

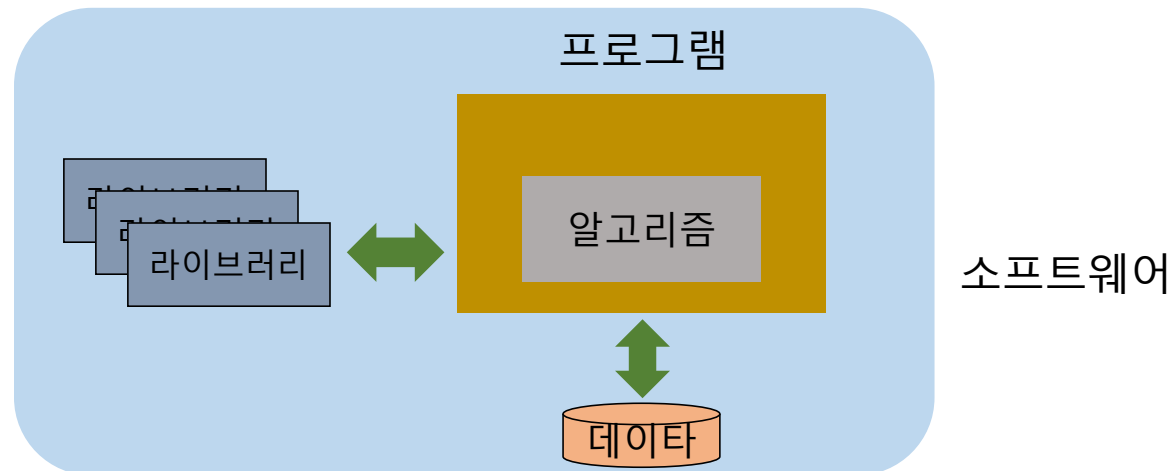
5주차

- 프로그램의 기능
- 함수
- 컴파일러

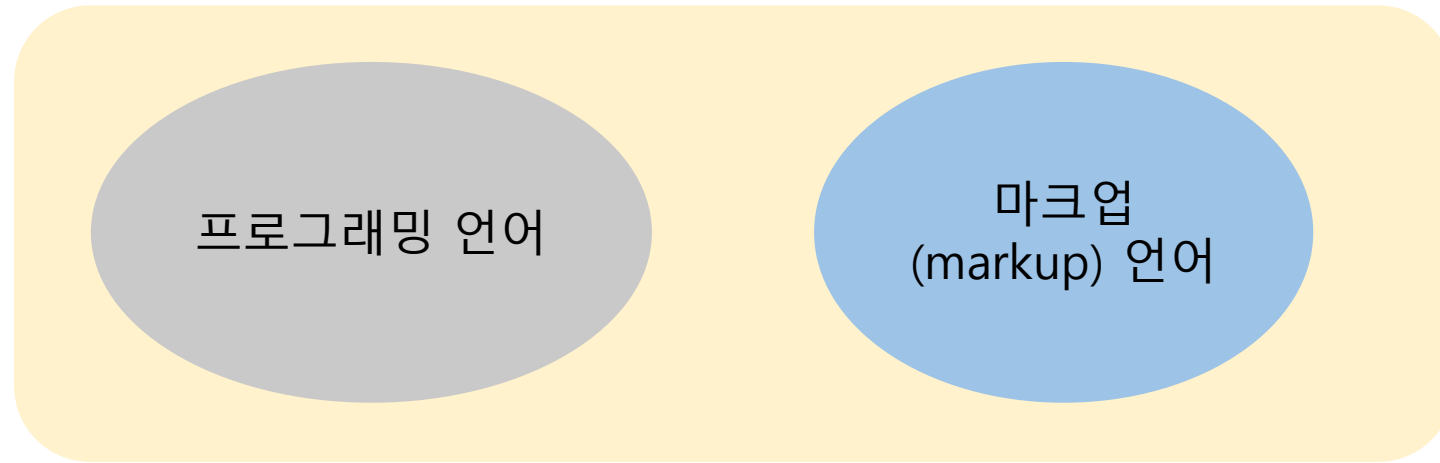
		강의 주제	주차
1부 컴퓨팅 사고력	컴퓨팅 사고력의 소개	IT 사회, 소프트웨어 세상, 컴퓨팅 사고력의 소개, 컴퓨팅 사고력의 개념	1
		컴퓨팅 사고력의 개념, 주위에서 볼 수 있는 컴퓨팅 사고력, 문제해결 방법	2
	문제해결 방법, 컴퓨터	문제해결 방법, 문제해결 과정 예, 문제해결을 위한 소프트웨어 설계 사상, 컴퓨터의 특징, 소프트웨어, 유한상태기계	3
2부 소프트웨어	알고리즘	알고리즘 소개, 알고리즘의 표현 방법, 의사코드, 흐름도	4
	프로그램	프로그램의 기능, 함수, 컴파일러	5
	파이썬	파이썬 소개 및 설치, 변수에 값 저장, 입력, 출력, 조건부 수행	6
		반복, 리스트, 함수, 출력 형식	7
3부 컴퓨팅 사고력 활용하기	데이터의 표현	이진수, 아스키코드, 오디오 데이터, 이미지 데이터, 자료구조	8
		인코딩 및 압축, 오류확인	9
	데이터의 저장과 검색	배열 및 연결 리스트, 선형검색, 이분검색, 색인순차검색, 해싱	10
		이진검색트리, 최대값 및 최소값 검색	11
	알고리즘설계	정렬, 분할정복 알고리즘, 탐욕적 알고리즘	12

프로그램은 무엇인가?

- 컴퓨터를 동작하게 하는 것은 소프트웨어
- 소프트웨어는 프로그램과 프로그램이 필요로 하는 데이터, 미리 만들어 놓은 프로그램에 해당하는 라이브러리들의 집합체
- 이 중에서 가장 중요한 역할을 하는 것이 프로그램
- 프로그램은 알고리즘을 포함하여 컴퓨터에서 가동할 수 있는 명령어들을 이용하여 수행하고자 하는 작업들을 정의한 것



컴퓨터 언어



- 일반적인 언어
- C, C++, Java, 파이썬
- HTML, SGML
- 프로그래밍 언어, 마크업 언어 외에 데이터베이스 질의어인 sql (structured query language) 언어 등이 있다.

프로그래밍 언어(programming language)

```
#include <iostream>
using namespace std;

int main( ){
    int a;
    cin >> a;
    cout << "Hello world " << a << endl;
    return 0;
}
```

C++ 프로그램 예

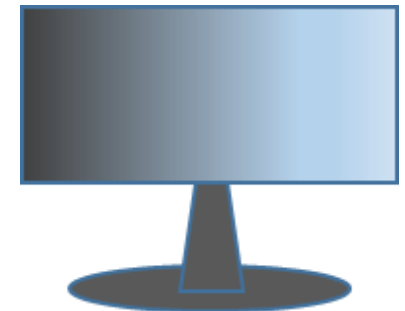
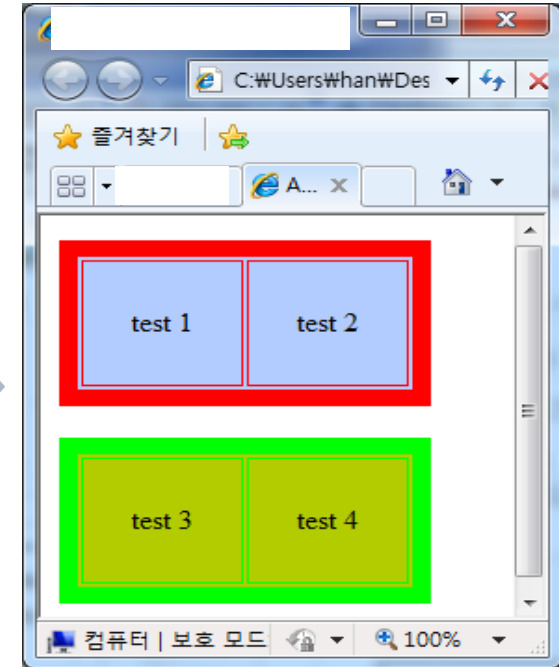


마크업 언어(markup language)

```
<p><table border=10 height="100" width=200 bordercolor="#ff0000" bgcolor="#B2CCFF">
<tr>
    <td width="100" align="center" valign="center">test 1</td>
    <td width="100" align="center" valign="center"> test 2</td></strong>
</tr>
</table>
</p>
<br><br>
<p><table border=10 height="100" width=200 bordercolor="#00ff00" bgcolor="#B2CC00">
<tr>
    <td width="100" align="center" valign="center">test 3</td>
    <td width="100" align="center" valign="center"> test 4</td></strong>
</tr>
</table>
</p>
```

html(HyperText Markup Language)

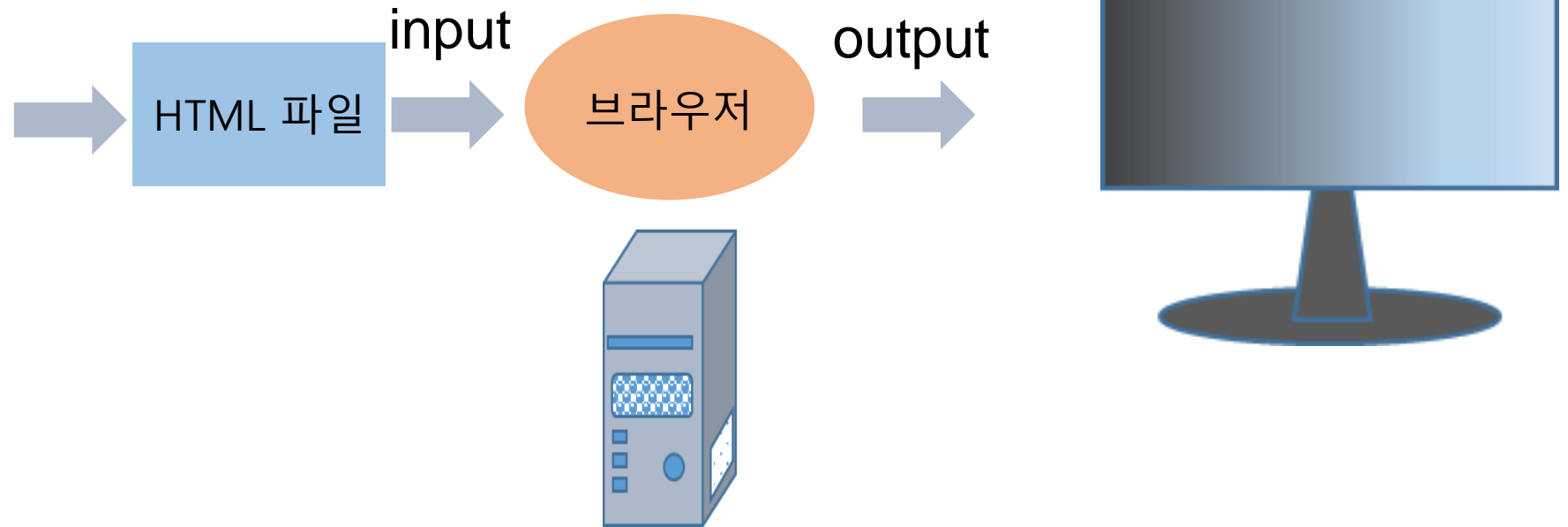
html 파일의 색깔 지정



html 파일과 브라우저(browser)의 관계



서버(server)



✓ 브라우저는 프로그래밍 언어로 만든다.

다양한 웹 브라우저



Mozilla Firefox



Internet Explorer



google chrome



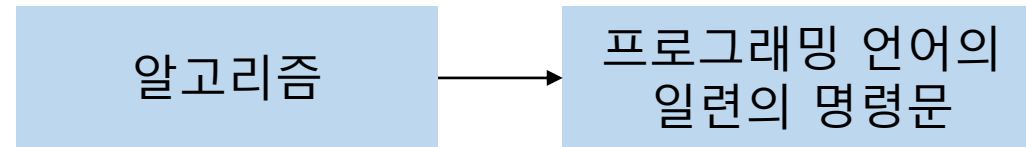
opera



safari

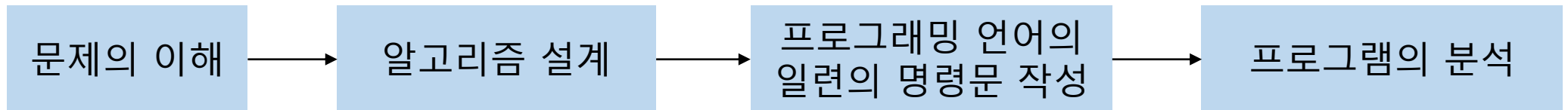
- 좁은 의미의 프로그래밍

- 수행해야 할 작업을 프로그래밍 언어의 일련의 명령어로 구성하는 작업

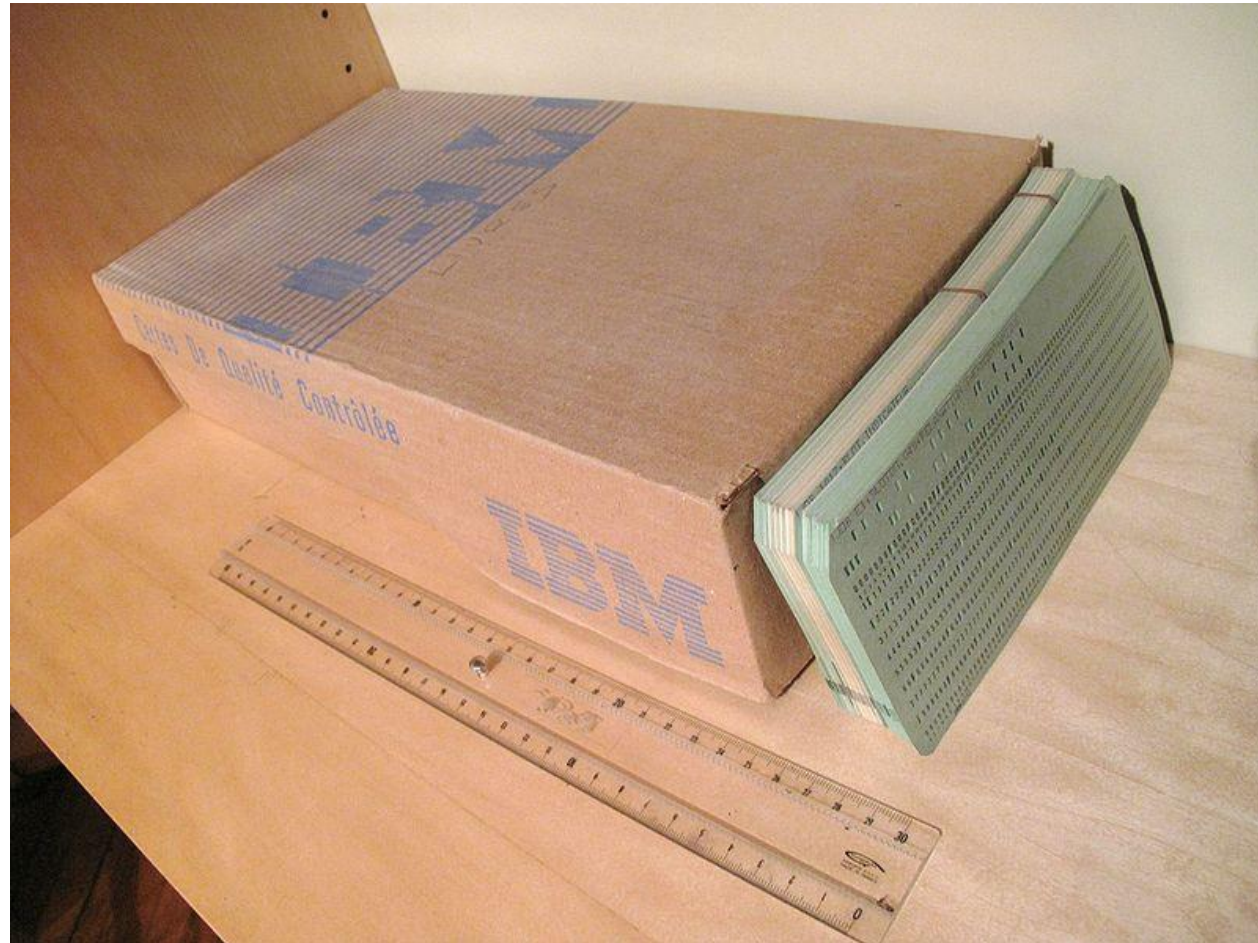
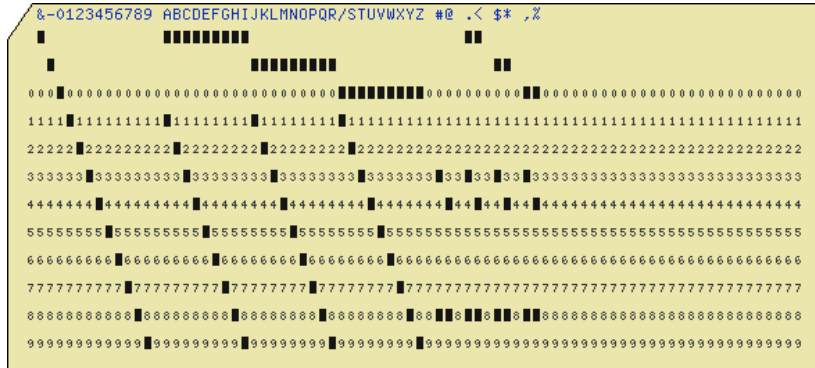


- 넓은 의미의 프로그래밍

- 프로그램 개발의 전체 과정

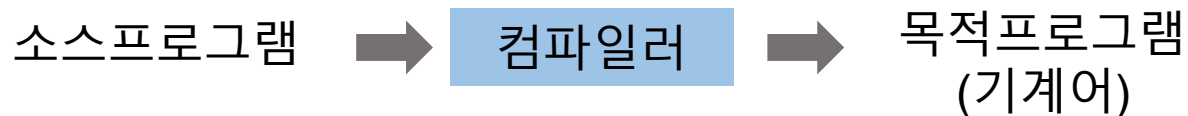


- 1980년 초까지 사용된 펀치카드를 이용한 하나의 프로그램
- 카드 한 장이 프로그램 1줄



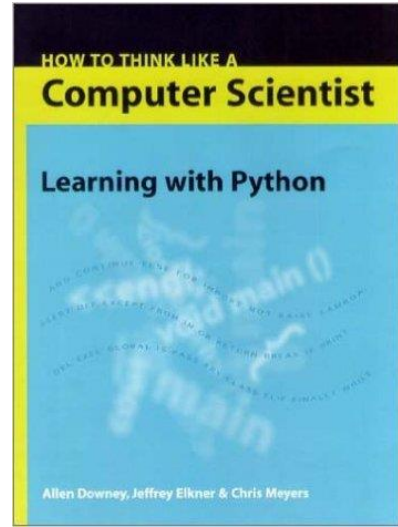
프로그래밍 언어의 특징

- 대부분 영어를 기반으로 구성되어 있다.
- 문법적인 구조를 갖는다
 - ✓ 어휘, 구문(syntax) – 언어의 형태
 - ✓ 의미(semantics)를 갖는다 - 언어의 의미
- 항상 동일한 의미로 해석된다.
- 프로그램을 기계어로 번역하는 컴파일러(compiler) 존재
- 인터프리터(interpreter) 언어 – 컴파일러 없이 바로 수행



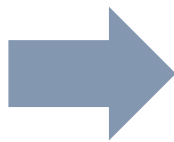
모든 컴퓨터 프로그래밍 언어에 있는 기능

Allen Downey와 2인, How To Think Like A Computer Scientist

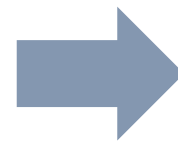


- 입력:
 - 키보드, 터치스크린, 마우스, 파일 등으로 부터 입력 데이터를 받는 기능
 - read, get, input, cin 등의 명령어를 사용한다.
- 출력:
 - 스크린, 파일, 종이 등으로 데이터를 출력하는 기능
 - write, put, print, cout 등의 명령어를 사용한다.
- 연산:
 - 더하기, 빼기, 곱하기, 나누기, 나머지연산, 논리연산 등의 연산을 수행하는 기능
 - +, -, *, /, %, >, <, =, == 등의 연산자를 사용한다.
- 선택-조건부 수행:
 - 조건을 판단하여 특정 작업을 수행하는 기능
 - if, else 등의 명령어를 사용한다.
- 반복:
 - 동일한 작업을 2회 이상 하는 기능
 - for, while, repeat, do 등의 명령어를 사용한다.

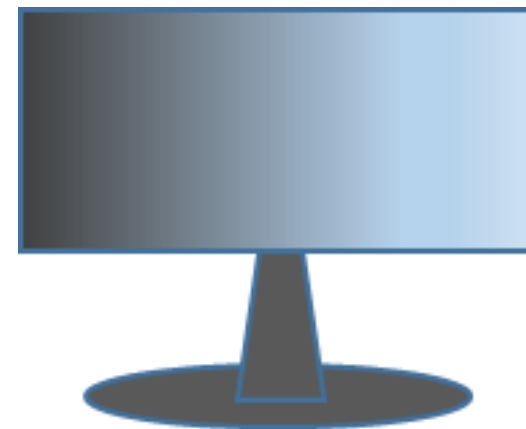




입력



출력



연산, 선택-조건부수행, 반복



연산 Arithmetic

- +
- -
- *
- /
- >
- <
- ==

선택-조건부 수행 (Conditional Execution)

- if
- else

반복 Repetition

- for
- while
- repeat
- do



언어에 따른 프로그래밍 기능 표현 방법

기능	python	java	C++	pascal	FORTTRAN
입력	input	read	cin	read, get	read
출력	print	println, print	cout	write, put, print	write
연산 (기본적인 연산자만 나열)	= + - * / ** == != < > > < >= <=	= + - * / % ++ -- == != > < >= <=	= + - * / % ++ -- == != > < >= <=	= + - * div mod < > < >= <=	= + - * / ** > < >= <= == /=
선택-조건부 수행	if ... else	if ... else	if ... else	if .. then .. else	if ... then ... endif
반복	for, while	for, while, do ... while	for, while, do ... while	for do while do repeat ... until	do

- 언어가 다르더라도 주요 기능은 유사하게 표현된다.



파이썬의 연산자

- arithmetic operators: + - * / % // **
- comparison operators: > < == != >= <=
- logical operators: and or not
- bitwise operators: & | ~ ^ >> <<
- assignment operators: = += -= *= /= %= //= **= &= !=
^= >>= <<=
- special operators: is 'is not'
- membership operators: in 'not in'

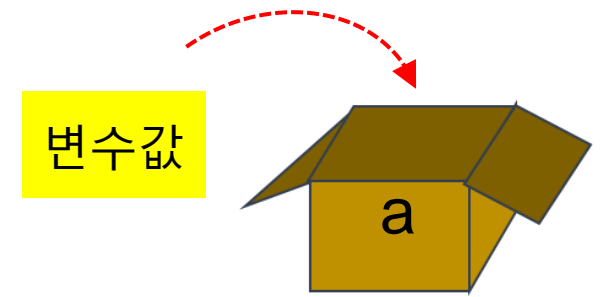


변수(variable)

- 변수는 하나의 물건을 저장하는 하나의 상자(box)와 같다.
- 하나의 값(value)을 저장하는 저장 위치의 이름
- 변수명: 영문 알파벳과 숫자로 구성.
- 변수명의 첫 문자는 영문 알파벳. 언어에 따라 첫 문자가 _(underbar, underscore)가 될 수도 있다.

(변수명 예) a, b123, c_12

- 저장하는 값은 숫자, 문자, 문자열(string) 등이 될 수 있다.
- 변수명과 변수가 갖고 있는 값은 다르다.
- 하나의 변수는 항상 하나의 값만 가질 수 있다.

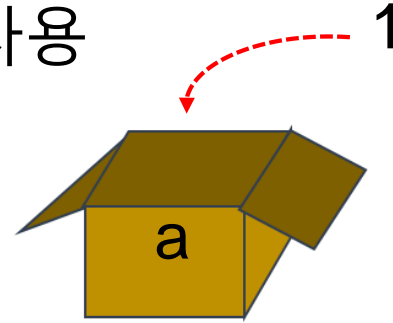


- 변수에 값을 저장하는 방법

- = 배정(assignment) 연산자를 사용

- $a = 1$

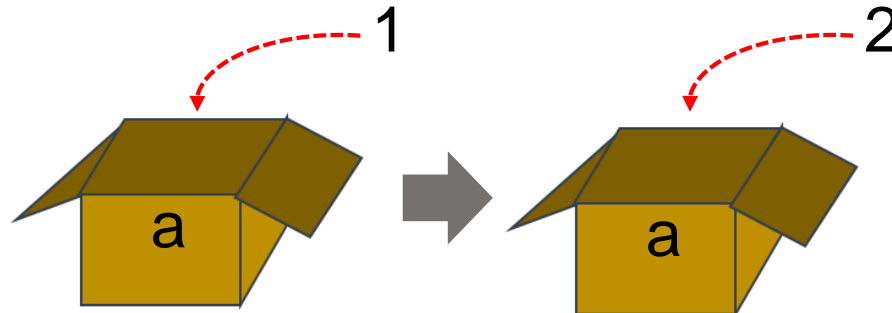
✓ 1의 값을 a에 저장한다



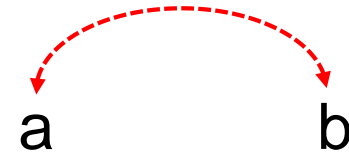
- $c = a + b$

✓ a와 b의 값을 더한 후 c에 저장한다.

- a=1 지정한 후 a=2를 수행하면 1의 값은 2의 값으로 변경. 전에 있던 값은 지워진다.

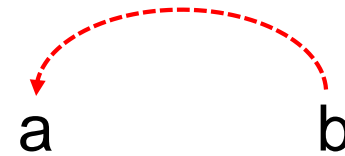


- $a=1, b=2$ 인 경우, 두 변수의 값을 맞교환할 때



틀린 방법

- $a=b$ 를 수행하면



- b 의 값을 a 에 저장한 결과가 된다.
- a 와 b 의 값이 같아진다.

올바른 방법

1. 새로운 변수 temp를 이용한다.
2. temp=a
3. a=b
4. b=temp

(1) temp
a b

(2) temp
a ↗ b

(3) temp
a ← b

(4) temp
a ↘ b



파이썬 프로그래밍 언어에서 변수의 값 교환 방법

(X)

```
a=5  
b=7  
print(a,b)  
a=b  
print(a,b)
```



5	7
7	7

(O)

```
a=5  
b=7  
print(a,b)  
temp=a  
a=b  
b=temp  
print(a,b)
```



5	7
7	5



입력, 출력, 연산 기능

- 입력
 - ✓ 데이터를 키보드, touch screen, 마이크를 통해 받아들이거나, 파일에서 읽어 들인다.
- 출력
 - ✓ 문자, 이미지를 화면에 보이게 하거나, 소리를 발생하거나, 프린터를 통해 프린트하거나, 또는 파일에 쓴다.
- 연산
 - ✓ 더하기, 빼기, 곱하기, 나누기 등의 연산을 수행한다.



● 일반적 프로그램 예

프로그램

입력: salary 값을 받아 들인다.

연산: salary 값에 0.1을 곱해 tax 에 저장

출력: 문자열과 tax를 출력

```
input salary
tax = salary * 0.1
print "세금:" tax "원"
```

화면

200
세금: 20원

- input, print 는 무슨 동작을 해야 하는지를 나타내는 명령문이다.
- salary, tax 는 값을 저장하는 변수다.
- 프로그램과 출력 설명

```
print "세금:" tax "원"
```

세금: 20원

“ - - “ 의 내용은 화면에
그대로 출력된다.

tax 변수가 갖고
있는 값을 출력한다.



입출력 프로그램 예

Pascal

```
program function1;  
  
VAR Name:string;  
  
BEGIN  
    Write('type your name: ');  
    Read(Name);  
    Write('Hello ', Name);  
END
```

출력

```
type your name: kim  
Hello kim
```



파이썬

```
a=input("type your name: ")  
print("Hello ", a)
```

출력

```
type your name: kim  
Hello kim
```

C++

```
#include <iostream>  
#include <string>  
using namespace std;  
  
int main(){  
  
    string a;  
  
    cout <<"type your name: ";  
    cin >> a;  
    cout << "Hello " << a <<endl;  
    return 0;  
}
```



연산 프로그램 예

파이썬

```
a=1
b=2
c=3
d=a+b*c
print(d)
```

출력

7

C++

```
#include <iostream>
using namespace std;

void main() {
    int a,b,c,d;
    a=1;
    b=2;
    c=3;
    d=a+b*c;
    cout << d;
}
```



선택 - 조건부 수행

내일 비가 오면 놀이공원에 가고 비가 안 오면 야구장에 갈 예정이다.

```
if (내일 비가 오면) then  
    놀이공원에 간다  
else  
    야구장에 간다
```



if (온도>30) then

날씨가 덥다

else if ($20 \leq \text{온도} < 30$) then

날씨가 온화하다

else

날씨가 춥다



조건부 수행의 중첩

```
if (내일 비가 오면) then
    if (친구가 시간이 된다면) then
        같이 놀이공원에 간다.
    else
        혼자 놀이공원에 간다.
else
    야구장에 간다
```



Java

```
public void processA ( ){  
    do {  
        int temperature = inputData();  
        if(temperature <0)  
            warmup(temperature);  
        else  
            cooldown(temperature);  
    }  
    while(1);  
}
```



[선택-조건부 수행 예] 두 숫자를 받아서, 큰 숫자를 출력한다.

파이썬

```
print("type two numbers :")
a=input("1st number :")
b=input("2nd number :")
if a>b:
    print(a)
else:
    print(b)
```

출력

```
type two numbers :
1st number : 3
2nd number : 5
5
```

C++

```
#include <iostream>
using namespace std;
void main(){
    int a,b;
    cout << "type two numbers :"<<endl;
    cout << "1st number : ";
    cin >>a;
    cout << "2nd number : ";
    cin >>b;
    if(a>b)
        cout << a;
    else
        cout << b;
}
```



반복의 기능

반복의 기능 예

1) $S = 1+2+3+\dots+ 10,000$

2) $S = 2 + 4 + 6 + \dots + 10,000$ (간격을 정의해야 함)

3) 실내 온도가 27도가 될 때까지 보일러를 가동하라 (while)



[반복 예] 1부터 10까지의 합을 계산하여 출력한다.

파이썬

```
sum=0
for i in range(1,11):
    sum += i
print("Sum is ", sum)
```

출력

Sum is 55

C++

```
#include <iostream>
using namespace std;

void main() {
    int i, sum=0;
    for(i=1; i<11; i++)
        sum += i;
    cout << "Sum is " << sum;
}
```



[반복 예] 1부터 n까지의 합이 100 이하가 되는, n의 최대값을 계산하여 출력한다.

```
n=1
sum=1
while sum <= 100:
    n=n+1
    sum=sum+n
print(n-1)
```

13



포트란(Fortran)

```
PROGRAM FIBONACCI
IMPLICIT NONE
INTEGER, PARAMETER :: N = 10 !numero de elementos desejados na sequencia de fibonacci
REAL, PARAMETER    :: A = ((1+SQRT(5.))/2) !proporcao Aurea
REAL, DIMENSION (1:N) :: FIB = [(NINT((A**I - (1-A)**I)/SQRT(5.))), I=1,N]
INTEGER :: I

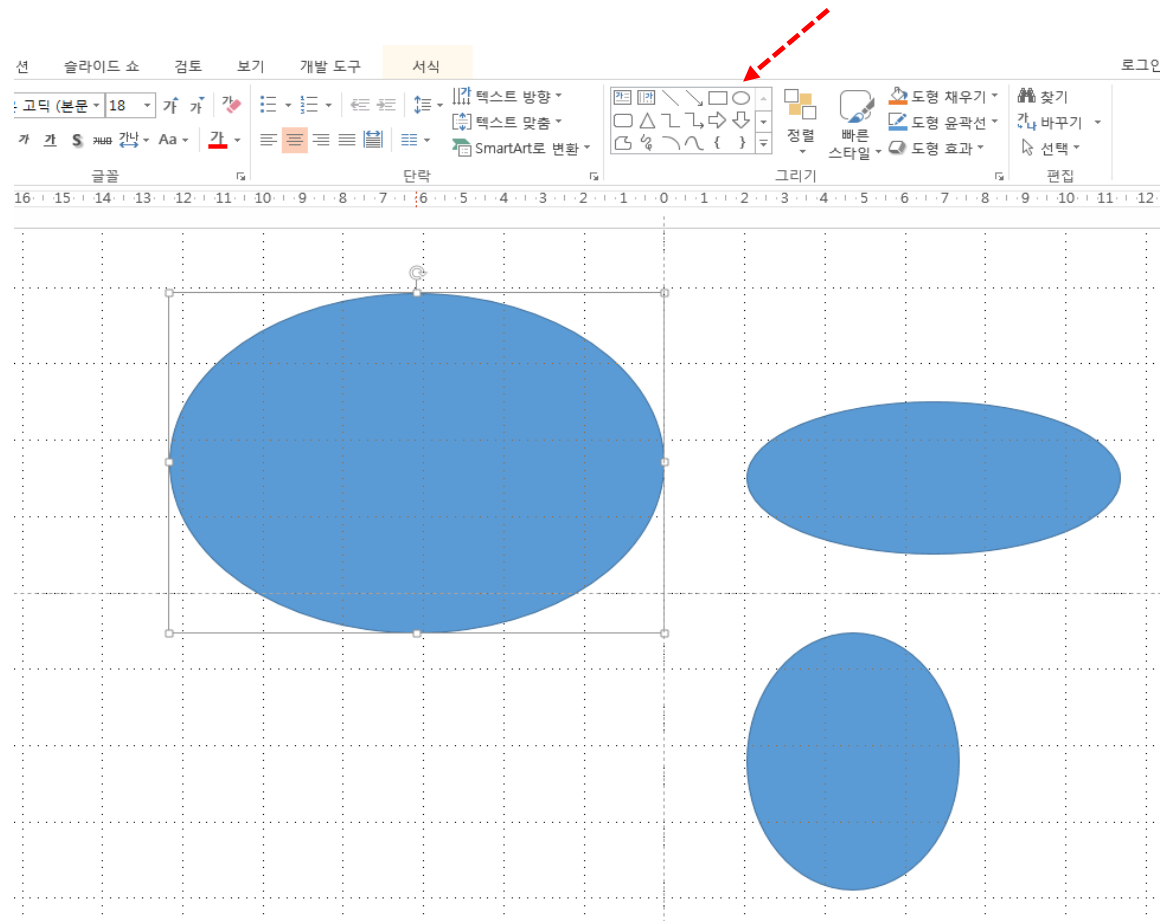
WRITE(*, "(A,I3,A) ") "SEQUENCIA DE FIBONACCI COM",N," ELEMENTOS"
DO I = 1,N
    WRITE(*,*) FIB(I)|
END DO

PAUSE

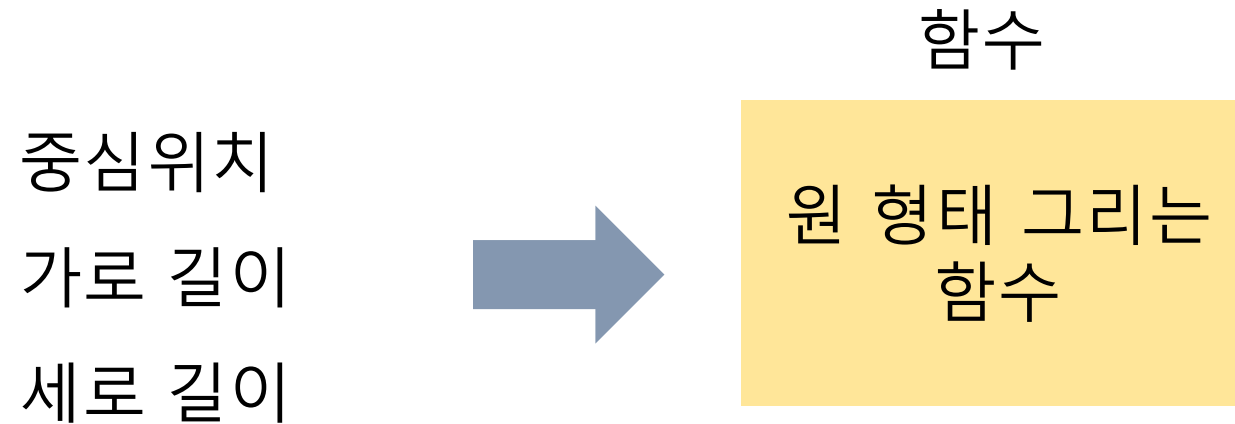
END PROGRAM FIBONACCI
```

함수

- 파워포인트 작성 시 원 그리는 기능을 이용해 다양한 원 형태(타원형 포함)를 그릴 수 있다.
- 중심위치와 가로, 세로 길이로 다양한 형태 생성 가능 – 함수 기능



함수 형태로 기능을 만들면 다양한 입력에 대해 공통적인 작업을 수행한다.



원 형태를 그린다.

함수(function)

작업 → 현재 현금이 200원 있다. 은행에 예금했을 때 연이자율이 10%이면 이자는 20원이 되고, 연이자율이 20%이면 이자는 40원이 된다.아래의 출력처럼 화면에 보이도록 한다.

출력

이자율 0.1	이자 20원
이자율 0.2	이자 40원



- 프로그램 예1

프로그램

```
interest = 200 X 0.1  
print “이자율 0.1 이자” interest “원”  
interest = 200 X 0.2  
print “이자율 0.2 이자” interest “원”
```

출력

이자율 0.1 이자 20원
이자율 0.2 이자 40원

- 프로그램과 출력 설명

```
print “이자율 0.1 이자” interest “원”
```

“ - - “의 내용은 화면에
그대로 출력된다.

interest 변수가 갖고
있는 값을 출력한다

이자율 0.1 이자 20원



반복



```
interest = 200 X 0.1  
print "이자율 0.1 이자" interest "원"  
interest = 200 X 0.2  
print "이자율 0.2 이자" interest "원"
```



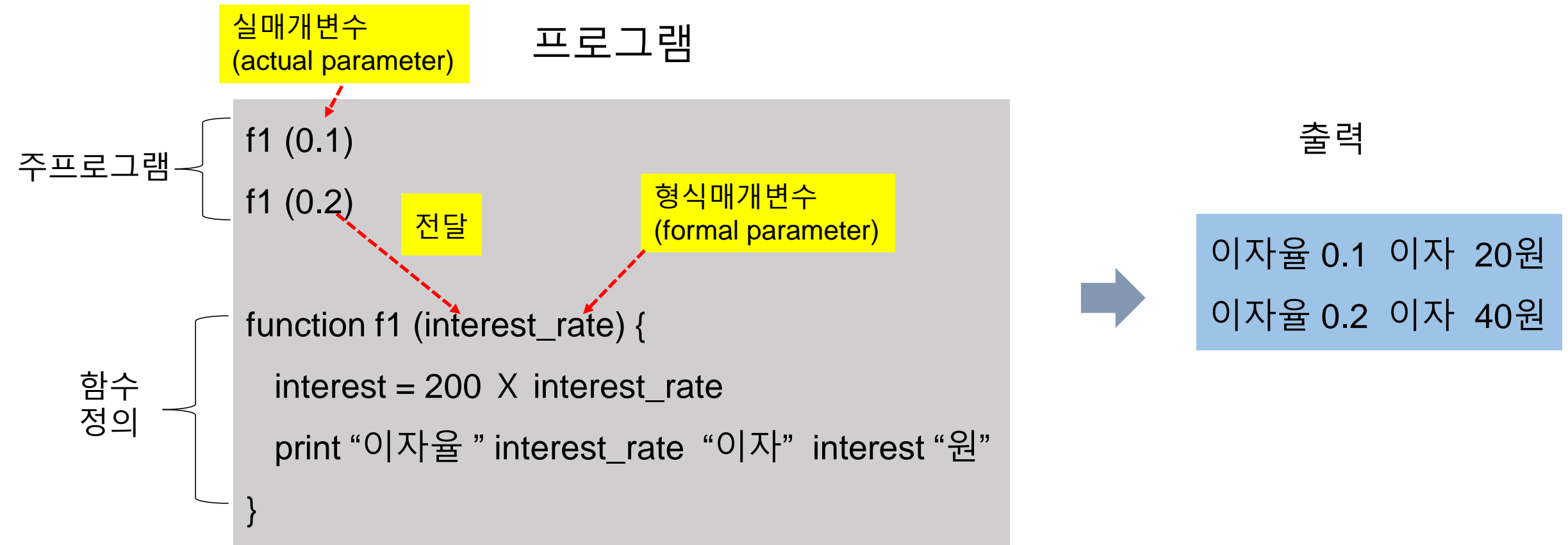
```
이자율 0.1 이자 20원  
이자율 0.2 이자 40원
```

반복

- 여기에는 동일한 작업이 2회 반복되고 있으므로, 반복되는 작업을 하나의 함수, 함수f1, 로 정의하고, main에서 그 함수를 2회 호출한다.
- 함수를 호출할 때 interest의 값만 달라진다.
 - ✓ 첫 번째는 0.1, 두 번째는 0.2



● 프로그램 예2(함수의 사용, 형식매개변수 사용)



- 주 프로그램 내에서 함수 f1을 호출(call)하였다 라고 한다.
- 주 프로그램 내에서 함수를 호출할 때 넣어 준 값(실매개변수)이, 함수 정의 내에 형식 매개변수로 전달되어 계산된다.
- 처음에는 0.1, 두 번째는 0.2가 전달되어 계산된다.



주프로그램 f1(0.1)

f1(0.2)

함수 정의

```
function f1(interest_rate){  
    ..  
    ..  
}
```



- 프로그램 예3(함수의 사용, 형식매개변수 2개 사용)

프로그램

```
f2 (200, 0.1)
```

```
f2 (200, 0.2)
```

순서대로 대응

```
function f2 (money, interest_rate) {  
    interest = money X interest_rate  
    print “이자율” interest_rate “이자” interest “원”  
}
```

출력

이자율 0.1 이자 20원
이자율 0.2 이자 40원

- main 내에서 함수를 호출할 때 넣어 준 두 개의 값이, 함수 정의 내에 형식매개변수로 순서대로 각각 전달되어 계산된다.



주프로그램

f2(200, 0.1)

f2(200, 0.2)

함수 정의

```
function f2(money, interest_rate){  
    ..  
    ..  
}
```



- 프로그램 예4(함수의 사용)

프로그램

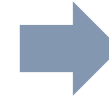
```
f2 (1400, 0.2)
```

```
f2 (2000, 0.15)
```

```
f2 (20000, 0.1)
```

```
function f2 (money, interest_rate) {  
    interest = money X interest_rate  
    print “이자율” interest_rate “이자” interest “원”  
}
```

출력

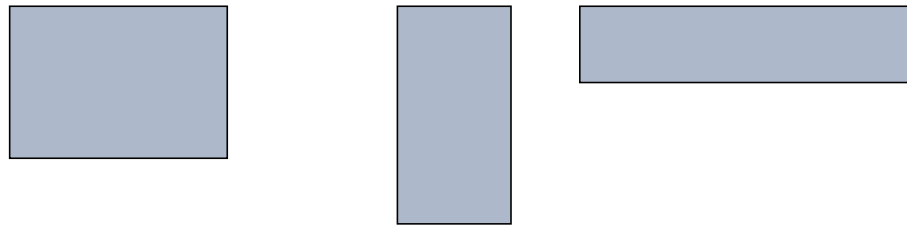


```
이자율 0.2   이자   280원  
이자율 0.15  이자   300원  
이자율 0.1   이자  2000원
```

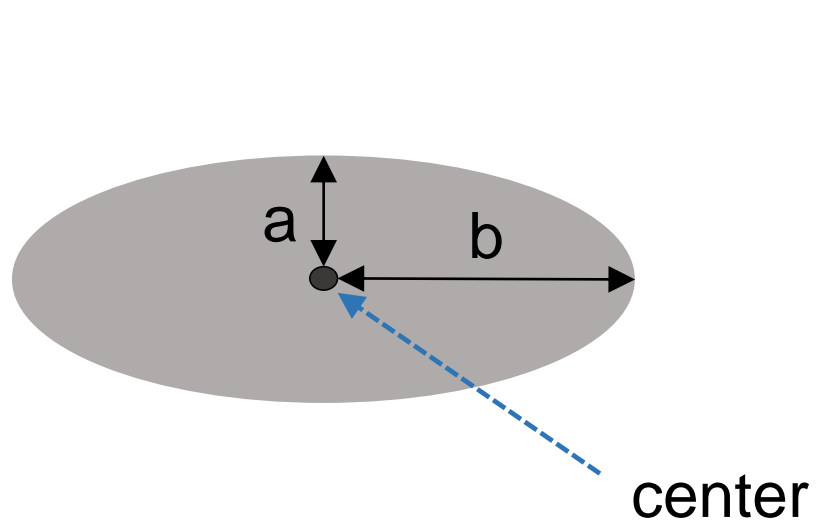


- ✓ 함수는 반복적으로 사용되는 절차를 하나의 단위로 만들어 매개변수의 값만 바꾸면서 다양한 입력에 대한 결과를 만들어 낼 수 있다.

(예) 사각형 그리는 함수, 성적 계산하는 함수 등



- 하나의 사각형 그리는 함수를 이용하여 다양한 사각형 생성 가능
- 가로, 세로의 길이를 매개변수로 설정



`circle(center, a, b)`



`rectangle(vertex, c, d)`



	국어	영어	수학	과학
홍길동	90	80	79	91
이순신	70	78	90	92
김유신	89	75	85	90
...				
...				

만일 function `average(학생이름)`가 각 학생의 점수를 받아서, 평균을 계산한다면,

main {
`average(홍길동)`
`average(이순신)`
`average(김유신)`

}

이 작업으로 모든 학생의 평균을 구할 수 있다.

● 함수의 기능

- 하나의 큰 프로그램은 많은 수의 함수들로 구성된다.
- 다양한 함수들이 이미 만들어져 있다.
- 라이브러리(library): 함수들의 집합
 - ✓ 그래픽 라이브러리
 - ✓ 통계분석 라이브러리
 - ✓ 수학연산 라이브러리
- 많은 프로그램들은 다양한 라이브러리를 연결하여 구축할 수 있다.

컴파일러

Compiler

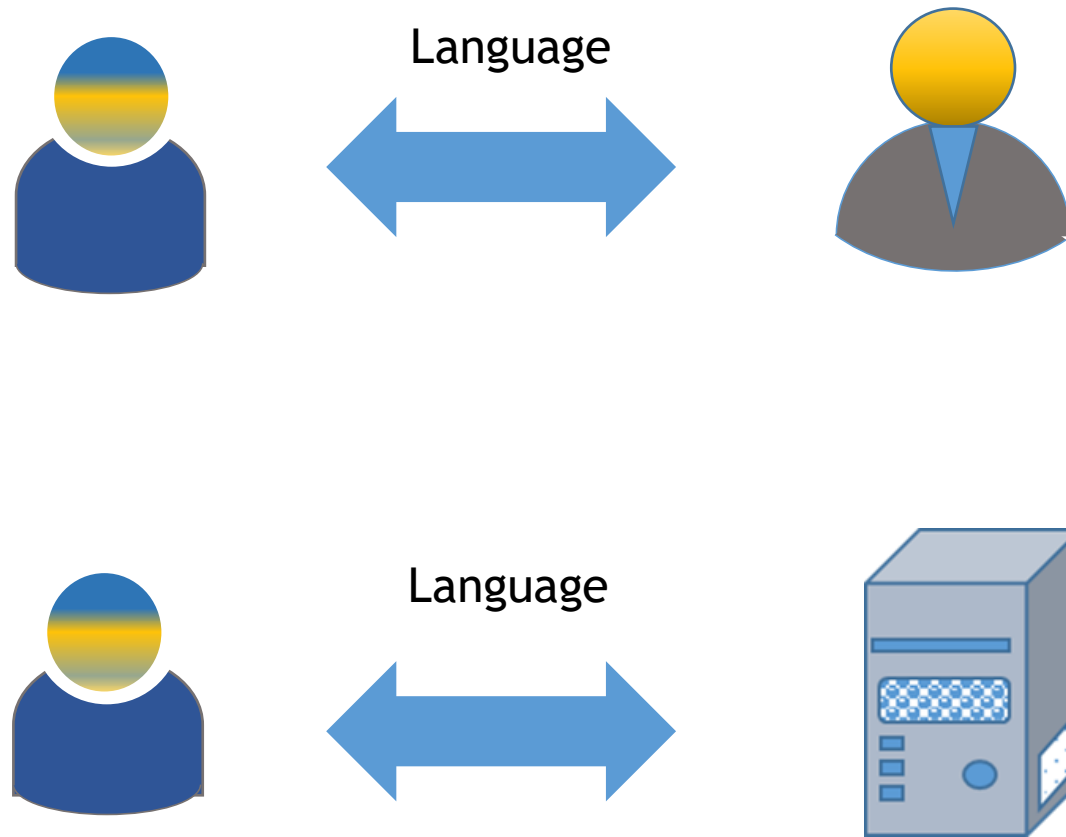
- 소스프로그램을 컴퓨터에서 수행가능 하도록 기계어로 바꾸어 주는 번역기(소프트웨어)
- C 컴파일러, C++ 컴파일러 등



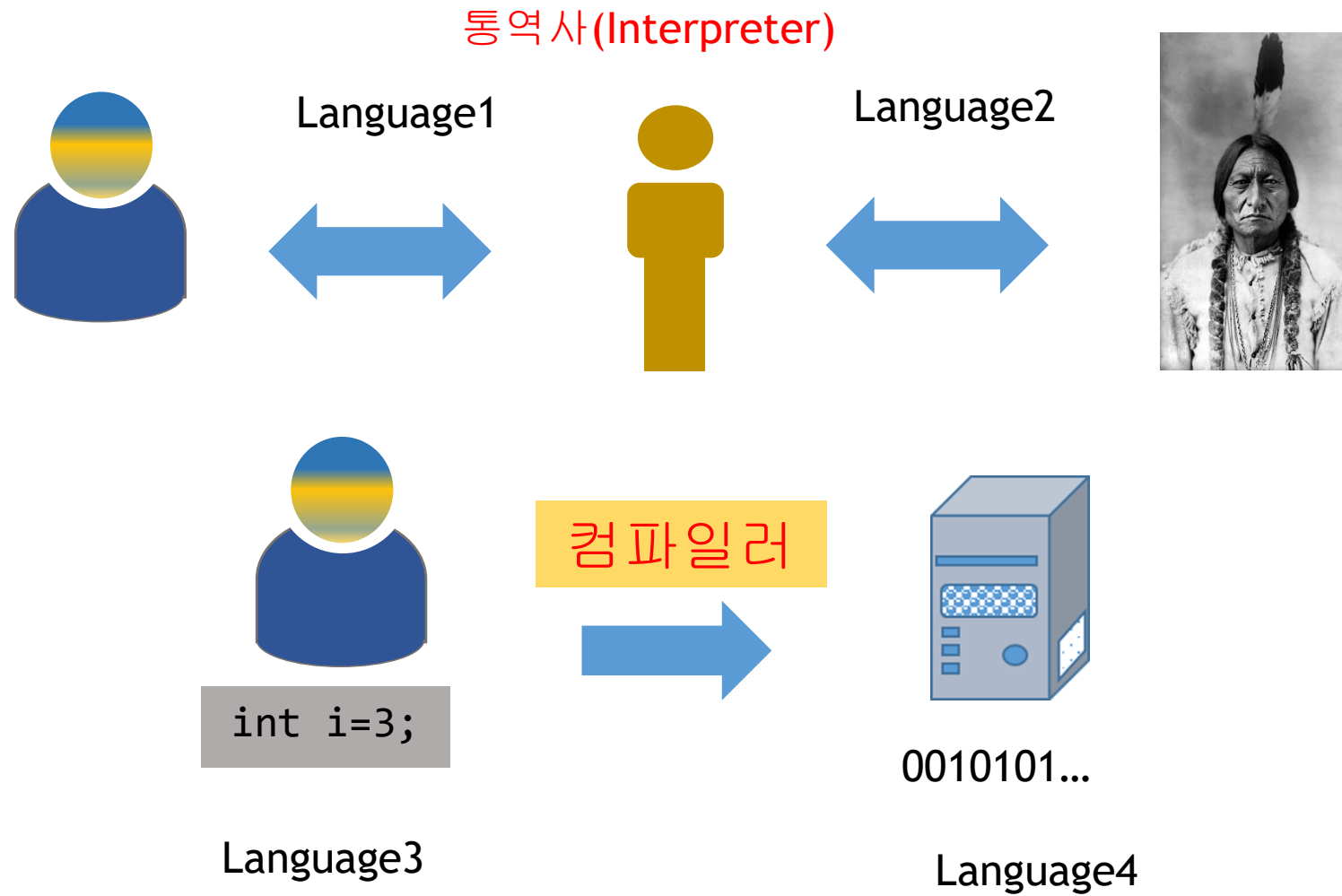
compile과정 유무에 따른 프로그래밍 언어 분류

- 컴파일러에 의해 컴파일 되는 언어: C, C++, COBOL, FORTRAN, Java
 - ✓ 번역된 후 수행된다.
- 인터프리터에 의해 수행되는 언어: python, basic, JavaScript, MATLAB, PHP, Lisp
 - ✓ 번역과정 없이 바로 수행된다.

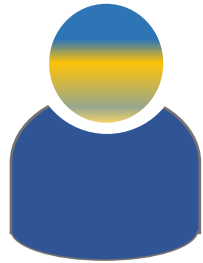
만일 같은 언어를 사용한다. 통역사가 필요하지 않다.



만일 서로 다른 언어를 사용한다면,.....

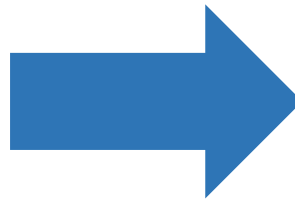


- ✓ 컴파일러는 사람이 사용하는 프로그래밍 언어로 작성된 프로그램을 기계가 인식하는 형식으로 바꾼다(번역한다).



```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!"<<endl;
    return 0;
}
```



컴파일러



1	0	1	0	1	1	0	1	0	1	0	1	1	0
1	0	1	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	0	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	0
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0	1	0	0	1
0	1	1	0	1	0	0	0	1	1	0	1	0	0
1	1	0	1	0	0	1	1	1	0	1	0	0	1
1	0	1	0	1	1	0	1	0	1	0	1	1	0
1	0	1	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	0	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	0
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0	1	0	0	1
0	1	1	0	1	0	0	0	1	1	0	1	0	0
1	1	0	1	0	0	1	1	1	0	1	0	0	1



컴파일러의 오류 메시지

C++

```
#include <iostream>
using namespace std;
void main() {
    int a,b;
    a=1;
    b=2;
    3=a+b;
}
```

error C2106: '=' : left operand must be l-value



오류의 종류가 다르다

	국문예	C++	오류 메시지와 확인단계
정상 문장	나는 학교에 간다	int i=2;	
어휘의 오류	나는 학교 예 간다	iny i=2;	<ul style="list-style-type: none"> - 'iny': undeclared identifier - 어휘분석(lexical analysis) 단계에서 오류 확인
구문의 오류	나는 위대한 간다	int float=2;	<ul style="list-style-type: none"> - 'int' followed by 'float' is illegal - 구문분석(syntax analysis) 단계에서 오류 확인

- 컴파일러는 작성된 프로그램을 확인하여 오류를 확인한다.
- 오류가 없을 경우 수행모듈을 생성한다.



파이썬의 오류 메시지

파이썬

```
>>> a=1
>>> b=3
>>> 3=a+b
SyntaxError: can't assign to literal
```

- ✓ 파이썬은 컴파일러가 아닌 인터프리터(interpreter)가 프로그램의 적합성을 검사

- 소스프로그램이 컴파일러를 거쳐 수행되기까지의 단계

