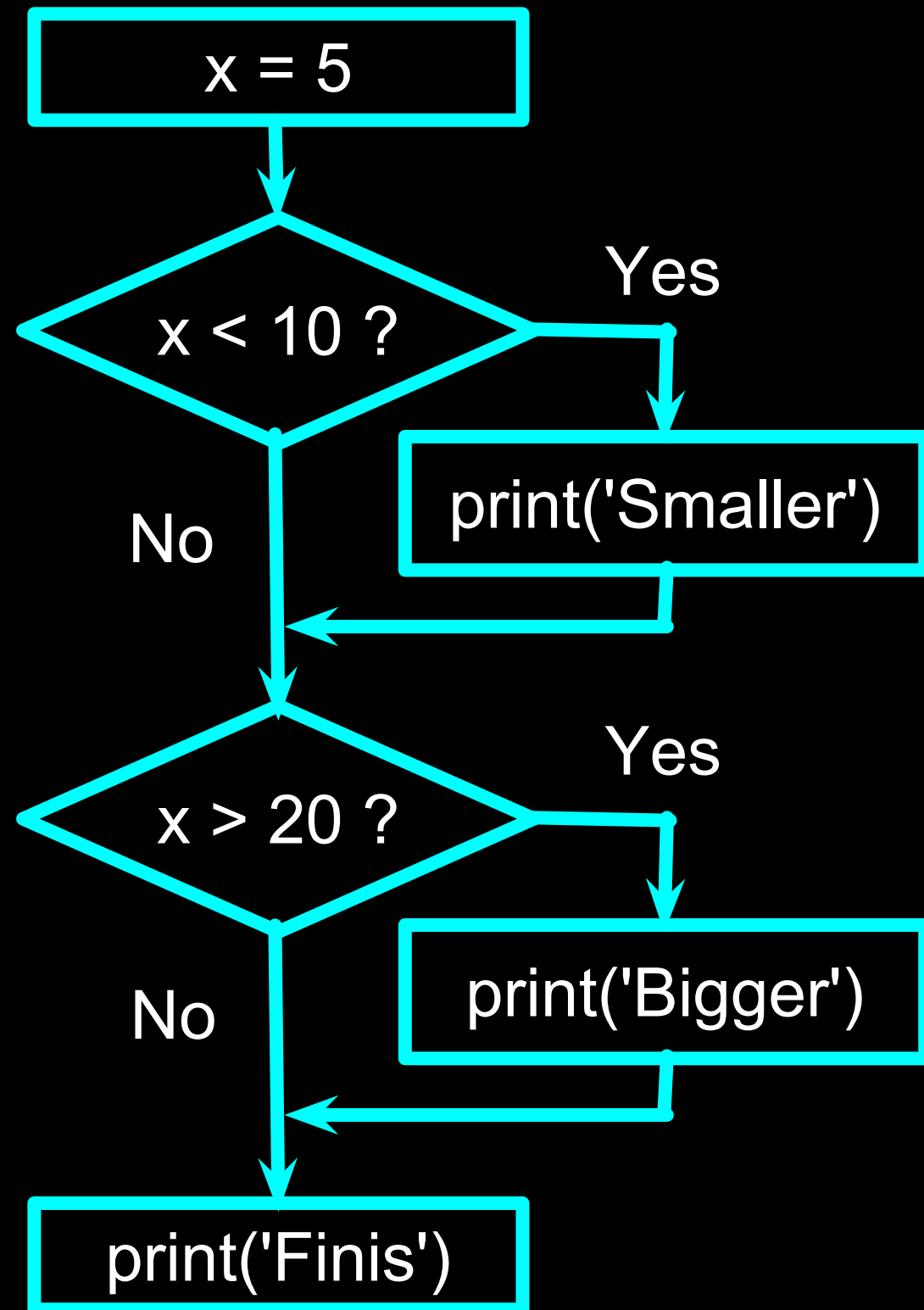
# 조건문

프로그램:

```
x = 5
if x < 10:
    print('Smaller')
if x > 20:
    print('Bigger')
print('Finis')
```

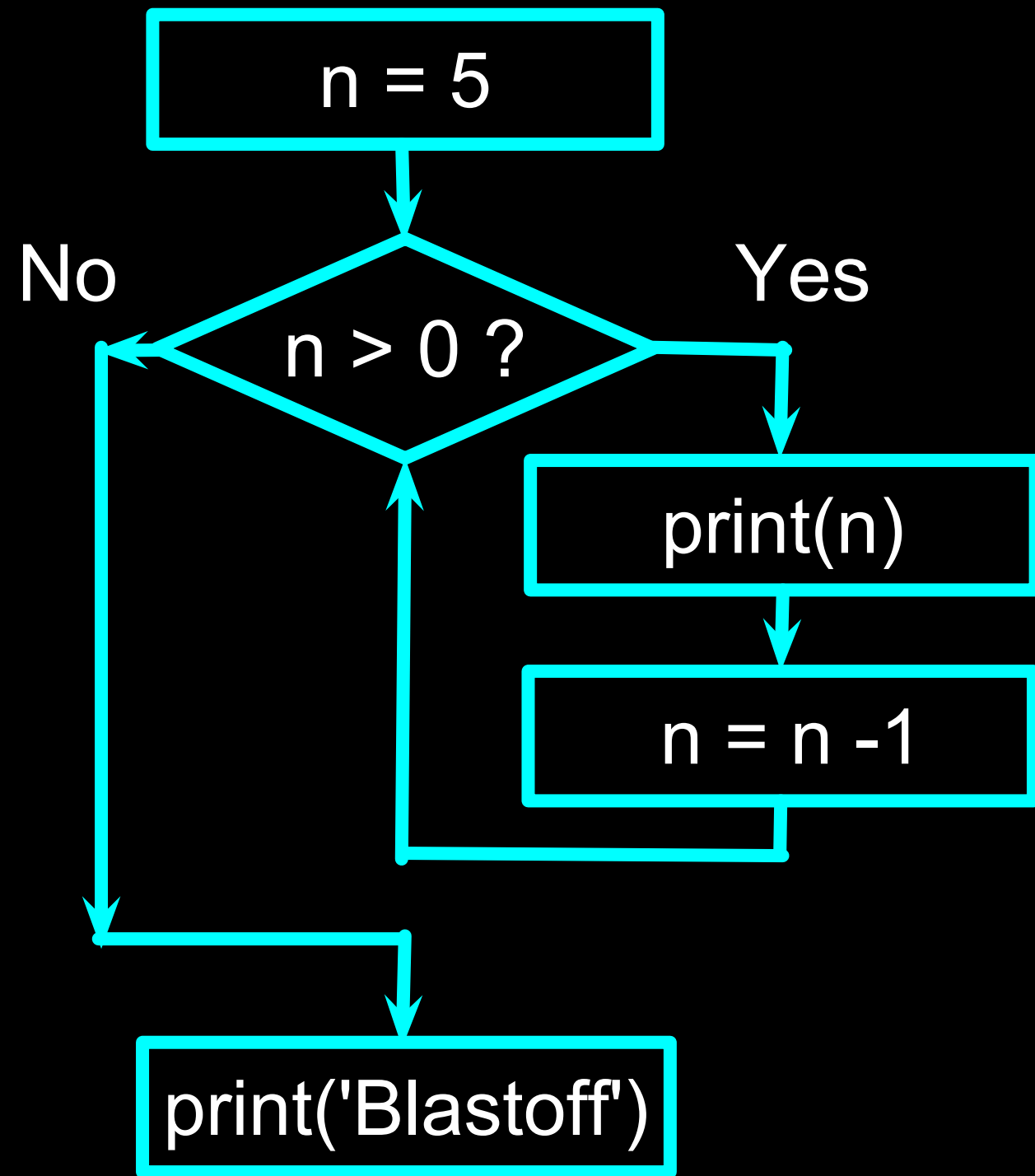
결과:

Smaller  
Finis





# 반복문



프로그램:

```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Blastoff!')
```

결과:

5  
4  
3  
2  
1  
Blastoff!

반복 변수는 루프 한 번을 돌 때마다 변함

```
name = input('Enter file:')
```

순차문

```
handle = open(name, 'r')
```

반복문

```
counts = dict()
```

조건문

```
for line in handle:
```

```
    words = line.split()
```

```
    for word in words:
```

```
        counts[word] = counts.get(word,0) + 1
```

```
bigcount = None
```

```
bigword = None
```

```
for word,count in counts.items():
```

```
    if bigcount is None or count > bigcount:
```

```
        bigword = word
```

```
        bigcount = count
```

```
print(bigword, bigcount)
```

```
name = input('Enter file:')  
handle = open(name, 'r')
```

```
counts = dict()  
for line in handle:  
    words = line.split()  
    for word in words:  
        counts[word] = counts.get(word, 0) + 1
```

```
bigcount = None  
bigword = None  
for word, count in counts.items():  
    if bigcount is None or count > bigcount:  
        bigword = word  
        bigcount = count
```

```
print(bigword, bigcount)
```

파일에 있는  
단어 수 세기

사용자에게 데이터를  
받아 읽는 단어

글자 수를  
업데이트하는 문장

제일 큰 숫자를  
찾는 문단

# 요약

- 제1장의 개요
- 여기서 나온 개념을 앞으로 배울 예정
- 큰 그림을 생각해야 함

# Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Continue...

Initial Development: Charles Severance, University of Michigan School of Information

Contributor:

- Seung-June Lee ([plusjune@gmail.com](mailto:plusjune@gmail.com))
- Connect Foundation

Translator:

- Saejin Park
- Jeungmin Oh ([tangza@gmail.com](mailto:tangza@gmail.com))