

파이썬 프로그래밍 입문

계산

최 윤 정

cris.lecture@gmail.com

오늘은

□ 사칙연산 수준에서 다양한 계산이 필요한 일들을 해봅니다.

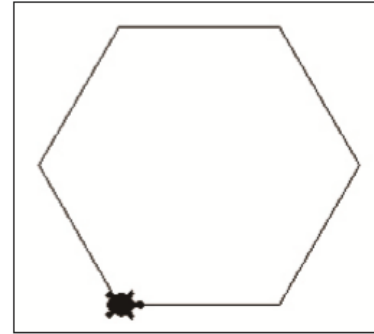
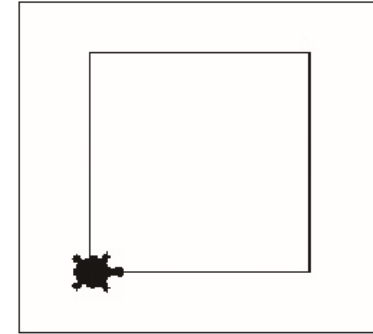
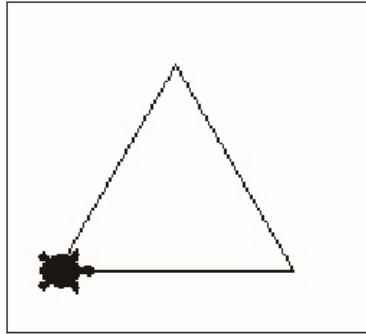
- 연산자와 피연산자
- 연산자의 우선순위와 () 의 사용

□ 실습

- 단위변환기
- 거스름돈 계산하기
- 거북이로 그리기 : 또 다른 방법으로 다각형 그리기

실습미리보기 : 하나의 프로그램으로

- 그리고 싶은 다각형을 입력하세요 : 3
- 그리고 싶은 다각형을 입력하세요 : 4
- 그리고 싶은 다각형을 입력하세요 : 6



아메리카노 개수: 1
카페라떼 개수: 2
카푸치노 개수: 3
총 18500 원 입니다.

받은 돈: 21000
물건값: 18500
거스름돈: 2600
1000원 개수 : 2
500원 개수: 1
100원 개수: 1

수식은 어디에나 있다.

□ 컴퓨터의 계산기능으로 이루어 지는 것들

- 컴퓨터비전, 영상처리, 그래픽스
- 우리가 즐겨보는 영화의 컴퓨터 그래픽 장면들
- 물리학의 여러 가지 공식들을 이용하여 컴퓨터로 계산한 결과를 화면에 표시합니다.



출처: 영화 어벤저스

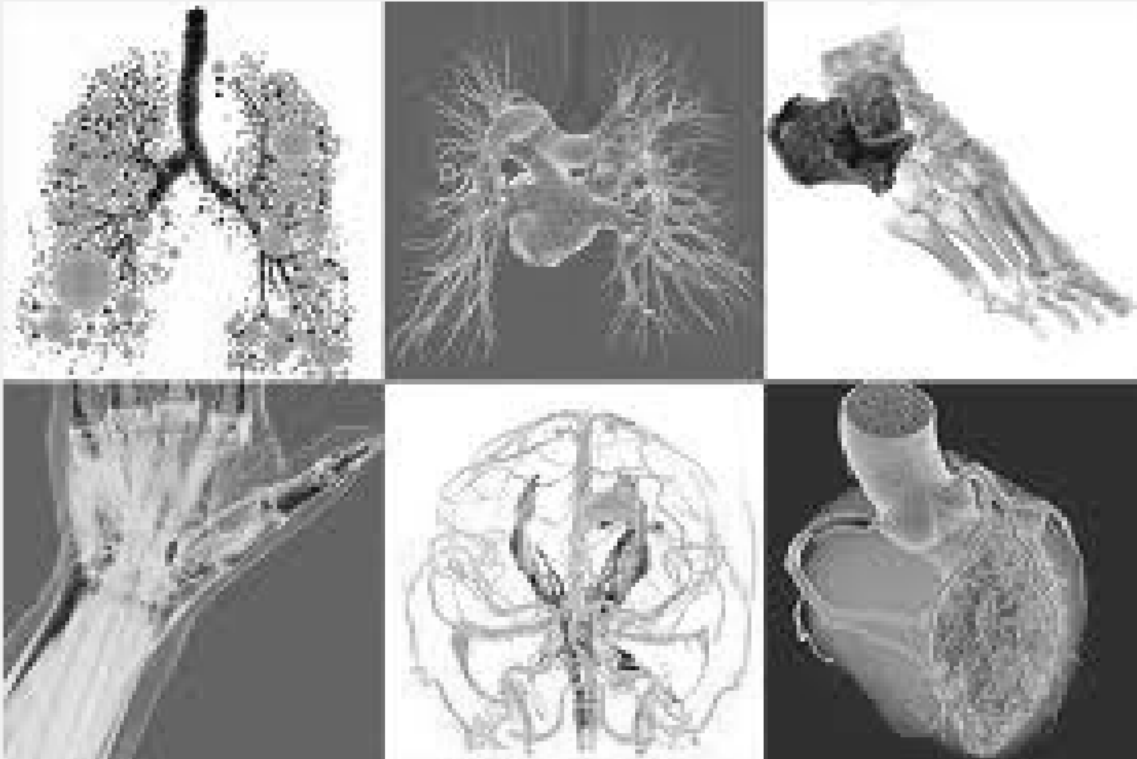
기본 수식(?)으로 가능한 처리

이미지읽기

Digital Image Processing
for Medical Applications

Picture3

Extend



Picture List

Picture 3

Color

- ☐ Normal
☒ GrayScale
☐ Negative

Bright

- +

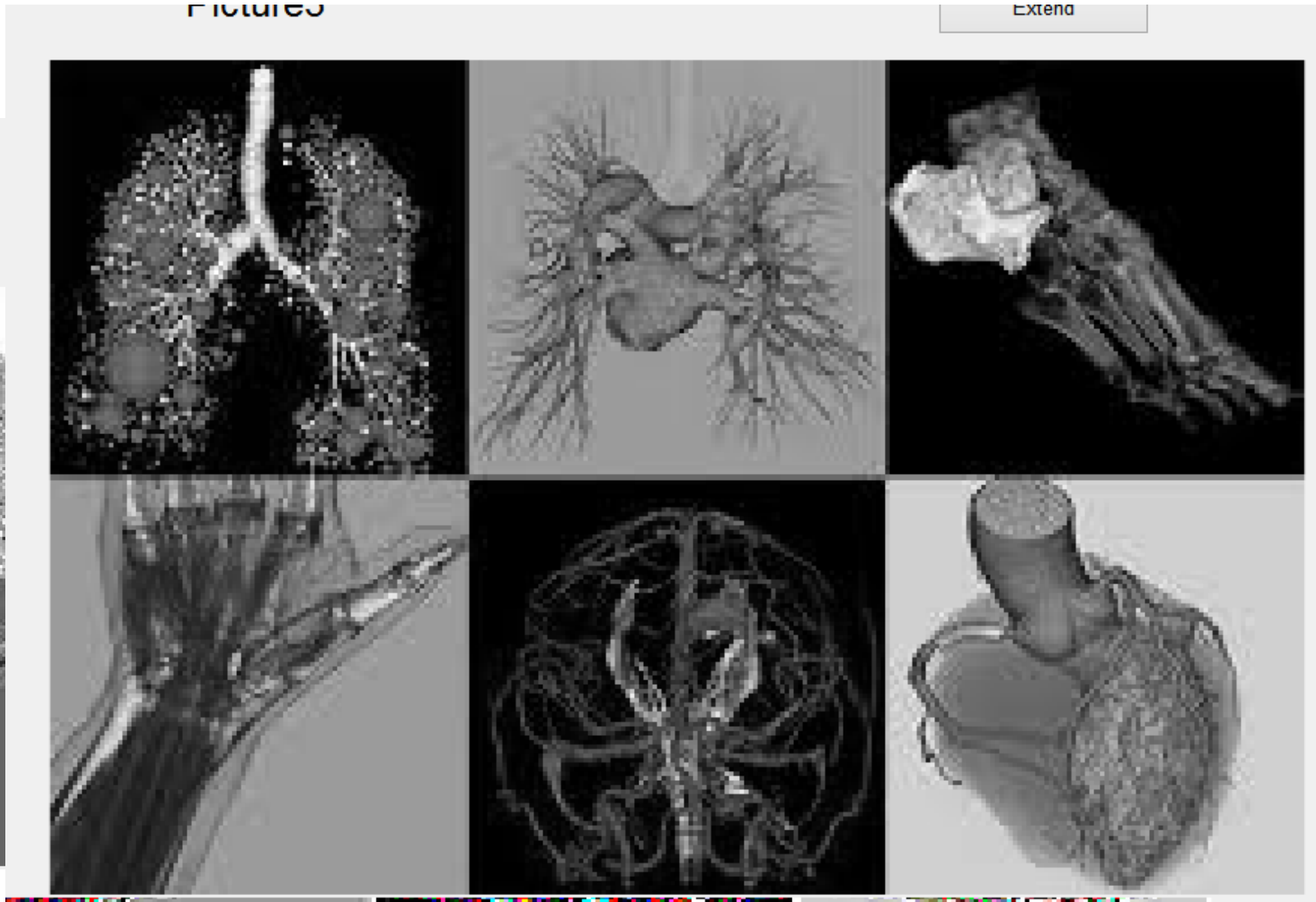
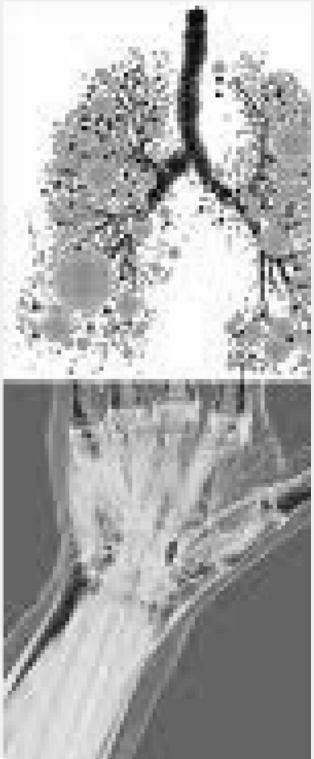
Effects

- ☐ Blurring
☐ Sharpening
☐ Embossing
☐ Edge Detect
☐ Thinning

기본 수식(?)으로 가능한 처리

이미지읽기

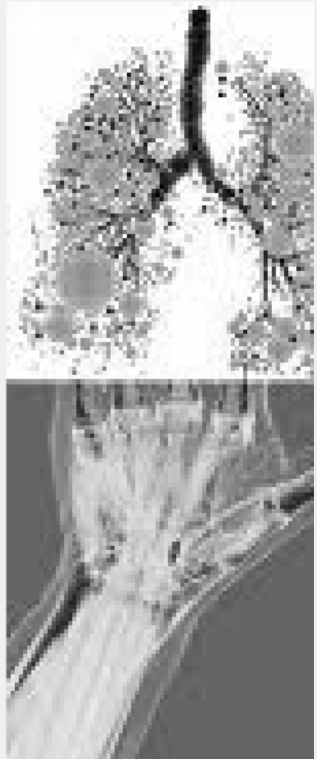
Picture3



기본 수식(?)으로 가능한 처리

이미지읽기

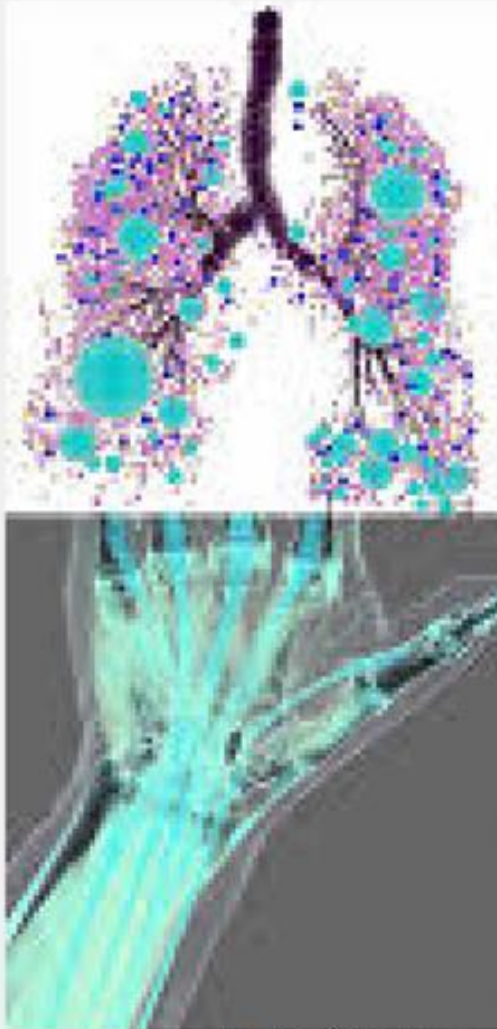
Picture3



Picture3

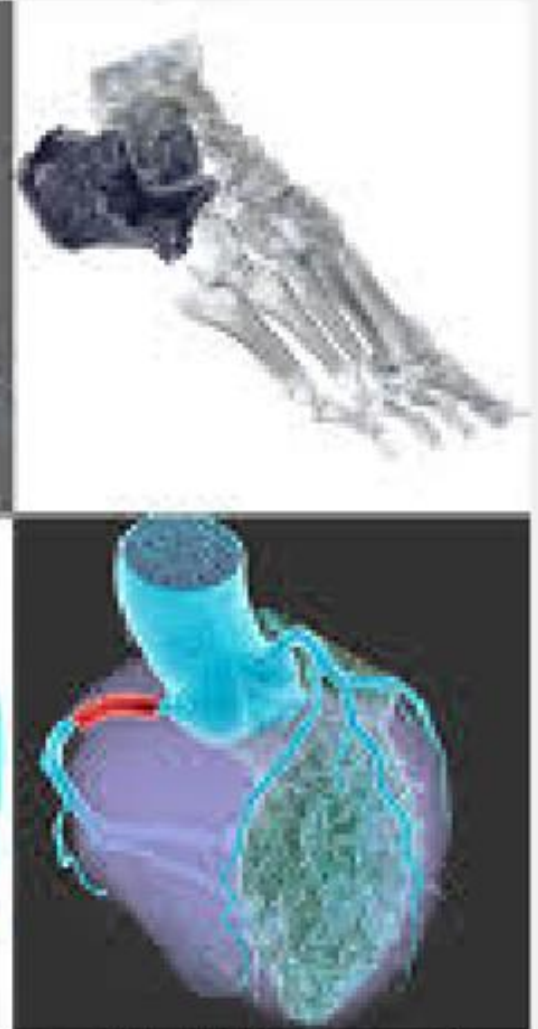
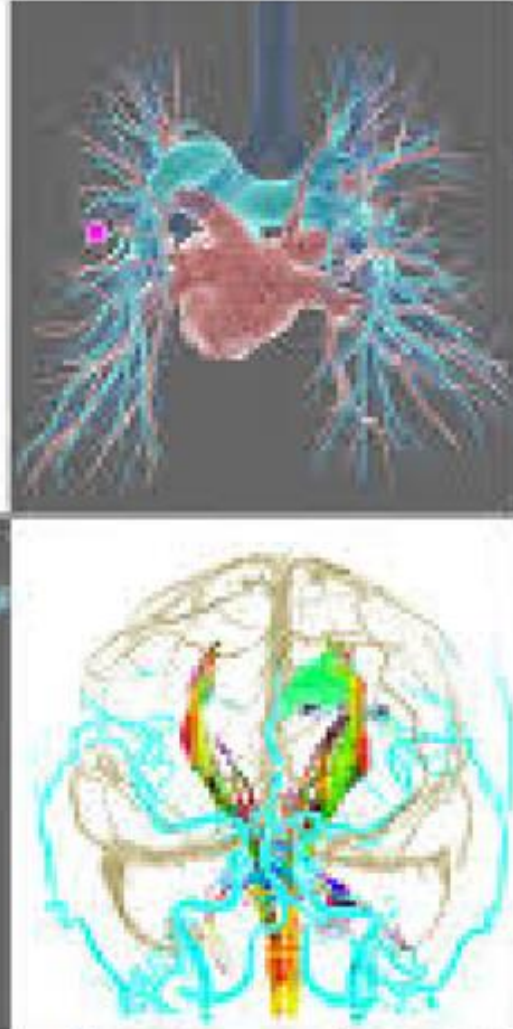


Picture3



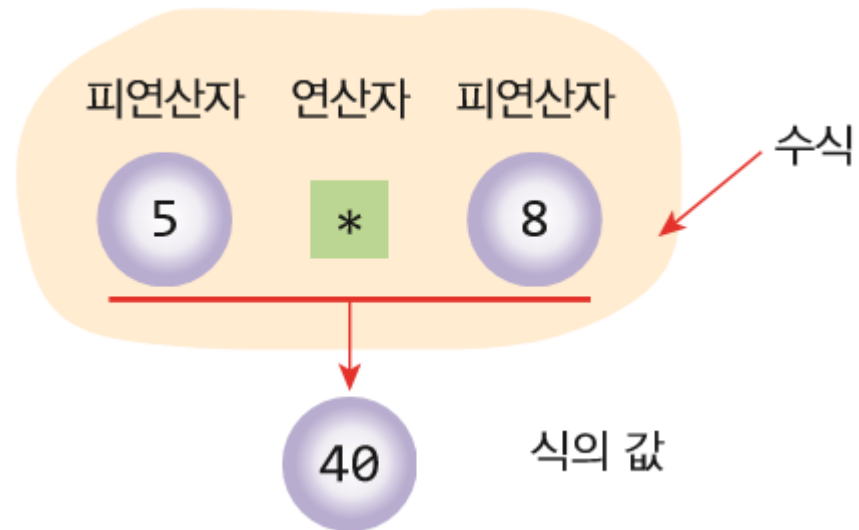
Extend

Extend



연산자와 피연산자

- 연산자(operator): 연산을 나타내는 기호
- 피연산자(operand): 연산의 대상이 되는 값
- 수식(expression) : 피연산자들과 연산자의 조합



산술 연산자

□ 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 연산

- 나눗셈 결과 : /
- 나눗셈의 몫 : //
- 나눗셈의 나머지 : %
- 지수 계산은 ** : $2^{**}3 = 8$

연산자	기호	사용례	결괏값
덧셈	+	$7 + 4$	11
뺄셈	-	$7 - 4$	3
곱셈	*	$7 * 4$	28
나눗셈	//	$7 // 4$	1
나눗셈	/	$7 / 4$	1.75
나머지	%	$7 \% 4$	3

나눗셈과 나머지 연산자

```
>>> 7 / 4 # 나눗셈결과
```

```
1.75
```

```
>>> 7 // 4 #몫
```

```
1
```

```
>>> 7%4 #나머지
```

```
3
```

문자열 입력받기

```
name = input()
```

숫자 입력받기

```
number = int(input()) #정수(소수점이 없음)
```

```
height = float(input()) #실수(소수점이 있음)
```

```
p = int(input("분자를 입력하시오: "))
```

```
q = int(input("분모를 입력하시오: "))
```

```
print("나눗셈의 몫=", p // q)
```

```
print("나눗셈의 나머지=", p % q)
```

```
분자를 입력하시오: 7
```

```
분모를 입력하시오: 4
```

```
나눗셈의 몫= 1
```

```
나눗셈의 나머지= 3
```

나머지 연산자의 용도

□ 짝수와 홀수의 구분

```
number = int(input("정수를 입력하시오: "))  
#짝수이면 0, 홀수이면 1을 출력한다  
print(number%2)
```

정수를 입력하시오: 28

0

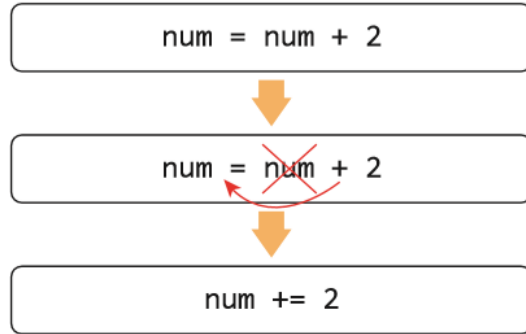
□ 초단위의 시간을 받아서 몇 분 몇 초인지를 계산하여 보자.

```
sec = 1000  
min = sec // 60  
remainder = sec % 60  
print(min, "분", remainder, "초")
```

16 40

복합 연산자

- 복합 연산자(compound operator)란 +=처럼 대입 연산자와 다른 연산자를 합쳐 놓은 연산자이다.



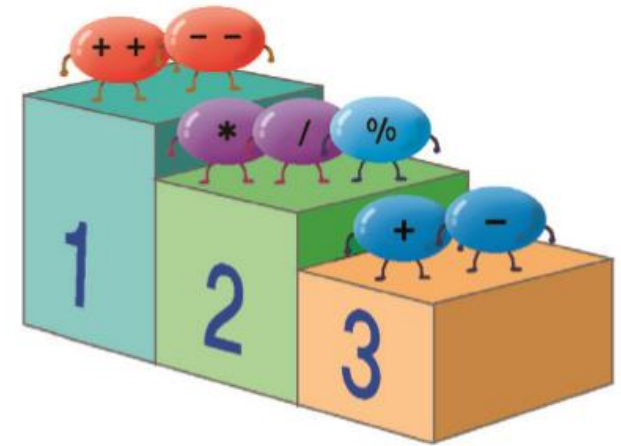
복합 연산자	의미
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x \%= y$	$x = x \% y$

```
x = 1000
print("초깃값 x=", x)
x += 2
print("x += 2 후의 x=", x)
x -= 2
print("x -= 2 후의 x=", x)
```

```
초깃값 x= 1000
x += 2 후의 x= 1002
x -= 2 후의 x= 1000
```

연산자 우선 순위표

순위	연산자	설명
1	**	지수 연산자
2	~ + -	단항 연산자
3	* / % //	곱셈, 나눗셈, 나머지 연산자
4	+ -	덧셈, 뺄셈
5	>> <<	비트 이동 연산자
6	&	비트 AND 연산자
7	^	비트 XOR 연산자, 비트 OR 연산자



여러가지 연산자가 쓰일 경우,

우선순위에 따라, 읽는 방향에 따라 계산된다.!

지수연산이 사용되면 ←, 그 외에는 → 방향으로 읽어가면서 우선순위를 적용한다.

연산자 우선 순위표

- 우선순위는 외우지 말고 , 먼저 계산할 곳에 ()를 사용합시다.

```
>>> 10 + 20 / 2
20.0

>>> (10 + 20) / 2
15.0
```

$$x + \underbrace{y * z}_{\textcircled{1}}$$

$\underbrace{\hspace{10em}}_{\textcircled{2}}$

$$\underbrace{(x + y)}_{\textcircled{1}} * z$$

$\underbrace{\hspace{10em}}_{\textcircled{2}}$

잠깐 Quiz : 평균 구하기

□ 잘못된 부분은 어디일까?

```
x = int(input("첫 번째 수를 입력하시오: "))  
y = int(input("두 번째 수를 입력하시오: "))  
z = int(input("세 번째 수를 입력하시오: "))  
avg = x + y + z / 3  
print("평균 =", avg)
```

첫 번째 수를 입력하시오: 10
두 번째 수를 입력하시오: 20
세 번째 수를 입력하시오: 30
평균 = 40.0



Solution

```
x = int(input("첫 번째 수를 입력하시오: "))  
y = int(input("두 번째 수를 입력하시오: "))  
z = int(input("세 번째 수를 입력하시오: "))  
avg = (x + y + z) / 3  
print("평균 =", avg)
```

첫 번째 수를 입력하시오: 10
두 번째 수를 입력하시오: 20
세 번째 수를 입력하시오: 30
평균 = 20.0

잠깐 Quiz : *과 **

- 계산 결과는 무엇일까요?
- 지수 표현(**)이 있는 수식은 <--- 방향으로 읽어오면서 우선순위와 ()에 따라 계산된다.
 - 먼저 계산할 곳에 ()를 씁니다!

$a=2^{**3^{**2}}$ # $2^{**}(3^{**2})$ 와 같음

$b=(2^{**3})^{**2}$

$c=2*1^{**3}$

$d=(2*1)^{**3}$

$e = 6 / 2 * 3$

$f = 6 / 2^{**3}$



Solution

main.py

```
1 a=2**3**2 #2**(3**2)와 같음
2 print(a)
3
4 b=(2**3)**2 #8**2와 같음
5 print(b)
6
7 c=2*1**3 #2*(1**3)과 같음
8 print(c)
9
10 d=(2*1)**3 #2**3과 같음
11 print(d)
12
13 e = 6 / 2 * 3 # 3*3과 같음
14 print(e)
15
16 f = 6 / 2 ** 3 #6/8과 같음
17 print(f)
18
```

```
512
64
2
8
9.0
0.75
```

주석(comment)

- 소스 코드에 붙이는 설명글.
- 주석은 프로그램의 실행 결과에 영향을 끼치지 않는다.
- # 또는 '''(따옴표 3개) 를 사용한다

```
# 1. 사용자로부터 화씨온도를 입력받는다.
```

```
ftemp = int(input("화씨온도: "))
```

```
ctemp = (ftemp-32.0)*5.0/9.0
```

```
print("섭씨온도:", ctemp)
```

```
# 2. 화씨온도->섭씨온도
```

```
# 3. 섭씨온도를 화면에 출력한다
```



Lab : 가벼운 실습

- 화씨->섭씨
- BMI지수 계산하기
- 커피메뉴자판기- 거스름돈 계산하기

미리보기 : Turtle 의 몇가지 함수들과 간단한 반복문 소개

- 다각형 그리기

Lab#1 : 화씨온도<-> 섭씨온도로 변환하기

- 화씨온도를 받아서 섭씨온도로 바꾸는 프로그램을 작성해보자.

$$C = (F - 32) * \frac{5}{9}$$



화씨온도: 100.0

섭씨온도: 37.77777777777778

화씨온도: 50.0

섭씨온도: 10.0

```
ftemp = float(input("화씨온도: "))  
ctemp = (ftemp-32)*5/9 # C나 Java에서는 ctemp = (ftemp-32)*5.0/9.0  
print("섭씨온도:", ctemp)
```

#좀 더 세심한 출력양식

%.2f는 소수 2자리로 출력해라, % (변수이름)

print("섭씨온도: %.2f" %(ctemp)) #소수점 둘째자리까지 반올림된 결과

print(f"섭씨온도: {ctemp:.2f}") ##소수점 둘째자리까지 반올림된 결과

도전!! : 섭씨온도를 입력받아 화씨온도로 바꾸어 보세요

Solution

main.py

```
1 ftemp = float(input("화씨 온도: "))
2 ctemp = (ftemp-32)*5/9
3 # C나 Java에서는 ctemp = (ftemp-32)*5.0/9.0 으로 씁니다!
4 print("섭씨 온도:", ctemp)
5 print(f"섭씨 온도: {ctemp:.2f}")
6 print("섭씨 온도: %.2f" %(ctemp) )
7
8 print("--"*10)
9
10 ctemp = float(input("섭씨 온도: "))
11 ftemp = ctemp*9/5 + 32
12 print("화씨 온도:", ftemp)
13 print(f"화씨 온도: {ftemp:.2f}")
14 print("화씨 온도: %.2f" %(ftemp) )
15
```

화씨 온도 : 50.0

섭씨 온도 : 10.0

섭씨 온도 : 10.00

섭씨 온도 : 10.00

섭씨 온도 : 15

화씨 온도 : 59.0

화씨 온도 : 59.00

화씨 온도 : 59.00

Lab#2 : BMI 계산하기



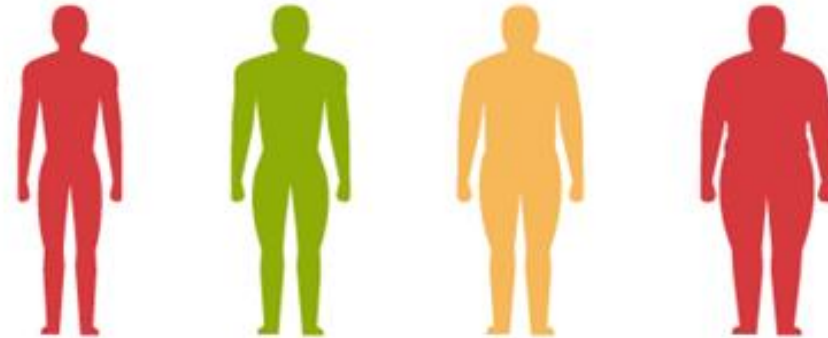
- 신장과 체중을 입력받아서 BMI 값을 출력하는 프로그램을 작성합니다.

몸무게를 kg 단위로 입력하시오: 100

키를 미터 단위로 입력하시오: 1.87

당신의 BMI= 28.59675712774171

BMI 값 : 28.60



BMI Chart

$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$

BMI less than 18.50	Underweight
BMI 18.50 - 24.99	Healthy weight
BMI 25.00 - 29.99	Overweight
BMI 30 or more	Obese

Solution

```
weight = float(input("몸무게를 kg 단위로 입력하시오: "))  
height = float(input("키를 미터 단위로 입력하시오: "))  
  
bmi = (weight / (height**2))  
print("당신의 BMI=", bmi)  
  
# %.2f는 반올림 후 소수 2자리로 출력하라는 의미. % ( 변수이름)  
# 둘 중에 편한 방법을 사용합니다.  
  
print(" BMI 값 : %.2f " %(bmi) )  
print(f" BMI 값 : {bmi : .2f} ")
```

Lab#3 : 거스름돈 계산하기



- 자동 판매기는 사용자로부터 투입한 돈과 물건값을 입력받는다.
- 물건값은 100원 단위.
- 자판기는 동전 500원, 100원짜리만 가지고 있다고 가정하자.
- 프로그램은 잔돈을 계산하여 출력한다.
- % 연산자를 활용하세요

투입한 돈: 5000

물건값: 2600

거스름돈: 2400

500원 동전의 개수: 4

100원 동전의 개수: 4

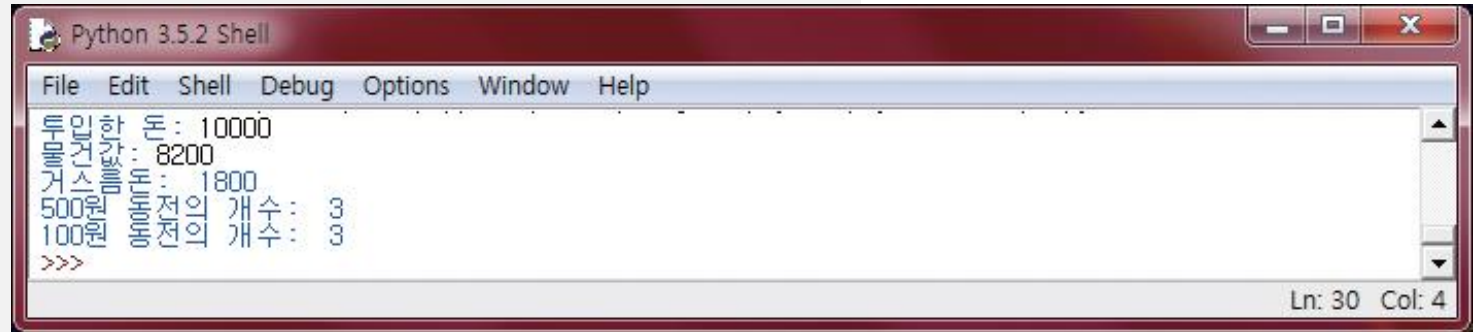
Solution

```
money = int(input("투입한 돈: "))  
price = int(input("물건 값: "))
```

```
change = money-price  
print("거스름돈: ", change)
```

```
coin500s = change // 500      # 500으로 나누어서 몫이 500원짜리의 개수  
change = change % 500        # 500으로 나눈 나머지를 계산한다.  
coin100s = change // 100     # 100으로 나누어서 몫이 100원짜리의 개수
```

```
print("500원 동전의 개수: ", coin500s)  
print("100원 동전의 개수: ", coin100s)
```



```
Python 3.5.2 Shell  
File Edit Shell Debug Options Window Help  
투입한 돈: 10000  
물건값: 8200  
거스름돈: 1800  
500원 동전의 개수: 3  
100원 동전의 개수: 3  
>>>  
Ln: 30 Col: 4
```

Lab #4 : 커피주문기

- 메뉴대로 개수 입력를 입력받고 가격을 계산한 후
- 총 가격을 표시한다



아메리카노 개수: 1
카페라떼 개수: 2
카푸치노 개수: 3
총 18500 원 입니다.

Solution

#가격 세팅 : 메뉴와 가격은 자유입니다.

americano_price = 2000

caffelatte_price = 3000

cappuccino_price = 3500

total_price=0;

#각 메뉴 별로 몇 잔인지 입력받기

americanos = _____;

caffelattes = _____;

cappuccinos = _____;

#메뉴 개수 * 메뉴 가격을 구하여 total_price에 합산하기

total_price = _____ #아메리카노 가격계산

total_price += _____ #카페라떼 가격을 계산하여 합산

total_price += _____ #카푸치노 가격을 계산하여 합산

print("총 ", price, "원 입니다.")



배열과 비교문/반복문을 배워가면서
좀 더 멋지게 바꾸어 보겠습니다.

Solution

```
americano_price = 2000
caffelatte_price = 3000
cappuccino_price = 3500

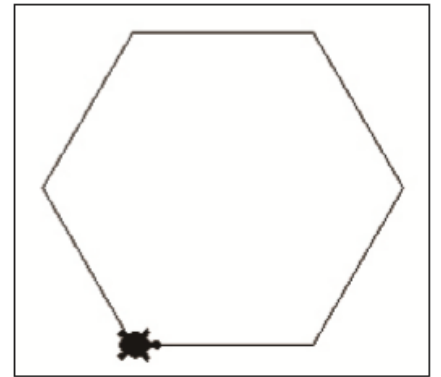
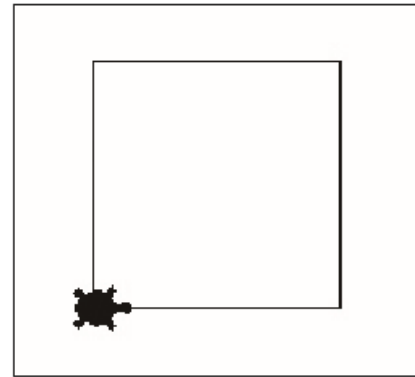
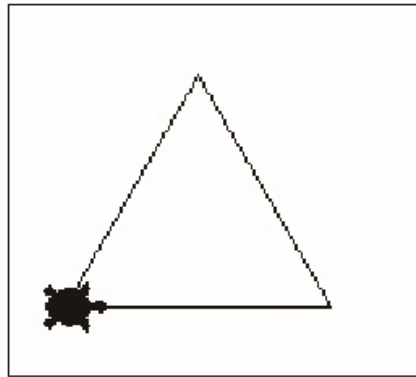
americanos = int(input("아메리카노 개수: "))
caffelattes = int(input("카페라떼 개수: "))
cappuccinos = int(input("카푸치노 개수: "))
price=0
price = americanos*americano_price
price += caffelattes*caffelatte_price
price += cappuccinos*cappuccino_price
print("총 ", price, "원 입니다.")
```



배열과 비교문/반복문을 배워가면서
좀 더 멋지게 바꾸어 보겠습니다.

Turtle 실습참고

- `import turtle`
- `t=turtle.Turtle()`
- `t.shape("turtle")`
- `t.forward(width) , t.fd(100)`
- `t.left(각도), t.right(각도)`
- 안보이면 `t.showturtle()`
- 화면을 지울때는 `t.clear()`
- 처음 나타나는 위치좌표는 (0,0), `t.goto(0,0)` 으로 처음 좌표로 이동
- 연결하지 않고 그릴 때는 `t.penup()` , 다시 그릴 때는 `t.pendown()`도 있어요
- `t.write("msg")` 해당 위치에 글씨출력
- 거북이 화면이 사라지는 것을 방지하려면 마지막줄에 `input()` 을 추가하세요



Lab#5 : 변수와 반복문을 사용한 다각형 그리기

몇 각형을 그리시겠어요?(3-6): 6

>>>

#간단한 반복문 구조

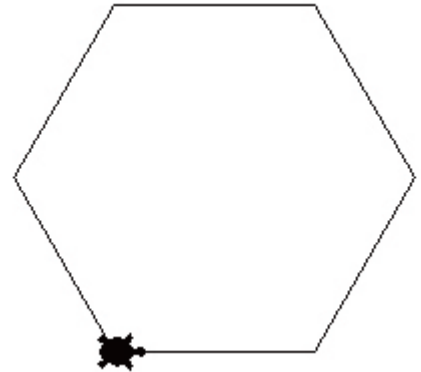
#변수 i를 사용하여 6번의 반복을 수행한다.

#for 범위내에서 i는 자동으로 1씩 증가한다.

for i in range(6) : # 6번 반복 ,변수를 사용하여 조정

 t.forward(100) # 반복하는 문장은 들여쓰기

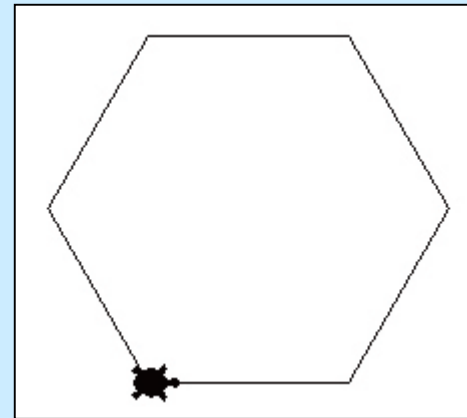
 t.left(60) # 변수를 사용하여 각도 조정.



반복은 아직 학습하지 않았지만 간단히 사용해봅시다.
들여쓰기에 주의하세요

Solution

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
n = int(input("몇 각형을 그리시겠어요?(3-6): "))  
  
for i in range(n):  
    t.forward(100)  
    t.left(360//n)  
print("end")
```



이번 장 정리

- 수식은 피연산자와 연산자로 이루어진다.
- 덧셈, 뺄셈, 곱셈, 나눗셈을 위하여 +, -, *, / 기호를 사용한다.
- 지수 연산자는 **이며 우선순위가 가장 높다!
- 나눗셈에서 몫을 계산하려면 // 연산자를 사용한다.
- 나눗셈에서 나머지를 계산하려면 % 연산자를 사용한다.
- *와 /가 +와 -보다 우선순위가 높다.
- +, *는 피연산자가 모두 숫자일 경우 수식연산을 수행하지만, 문자와 같이 사용되면 이어붙인다.
 - `a = "aaa"+"bbb"` #a에는 "aaabbb"가 저장된다.
 - `b = "aa!" * 3` #b에는 "aa!aa!aa!"가 저장된다.
- 연산자의 우선 순서를 변경하려면 괄호를 사용한다.

수고하셨습니다.