

정보 : 담당교사 민정원/장희주

민정원 : 1,3,5,7,9,11 장희주 : 2,4,6,8,10

시수배정 : 3시간

1. 평가계획 : 지필평가(40%) + 수행평가(60%)

항목	세부내용	반영 비율	비고
지필 평가	개념평가	40%	학습내용을 이해한 정도를 평가
수행 평가	프로그래밍	30%	제시된 문제를 프로그램으로 작성
	과제물	30%	학습과제물 수행

\* 학습지는 결석일로부터 2주 동안만 배부가능

준비물 : 필기도구, 교과서, 파일철

컴퓨터실 사용시 주의사항 : 음식물 노노, 자리가동금지, 정리정돈



## 2. 수행평가계획

영역	배점	평가항목	평가기준	점수
프로그래밍	100	제시된 문제의 프로그램 완성도 평가	부여된 문제의 통과비율에 따라 배점을 다르게 부여 (* 소수 둘째자리에서 반올림)	(통과비율/2.5) + 60
			미참석	40
과제물	100	수시로 과제물을 부여하여 과제수행 비율로 평가	부여한 과제의 96% 이상 수행	100
			학습과제물의 수행비율에 따라 배점을 다르게 부여 (* 소수 둘째자리에서 반올림)	(통과비율/2.4) + 60
			미제출	40

- ① 수행평가에 정상적으로 참여한 학생의 기본 점수는 영역별 수행평가 점수의 40%(40점)를 부여한다.
- ② 수행평가에 참여하지 못한 학생은 가급적 추후 기회를 다시 부여하되 단, 수행평가에 참여하지 않아 기본점수를 부여할 수 없는 경우(미참석자, 미완성자, 수업불성실참여자, 무단결과 등)는 영역별 수행평가 만점의 30%(30점)를 부여한다.
- ③ 전편입, 장기결석 등으로 부득이하게 추후평가가 이루어지지 못하는 경우, 석차산출단위별 평균점수를 기준점수로 하여 공결은 100%, 병결 80%를 부여하고, 무단결석은 영역별 수행평가 만점의 30%(30점)부여한다.

- IT기술이 자신의 진로분야에 어떻게 적용되는지 3가지 이상 조사하고 기술 활용을 위해서 갖추어야 할 역량에 대해 비주얼씹킹해보자!
- 비주얼씹킹 : 단순그림이 아니라 자신만의 방식으로 표현하면서 생각하는 힘을 기를 수 있는 공부방법

### Ⅲ. 문제해결과 프로그래밍

#### 1. 추상화

CT = Computational Thinking

분해 : 복잡한 문제를 작은 부분으로 분해

추상화 : 세부사항 생략 일반적인 문제해결법 탐구

알고리즘 : 문제들 간에 유사점 발견

패턴인식 : 문제해결을 위해 단계별로 알고리즘화

1-1. 문제분석 : 문제의 본질을 파악

1-2. 문제분해 : 복잡한 문제를 여러 개의 작은 단위의 문제로 나눔

1-3. 모델링 : 핵심요소만을 남겨 구조화 시킨 것(표,그림,그래프 등등)

#### 2-1. 알고리즘 설계

알고리즘 : 문제를 해결하기 위한 논리적인 절차나 방법

##### ■ 순서도에 사용되는 기호와 의미

기호	의미	기호	의미
	순서도의 시작과 끝		조건의 참,거짓 판단
	데이터 처리		작업을 반복수행
	자료의 입출력		순서도의 흐름

##### ■ 알고리즘과 제어구조

순차	선택	반복
순서대로 진행	조건에 따라 선택해서 진행	주어진 조건을 만족하면 특정부분 반복

\* 주변의 문제상황을 해결할 수 있는 알고리즘을 설계해 보자.

- 문제를 해결하기 위한 절차를 순서도로 표현

예) 유튜브 동영상을 추천,  
온습도 자동으로 조절하는 스마트홈,  
탄소중립을 위한 쓰레기 분리수거 시스템

2-2. 알고리즘 분석

■ 탐색 : 여러 개의 자료 중에서 원하는 자료를 찾아내는 것

- 1) \_\_\_\_\_ : 처음부터 차례대로 비교하여 원하는 자료를 찾는 방법
- 2) \_\_\_\_\_ : 자료가 정렬된 상태에서 중앙에 있는 값을 기준으로 왼쪽과 오른쪽 자료를 비교하면서 탐색

탐색방법	순차탐색							이진탐색						
탐색 전 카드	3	4	5	6	7	8	9	3	4	5	6	7	8	9
1단계	3	4	5	6	7	8	9	3	4	5	6	7	8	9
2단계	3	4	5	6	7	8	9	3	4	5	6	7	8	9
3단계	3	4	5	6	7	8	9	3	4	5	6	7	8	9
4단계	3	4	5	6	7	8	9	3	4	5	6	7	8	9
5단계	3	4	5	6	7	8	9	3	4	5	6	7	8	9
탐색횟수														

3) 두 탐색 방법의 장단점

구분	순차탐색	이진탐색
장점		
단점		

■ 정렬 : 여러 개의 자료를 기준에 따라 나열하는 것

- 1) \_\_\_\_\_ : 인접한 두 개의 자료를 비교하여 자리를 교환하는 방법
- 2) \_\_\_\_\_ : 정렬하지 않은 자료의 첫 번째 위치를 기준위치로 정하고, 나머지 자료 중 가장 작거나 큰 자료를 찾아서 기준위치에 자료를 교환

정렬방법	버블정렬					선택정렬				
단계	5	9	7	3	4	5	9	7	3	4
최종										
비교횟수										
교환횟수										

3) 두 정렬 알고리즘의 장단점

구분	버블정렬	선택정렬
장점		
단점		

### 3. 프로그래밍

#### 3-1. 프로그래밍 환경

\_\_\_\_\_ : 컴퓨터가 이해할 수 있게 만든 기계 중심언어

예 : \_\_\_\_\_

\_\_\_\_\_ : 사람이 이해할 수 있게 만든 인간 중심언어

예 : \_\_\_\_\_

#### ■ 언어번역 프로그램 종류

\_\_\_\_\_ : 어셈블리어를 기계어로 번역

\_\_\_\_\_ : 고급언어로 작성된 원시코드를 한꺼번에 번역

\_\_\_\_\_ : 고급언어를 한 줄씩 통역

#### ■ 파이선 프로그램 개발 환경 혹은 Thonny

1) [www.python.org/](http://www.python.org/) 혹은 [www.thonny.org](http://www.thonny.org)

2) 다운로드 > 운영체제에 맞는 최신버전 저장 및 실행


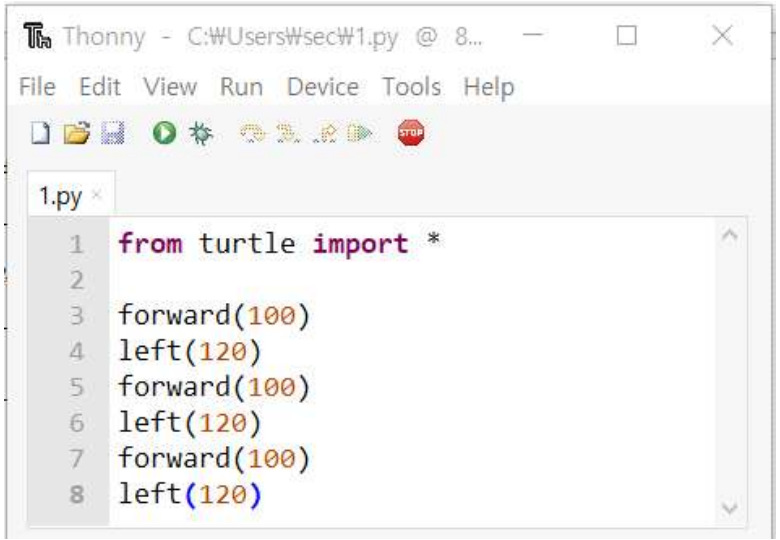
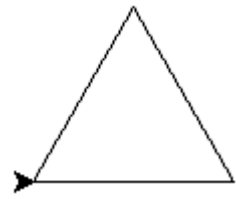
**Thonny**  
Python IDE for beginners

Download version **3.2.4** for  
[Windows](#) • [Mac](#) • [Linux](#)

**NB!** Windows installer is signed with new identity and you may receive a warning dialog from Defender until it gains more reputation.

Just click "More info" and "Run anyway".

#### ■ Thonny로 파이선 프로그램 맛보기

step1 >	 Thonny 실행
터틀 모듈로도 형 그리기	 <pre> 1 from turtle import * 2 3 forward(100) 4 left(120) 5 forward(100) 6 left(120) 7 forward(100) 8 left(120)         </pre>
실행 결과	

### 3-2. 변수와 자료형

\* 변수: 자료를 저장할 기억장소

\* 특징: 가장 마지막 값만 기억

\* 자료형 : 변수에 저장될 자료의 형태  
예) 정수형(int), 실수형(float), 문자형(str)

\_ int : \_\_\_\_\_

\_ float : \_\_\_\_\_

\_ str : \_\_\_\_\_

#### ■ 변수명 작성 방법

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

\*잠깐~! 프로그래밍 실력을 업시켜주는 코드업([codeup.kr](http://codeup.kr)) 알아?


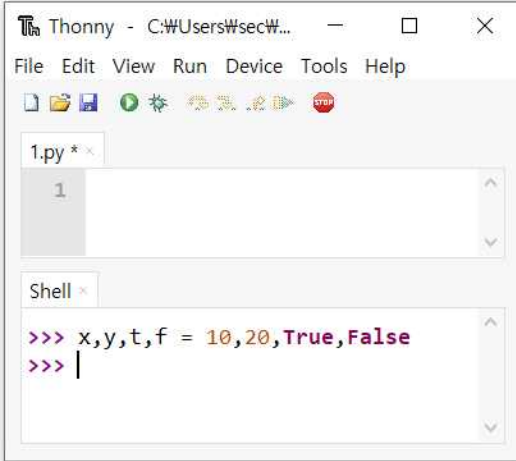
step1>		IDLE 실행 -> 파이썬 셸창 실행
step2>		<p>명령어 입력 후 변수의 자료형 확인 a,b,c,d의 변수형을 쓰시오.</p> <p>a: b: c: d:</p>
step3>		<p>print(x+y)의 실행결과와의 차이점을 생각해보자.</p>

### 3-3. 연산자

종류	연산방향	연산자	기능	사용예
산술			더하기	
			빼기	
			곱하기	
			나누기	
			나머지	
비교			작다	
			크다	
			작거나 같다	
			크거나 같다	
			같다	
논리			같지 않다	
			논리곱	
			논리합	
대입			논리부정	
			대입	

#### 연산자 우선 순위


1	** (거듭제곱)
2	+, - (부호)
3	*, /, %
4	+, - (더하기, 빼기)
5	<, >, <=, >=
6	==, !=
7	=, +=, -=, *=, /=, %=

step1>		IDLE 실행 -> 파이선 셸창 실행																
step2>		<ul style="list-style-type: none"> <li>사칙</li> <li>비교</li> <li>논리</li> <li>대입</li> </ul> <p>연산명령어 입력 하고 F5를 눌러 연산결과를 확인한다.</p> <ul style="list-style-type: none"> <li>True,False 대소문자 주의</li> </ul>																
step3>	<p>연산의 결과는?</p> <table border="1"> <thead> <tr> <th>연산식</th> <th>결과</th> </tr> </thead> <tbody> <tr> <td><math>(x + y) &lt;= (x - y)</math></td> <td></td> </tr> <tr> <td><math>(x * y) != (x / y)</math></td> <td></td> </tr> <tr> <td><math>x \% y == x</math></td> <td></td> </tr> <tr> <td><math>x &lt; y \text{ and } x &gt;= y</math></td> <td></td> </tr> <tr> <td><math>x != y \text{ or } x == y</math></td> <td></td> </tr> <tr> <td><math>\text{not } x &gt; y</math></td> <td></td> </tr> <tr> <td><math>x &lt;= y \text{ and } x == y \text{ or } x != y</math></td> <td></td> </tr> </tbody> </table>	연산식	결과	$(x + y) <= (x - y)$		$(x * y) != (x / y)$		$x \% y == x$		$x < y \text{ and } x >= y$		$x != y \text{ or } x == y$		$\text{not } x > y$		$x <= y \text{ and } x == y \text{ or } x != y$		<p>x,y에 다양한 값을 넣어 연산의 결과를 확인해 보자</p> <p>예제 바꾸기 and or 섞어서 한번에 계산이 나오지 않게 먼저 직접계산해볼 수 있도록 지도</p>
연산식	결과																	
$(x + y) <= (x - y)$																		
$(x * y) != (x / y)$																		
$x \% y == x$																		
$x < y \text{ and } x >= y$																		
$x != y \text{ or } x == y$																		
$\text{not } x > y$																		
$x <= y \text{ and } x == y \text{ or } x != y$																		



### 3-4. 표준 입출력

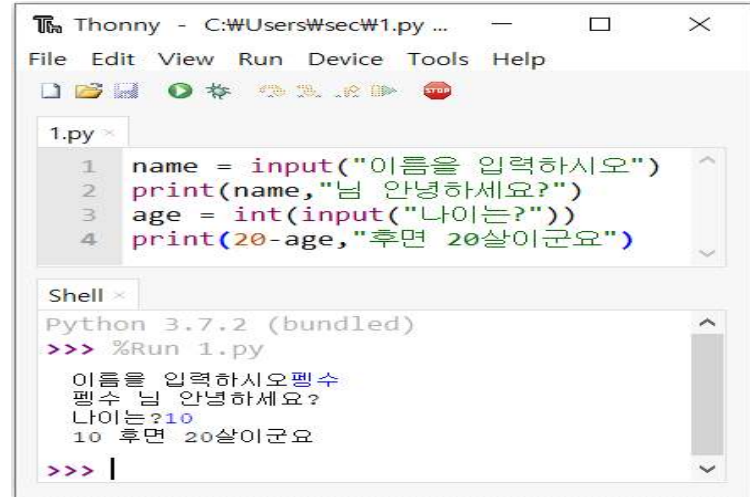
input()	print()
- _____	- _____
- _____	- _____
- _____	- _____
- _____	- _____
- _____	- _____

step1>		Thonny 실행
step2>	<p>* a와 b의 값을 출력하기</p> <p>* 원의 둘레 계산하기(input())</p>	<p>input/print 명령어를 입력하고 컴파일하여 결과를 확인한다.</p>

미션

프로그램의 빈칸을 완성하여 shell 창의 실행결과를 만드시오.

나이계산 프로그램



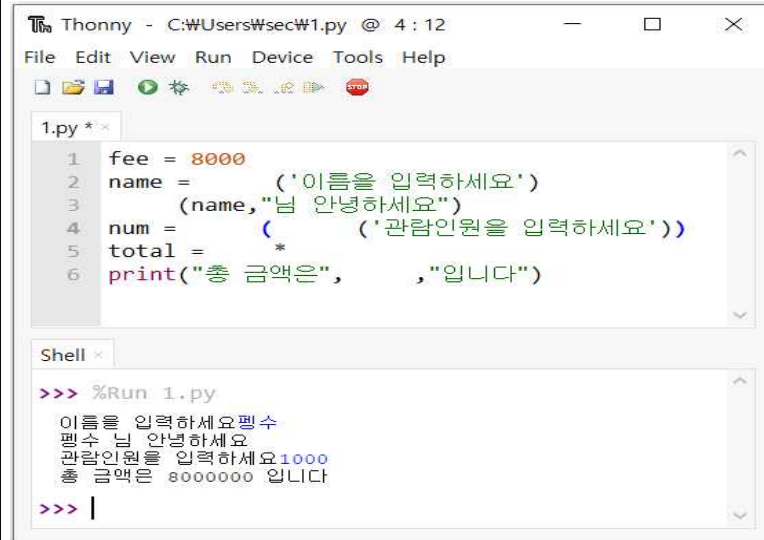
```

Thonny - C:\Users\secw1.py ...
File Edit View Run Device Tools Help

1.py
1 name = input("이름을 입력하십시오")
2 print(name, "님 안녕하세요?")
3 age = int(input("나이는?"))
4 print(20-age, "후면 20살이군요")

Shell
Python 3.7.2 (bundled)
>>> %Run 1.py
이름을 입력하십시오: 팽수
팽수 님 안녕하세요?
나이는? 10
10 후면 20살이군요
>>> |
  
```

총금액 계산하기



```

Thonny - C:\Users\secw1.py @ 4:12
File Edit View Run Device Tools Help

1.py
1 fee = 8000
2 name = ('이름을 입력하세요')
3 (name, "님 안녕하세요")
4 num = ('관람인원을 입력하세요')
5 total = *
6 print("총 금액은", , "입니다")

Shell
>>> %Run 1.py
이름을 입력하세요: 팽수
팽수 님 안녕하세요
관람인원을 입력하세요: 1000
총 금액은 8000000 입니다
>>> |
  
```

\* 자료를 입력(input)받아 처리한 결과를 보여(print)주는 프로그램을 완성해보자

탐구과제 1.

[문제정의]

[해결방법]

[순서도/프로그램]

탐구과제 2.

[문제정의]

[해결방법]

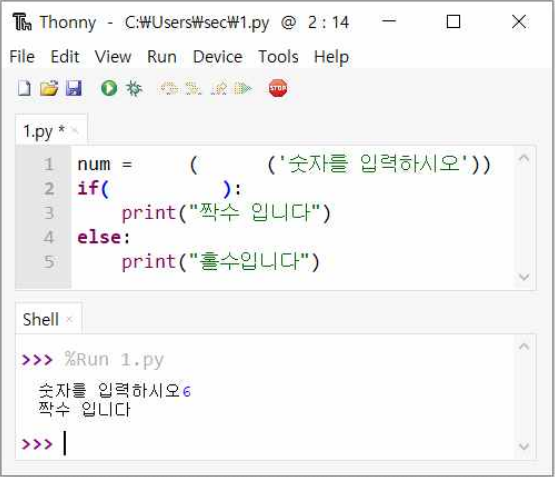
[순서도/프로그램]

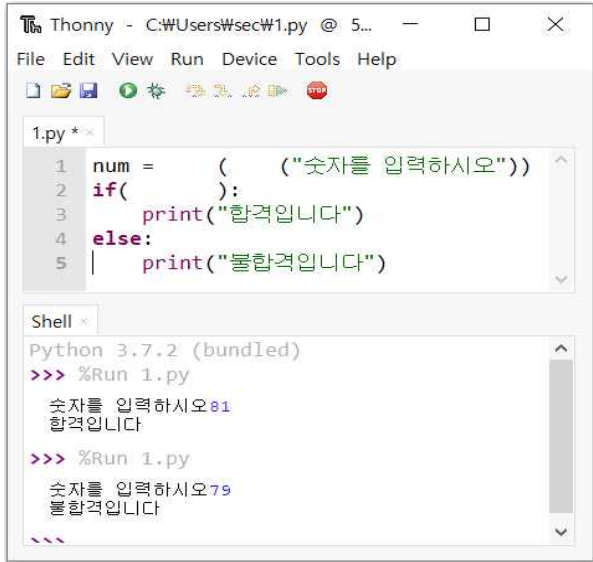
### 3-5. 제어구조의 활용(선택/판단문)

- 제어구조 : 명령어를 실행하는 순서를 결정하는 구조(순차, 선택, 반복)
- 선택구조

if~else (둘중 하나 선택)	if~elif~else(셋중 하나 선택)
_____	_____
_____	_____
_____	_____
_____	_____

- 짝 · 홀수 판별 프로그램을 빈칸을 채워 완성해보자

프로그램	순서도
	

미션	입력된 점수가 80점이상이면 합격 나머지는 불합격을 판정하는 프로그램을 완성하시오.
프로그램과 실행결과	
순서도	

- if~elif~else를 이용해서 프로그램을 완성하시오.

실습	입력된 수가 '양수', '음수' 혹은 '0' 인지 판별하는 프로그램을 완성하시오
프로그램	순서도
<div><div>Thonny - C:\Users\sec#\1.py @ 4... File Edit View Run Device Tools Help</div><div>1.py * 1 num = ( ("숫자를 입력하시오")) 2 if( ): 3 print("양수입니다") 4 ( ): 5 print("음수입니다") 6 else: 7 print("0입니다")</div><div>Shell &gt;&gt;&gt; %Run 1.py 숫자를 입력하시오5 양수입니다 &gt;&gt;&gt; %Run 1.py 숫자를 입력하시오-9 음수입니다 &gt;&gt;&gt; %Run 1.py 숫자를 입력하시오0 0입니다</div></div>	

미션	입력된 점수가 90점 이상이면 '매우우수', 70점 이상이면 '우수', 나머지는 '노력요함' 을 출력하는 프로그램을 완성하시오.
프로그램	순서도

\* 자료를 입력(input)받아 조건(if)에 맞게 판단하고 결과를 보여(print)주는 프로그램을 완성해보자

탐구과제 1.

[문제정의]

[해결방법]

[순서도/프로그램]

탐구과제 2.

[문제정의]

[해결방법]

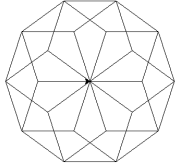
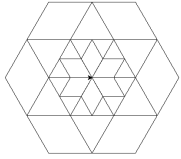
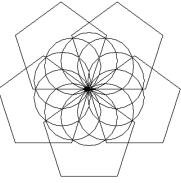
[순서도/프로그램]



■ 중첩구조 - 제어구조 안에 또 다른 제어구조

step1>		
step2>	<pre> 1 from turtle import * 2 3 forward(100) 4 left(120) 5 forward(100) 6 left(120) 7 forward(100) 8 left(120) </pre>	<p>철자에 주의하여 입력 후 컴파일한다.</p>
step3>	<pre> 1 from turtle import * 2 3 for i in range(3): 4     forward(100) 5     left(120) </pre>	<ul style="list-style-type: none"> <li>for문</li> <li>들여쓰기 주의</li> <li>속도조절 speed(0)</li> </ul>
step4>	<pre> 1 from turtle import * 2 3 for j in range(4): 4     for i in range(3): 5         forward(100) 6         left(120) 7     left(90) </pre>	<ul style="list-style-type: none"> <li>for문을 2개 사용해서 그려보세요</li> <li>중첩 for문 사용시 들여쓰기 주의</li> </ul>

■ 각도와 횟수를 달리하여 다양한 도형을 그려보자.

step3>		 <p>5.10</p>
step4>		 <p>6.6 * 2</p>
step5>		 <p>5.5 원.10</p>

\* 실습 1. 태극모양을 그리는 프로그램

<프로그램>

\* 나만의 패턴 Graphic Art 그려보자

<프로그램>

[나만의 패턴]

\* 1단부터 9단까지 구구단 출력하기에 도전하기



\* 실생활 속 문제를 해결할 수 있는 나만의 프로그램을 완성해보자!

[문제정의]

[해결방법]

[프로그램]

\* 실습 2. UpDown 게임만들기

〈순서도〉

〈프로그램〉

\* 실습 3. 입력된 10개의 수 중 최대값을 구하시오.

<실행결과>

숫자를 입력하시오5  
숫자를 입력하시오6  
숫자를 입력하시오9  
숫자를 입력하시오8  
숫자를 입력하시오5  
숫자를 입력하시오6  
숫자를 입력하시오3  
숫자를 입력하시오2  
숫자를 입력하시오1  
숫자를 입력하시오4  
최대값은 9

<프로그램>

<순서도>

실습 4. 입력된 수의 약수를 모두 출력하시오.(심화학습)

<실행결과>

숫자를 입력하시오 14

14 의 약수 1

14 의 약수 2

14 의 약수 7

14 의 약수 14

14 의 약수의 갯수는 4 입니다

숫자를 입력하시오 13

13 의 약수 1

13 의 약수 13

13 는 소수입니다

<프로그램>

<순서도>

### 3-7. 배열의 활용

배열 : 같은 성격의 데이터를 연속적인 공간에 저장

\* 초기값 부여

배열명 = [ 값, 값, 값,.....]

예) date = [ '월', '화', '수', '목', '금', '토', '일' ]

\* 배열 자료 접근 : 배열명[참자]

예) date[0]

### 실습 6. 가위, 바위, 보 게임만들기

<실행결과>

```
>>> %Run 928.py
```

```
가위바위보 게임을 시작합니다
1:가위, 2:바위, 3:보 당신의 선택은?: 3
컴퓨터는: 보 당신은: 보
비겼습니다
```

```
>>> %Run 928.py
```

```
가위바위보 게임을 시작합니다
1:가위, 2:바위, 3:보 당신의 선택은?: 1
컴퓨터는: 보 당신은: 가위
```



<프로그램 및 순서도>

실습 5. 입력된 10개의 수를 버블정렬로 오름차순정렬하시오.

<실행결과>

1 번째 정렬 후 결과	[5, 7, 6, 4, 9]
2 번째 정렬 후 결과	[5, 6, 4, 7, 9]
3 번째 정렬 후 결과	[5, 4, 6, 7, 9]
4 번째 정렬 후 결과	[4, 5, 6, 7, 9]
최종 버블정렬 결과	[4, 5, 6, 7, 9]

<순서도 및 프로그램>

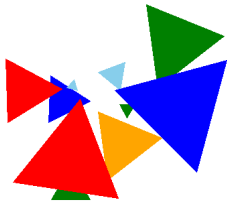
### 3-8. 함수의 활용

함수 :

정의된 기능만을 전담하여 처리하는 독립된 작은 부속 프로그램

선언방법	호출방법
<pre>def 함수명(매개 변수):     명령문1     ...     명령문K     return 반환값</pre>	함수명(인수)
<pre>def tri(k):     for i in range(3):         fd(k)         lt(120)     return</pre>	tri(100)

\* 삼각형을 색상, 위치, 크기를 랜덤하게 10개를 그려보려고 한다. 함수를 활용해서 문제를 해결해 보자.



<순서도 및 프로그램>

### 3-9. 프로그래밍 응용

### 〈아이디어 스케치〉

무엇이든 좋아요!  
여러분이 생각한 걸 컴퓨터에게 시켜보세요.  
알아듣게 코딩만 해준다면 컴퓨터는 여러분의 든든한 지원군이 되어 줄  
겁니다.

주제 :

기획의도: