

파이썬 기초

04 연산자



홍필두 교수
(파이썬기초)



학습내용

- 01 대입 연산자
- 02 산술 연산자
- 03 타입 캐스팅





학습목표

- 연산자에서 대입 연산자를 이해하고 사용할 수 있다.
- 연산자에서 산술 연산자를 이해하고 사용할 수 있다.
- 서로 자료형이 다를 경우 타입캐스팅을 통하여 자료형을 변환할 수 있다.



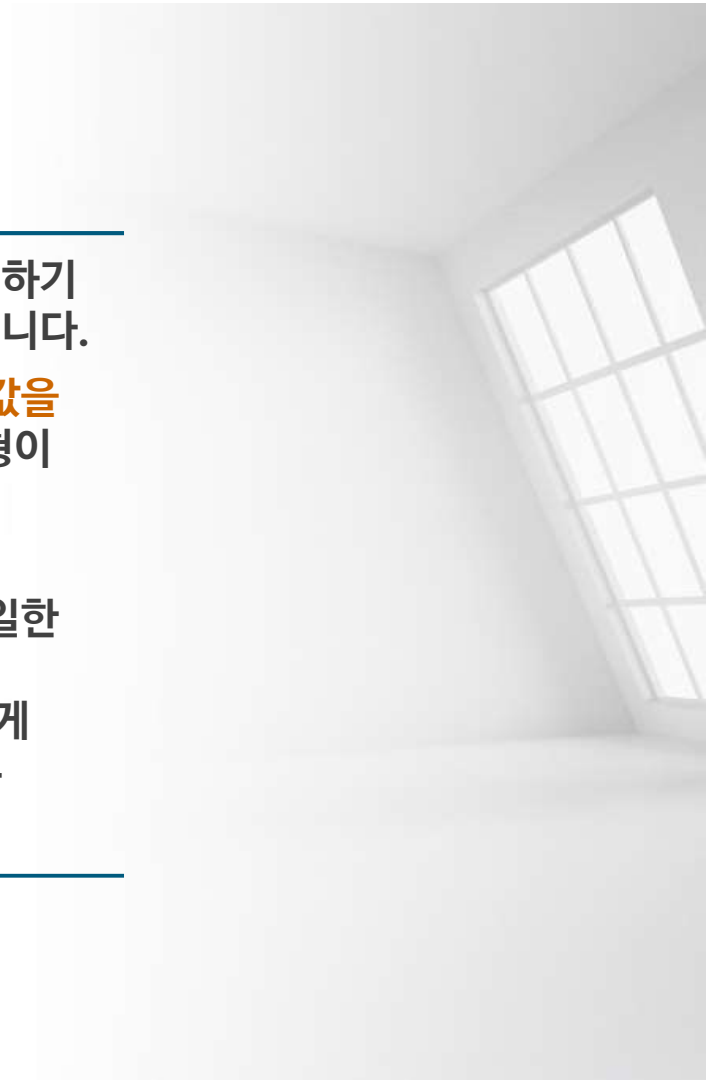
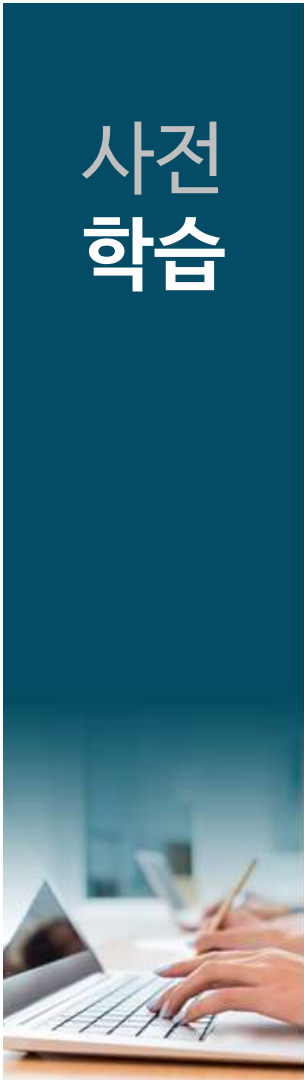
사전 학습

“프로그램에서 연산처리”

프로그래밍의 주요 처리 방법에서 연산을 처리하기 위하여 연산자를 사용하는 것은 중요한 사항입니다.

연산은 우리가 알고 있는 수 **치연산** 및 **변수에 값을 대입하는 연산**이 있습니다. 또한 변수의 자료형이 다른 경우 값을 대입하기 위한 **타입 캐스팅**이 있습니다.

일반적인 프로그래밍 언어 C, java 에서도 동일한 연산이 사용됩니다. 여러분은 파이썬과 다른 프로그래밍 언어에서 연산의 처리방법이 어떻게 이루어 지는지 검색을 통하여 미리 알아보도록 합시다.





01

대입 연산자

- 1) 단순 대입 연산자
- 2) 복합형태 대입 연산자

1) 단순 대입 연산자

단순 대입 연산자

변수의 값을 집어넣는 형태

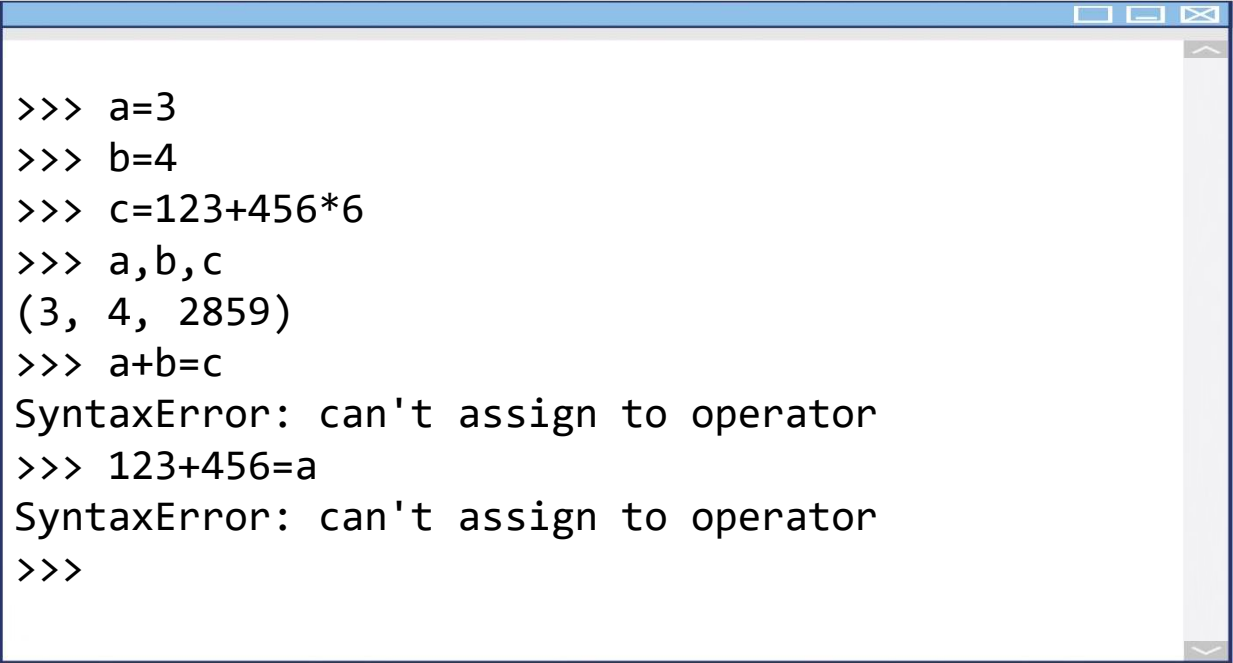


변수 = 값, 변수 = 수식의 형태로 사용함



수식 = 변수는 불가능함

1) 단순 대입 연산자



```
>>> a=3
>>> b=4
>>> c=123+456*6
>>> a,b,c
(3, 4, 2859)
>>> a+b=c
SyntaxError: can't assign to operator
>>> 123+456=a
SyntaxError: can't assign to operator
>>>
```

A screenshot of a Python interactive shell window. The window has a blue title bar with standard window controls (minimize, maximize, close). The main area is white with a vertical scrollbar on the right. The text shows a series of commands and their outputs. The first three commands are assignments: `a=3`, `b=4`, and `c=123+456*6`. The fourth command is `a,b,c`, which returns the tuple `(3, 4, 2859)`. The fifth command is `a+b=c`, which results in a `SyntaxError: can't assign to operator`. The sixth command is `123+456=a`, which also results in a `SyntaxError: can't assign to operator`. The prompt `>>>` is shown at the end of the last line.

1) 단순 대입 연산자



변수의 값을 집어넣는 순간 변수의 자료형이 결정됨



자료형이 결정된 이후 자료형 형태로 사용하여야 함



다시 다른 자료형을 대입 시 자료형이 변경됨

1) 단순 대입 연산자

```
>>> a=1
>>> a=a+"안녕" # a는 수치형
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    a=a+"안녕" # a는 수치형
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> a="abc" #a에 다시 문자를 대입하는 순간 a는 문자형이 됨
>>> a=a+"안녕"
>>> a
'abc안녕'
>>>
```

2) 복합형태 대입 연산자



a,b=b,a 로 a,b변수의 값을 서로 바꿀 수 있음

```
>>> a=1
>>> b=2
>>> a,b
(1, 2)
>>> a,b=b,a
>>> a,b
(2, 1)
```

2) 복합형태 대입 연산자



여러 변수의 값을 한번에 대입할 수 있음

```
>>> a,b,c,d,e,f=1,2,"안녕","하세요",3,4  
>>> a,b,c,d,e,f  
(1, 2, '안녕', '하세요', 3, 4)
```

2) 복합형태 대입 연산자

단순 변수는 값을 대입하는 형태로 이해 가능하나,
컬렉션의 경우 같은 변수를 공유하는 것으로
이해하여야 함

※ 주의, 추후 다시 반복하여 나옴

```
>>> a=b=1
>>> a,b
(1, 1)
>>> b=2
>>> a,b
(1, 2)
```

2) 복합형태 대입 연산자

단순 변수는 값을 대입하는 형태로 이해 가능하나,
컬렉션의 경우 같은 변수를 공유하는 것으로
이해하여야 함

※ 주의, 추후 다시 반복하여 나옴

```
>>> a=b=[1,2,3]
>>> a,b
([1, 2, 3], [1, 2, 3])
>>> a[1]=4
>>> a,b
([1, 4, 3], [1, 4, 3])
>>>
```

2) 복합형태 대입 연산자

심화학습



컬렉션에서 값을 복사하는 경우는 아래와 같이 사용

```
>>> a=[1,2,3,4,5]
>>> b=a[:]
>>> a,b
([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
>>> a[2]=5
>>> a,b
([1, 2, 5, 4, 5], [1, 2, 3, 4, 5])
>>>
```

2) 복합형태 대입 연산자

심화학습

파이썬에서
실제 변수에 값을
집어 넣는 경우

값이라는 오브젝트(Object)를
어디에 정의해 놓고 참고해서
사용하는 방식

2) 복합형태 대입 연산자

심화학습

```
>>> import sys
>>> a=123 → a=123 이라고 정의
>>> sys.getrefcount(123) → 123을 어느 공간에 만들어  
6                             놓고 참조하는 방식
>>> b=123; sys.getrefcount(123) → b=123을  
7                             만나면  
>>> c=123; sys.getrefcount(123) 참조횟수가  
8                             증가함
```




02

산술 연산자

- 1) 일반 산술 연산자
- 2) 복합 연산자

1) 일반 산술 연산자



사칙연산 순으로 계산

→ (+, -, *, /, **) → 덧셈, 뺄셈, 곱셈, 나눗셈, 거듭제곱

```
>>> 1+2*3/4
2.5
>>> 2**3
8
>>> 10/3
3.3333333333333335
>>> 1e12*123456789012
1.23456789012e+23
>>> a=1+2j
>>> a+(123+12j)
(124+14j)
>>>
```

1) 일반 산술 연산자

정수형 연산자

//(정수형 나눗셈), %(나머지)

학교에서의 실수연산

초등학교
고학년이상(실수연산)
5 나누기 2는 2.5임

실생활에서의 실수연산

실생활에서는
5를 2로 나누면
몫은 2 나머지는 1이 됨

1) 일반 산술 연산자



실수형 계산이 기본이나 정수형 연산자를 사용하면
정수형 연산임

```
>>> 5/2
2.5
>>> 5//2
2
>>> 5%2
1
>>>
```

2) 복합 연산자

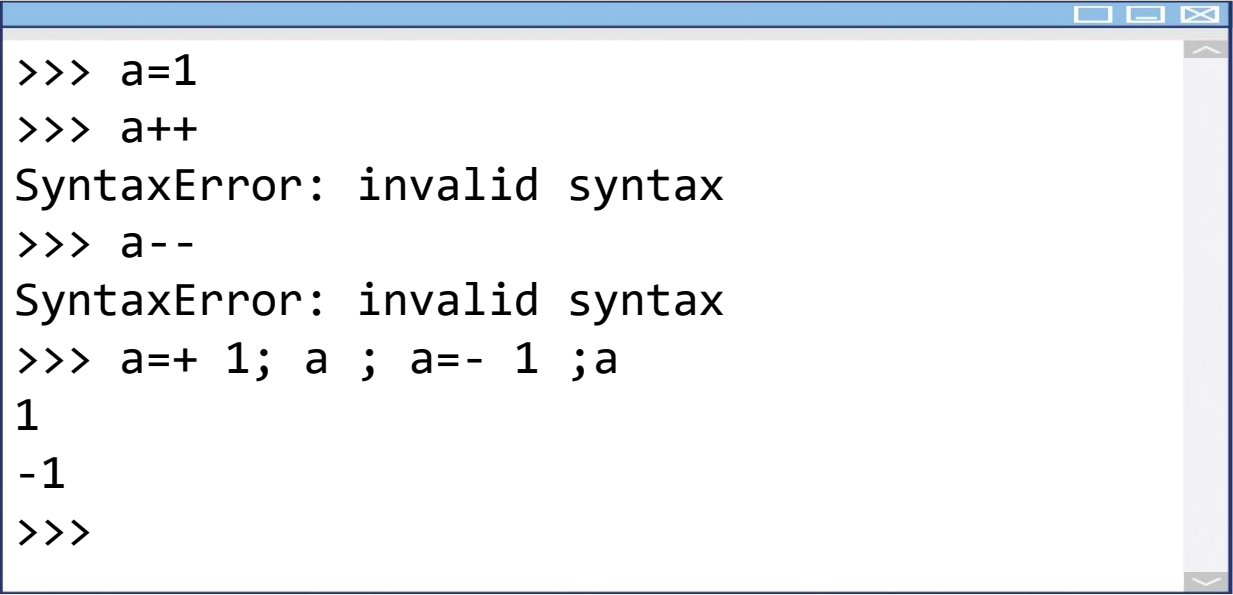
```
>>> a=1;a=a+1;print(a) → 누적 : a= a+1  
2                      → a +=a 로 표시  
>>> a=1;a+=1;print(a)  
2  
>>> a*=2;a → 누곱 : a=a*2 → a *= 2 로 표시  
4  
>>> a*=2;a  
8  
>>> a=1;a-=1;a  
0  
>>> a-=1;a  
-1  
>>>
```

2) 복합 연산자

주의사항

- C, Java에 있는 1씩 더하거나 빼는 $a++$, $a--$ 는 사용할 수 없음
- 순서를 바꾼 $=+$, $=-$ 를 사용하는 경우 양수, 음수를 저장한 경우가 됨(C, Java와 다름)

2) 복합 연산자



```
>>> a=1
>>> a++
SyntaxError: invalid syntax
>>> a--
SyntaxError: invalid syntax
>>> a+= 1; a ; a-= 1 ;a
1
-1
>>>
```



03

타입 케스팅

- 1) 문자형 변환
- 2) 숫자형 변환

1) 문자형 변환



변수의 형태가 다르면 서로 연산처리를 할 수 없음,
이때는 변수의 형 변환(Type casting)처리를 하여야 함



문자형 변수의 덧셈은 문자열을 붙이는 것으로 처리됨

```
>>> a="멍멍이"
>>> b="고양이"
>>> a+b
'멍멍이고양이'
>>> a*3+b*2
'멍멍이멍멍이멍멍이고양이고양이'
>>> "-"*20
'-----'
```

1) 문자형 변환



문자형과 숫자형의 덧셈은 에러이며, 이때 숫자형을 문자형으로 변환하여야 함



숫자형의 문자형으로 변환할 때는 `str()` 함수를 사용함

```
>>> a=1
>>> b="강아지"+a+"마리"
Traceback (most recent call last):
  File "<pyshell#123>", line 1, in <module>
    b="강아지"+a+"마리"
TypeError: can only concatenate str (not "int") to str
>>> b="강아지"+str(a)+"마리"
>>> b
'강아지1마리'
>>>
```

1) 문자형 변환



문자형을 다루는 ord() 함수는 문자의 내부코드값을 반환



chr 함수는 내부코드값을 가지고 해당 글자를 반환

```
>>> ord("가")
44032
>>> hex(ord("a"))
'0x61'
>>> for c in range (ord('a'), ord('z')+1):
        print(hex(c),"=",chr(c)," ",end="")

0x61 = a , 0x62 = b , 0x63 = c , 0x64 = d , 0x65 = e , 0x66 = f , 0x67 = g ,
0x68 = h , 0x69 = i , 0x6a = j , 0x6b = k , 0x6c = l , 0x6d = m , 0x6e = n ,
0x6f = o , 0x70 = p , 0x71 = q , 0x72 = r , 0x73 = s , 0x74 = t , 0x75 = u ,
0x76 = v , 0x77 = w , 0x78 = x , 0x79 = y , 0x7a = z ,
>>>
```

2) 숫자형 변환

 int()



문자형을 숫자형(정수형)으로 바꾸는 경우 int() 함수를 사용함



동일한 이름으로 실수형을 정수형으로 바꾸는 경우 int() 함수를 사용함

2) 숫자형 변환

 int()

```
>>> a="123"; a+1
Traceback (most recent call last):
  File "<pyshell#137>", line 1, in <module>
    a="123"; a+1
TypeError: can only concatenate str (not "int") to str
>>> a="123"; int(a)+1
124
>>> a=12.3; int(a)
12
>>>
```

2) 숫자형 변환

float()

구분	주요 내용
문자형	숫자형(실수형)으로 바꾸는 경우 float() 함수를 사용함
정수형	연산 시 하나라도 실수형을 만나면 실수형의 연산을 수행함
실수형	내용을 정수형으로 형변환시 소수점 이하 자릿수는 버림

2) 숫자형 변환

float()

```
>>> a="12.3";int(a)
Traceback (most recent call last):
  File "<pyshell#140>", line 1, in <module>
    a="12.3";int(a)
ValueError: invalid literal for int() with base 10: '12.3'
>>> float(a)
12.3
>>> a=1.1;1+a+12
14.1
>>> 1+int(a)+12
14
>>>
```

2) 숫자형 변환

 올림, 버림, 반올림



int() 함수는 소수점 이하의 숫자를 버림



반올림 함수는 round()를 사용함

구분	반올림
양수	소수점 이하 해당자리에서 반올림 함
음수	소수점 이상 해당자리에서 반올림 함

2) 숫자형 변환

 올림, 버림, 반올림

```
>>> int(3.14)
3
>>> round(3.14);round(3.14,1);
round(123456, -4)
3
3.1
120000
>>>
```

2) 숫자형 변환

 올림, 버림, 반올림

▶ 버림 계산 예시

```
>>> int(3.14)
3
>>> round(3.14);round(3.14,1);
round(123456,-4)
3
3.1
120000
>>>
```

2) 숫자형 변환

 올림, 버림, 반올림

▶ 올림 계산 예시

```
>>> a=1234567
>>> if(a%1000 != 0 ) : #1000으로 나누어 나머지가 있다면
        a=int(a/1000)*1000+1000 #버린값에다 한자리를 올려 줌
else:
        a=int(a/1000)*1000

>>> a
1235000
>>>
```

04

실습하기 I

- 1) 대입 연산자 - 단순 대입 연산자
- 2) 대입 연산자 - 복합형태 대입 연산자
- 3) 산술 연산자 - 일반 산술 연산자

실습내용

- 1) 대입 연산자 - 단순 대입 연산자
- 2) 대입 연산자 - 복합형태 대입 연산자
- 3) 산술 연산자 - 일반 산술 연산자

05

실습하기 II

- 1) 산술 연산자 - 복합 연산자
- 2) 타입 캐스팅 - 문자형 변환
- 3) 타입 캐스팅 - 숫자형 변환

실습내용

- 1) 산술 연산자 - 복합 연산자
- 2) 타입 캐스팅 - 문자형 변환
- 3) 타입 캐스팅 - 숫자형 변환



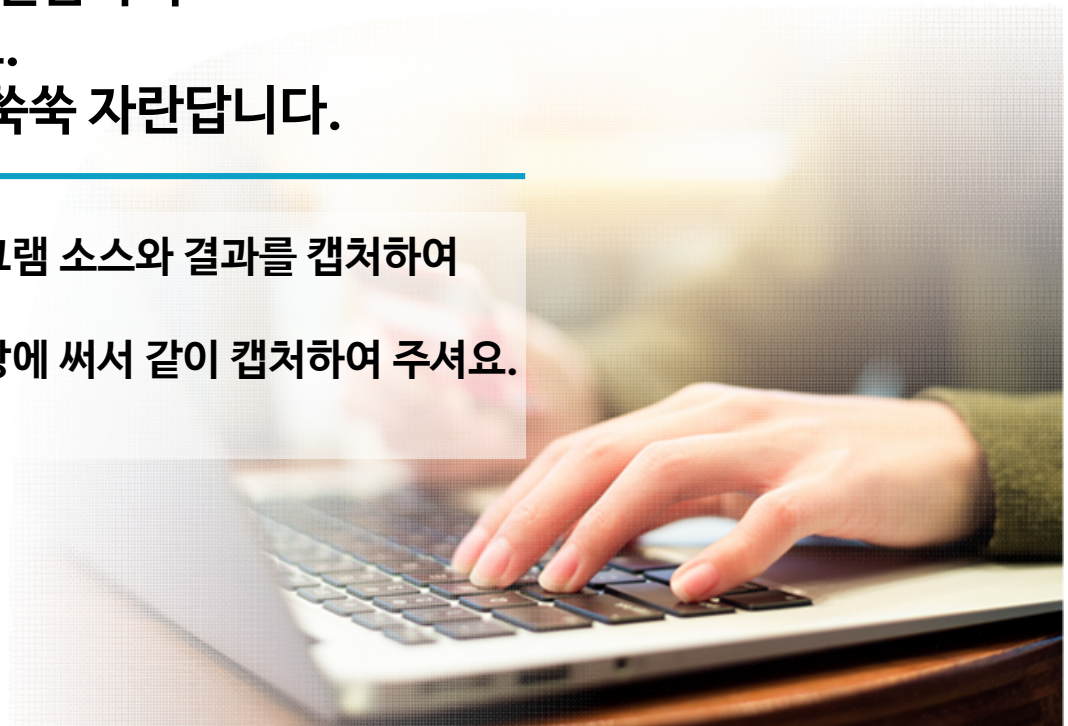
학습활동

일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

오늘 배운 내용을 스스로 실습하여
자유게시판에 올려주세요.
이렇게 정리하면 실력이 쑥쑥 자란답니다.

- ① 본인이 실습한 내용을 프로그램 소스와 결과를 캡처하여 올려주세요.
- ② 본인의 학번과 이름을 메모장에 써서 같이 캡처하여 주세요.
- ③ 그리고 설명도 달아주세요.





학습활동에 대한 교수님 의견

Q

오늘 배운 내용을 스스로 실습하여 자유게시판에 올려주세요.
이렇게 정리하면 실력이 쑥쑥 자란답니다.

A

[오늘 학습한 내용의 실습 사항]

- ① 대입 연산자 - 단순 대입 연산자
- ② 대입 연산자 - 복합형태 대입 연산자
- ③ 산술 연산자 - 일반 산술 연산자
- ④ 산술 연산자 - 복합 연산자
- ⑤ 타입 캐스팅 - 문자형 변환
- ⑥ 타입 캐스팅 - 숫자형 변환

학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

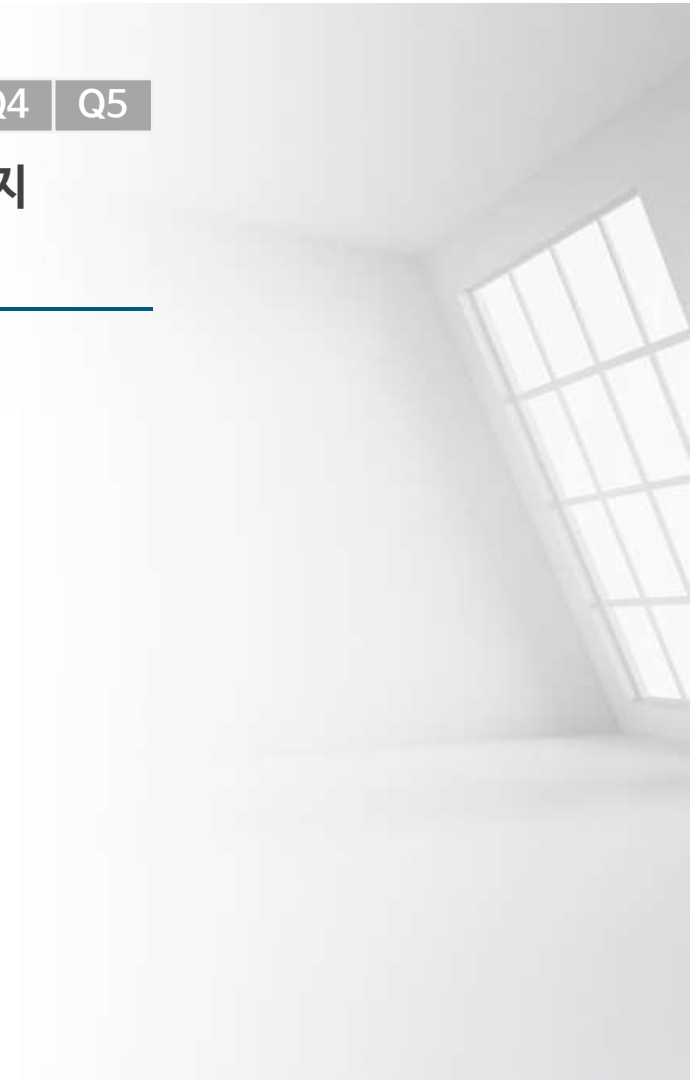
다음 중 대입연산자의 사용이 적합하지
않은 경우는?

1 $a=0x3a$

2 $a=a+1$

3 $a+1=a$

4 $a=b=1$



학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 중 대입연산자의 사용이 적합하지
않은 경우는?

1 $a=0x3a$

2 $a=a+1$

☒ 3 $a+1=a$

4 $a=b=1$

정답

3번

해설

변수=값, 또는 변수=수식의 형태로 표현합니다.

학습 평가

Q1

Q2

Q3

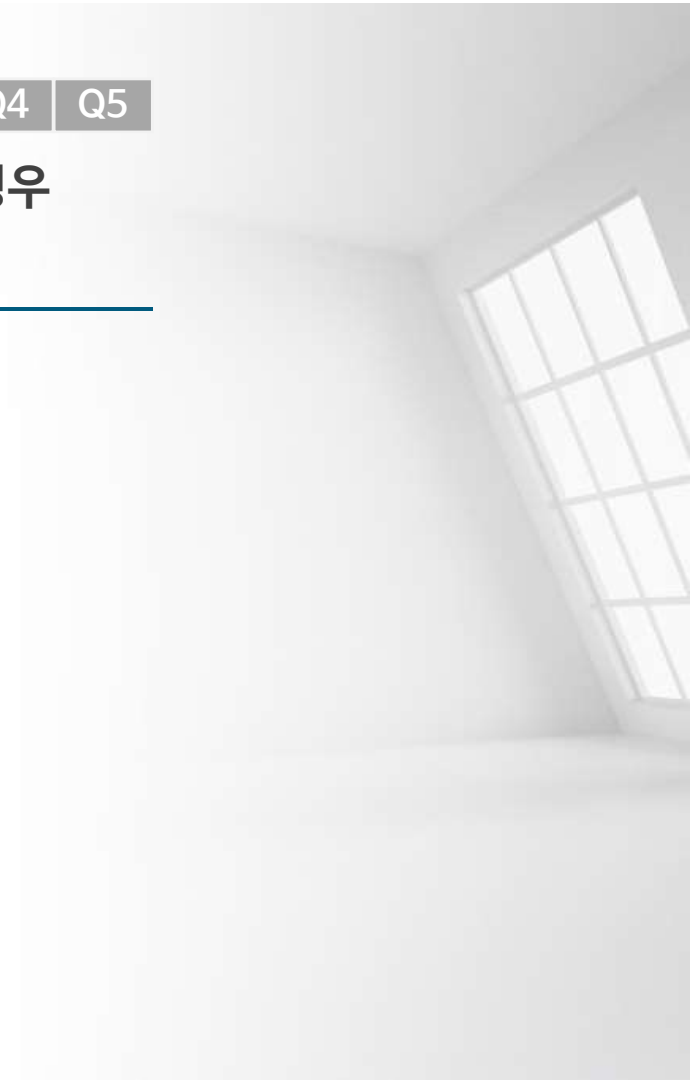
Q4

Q5

Q2

다음 $a=1; b=2$ 로 처음 값을 대입한 경우
 $a=2, b=1$ 이 되는 연산?

- 1 $a==b$
- 2 $a,b=a,b$
- 3 $a,b=b,a$
- 4 $a\langle\rangle b$



학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음 $a=1;b=2$ 로 처음 값을 대입한 경우
 $a=2, b=1$ 이 되는 연산?

1 $a==b$

2 $a,b=a,b$

☒ 3 $a,b=b,a$

4 $a\langle\rangle b$

정답

3번

해설

$a,b=b,a$ 의 경우 $a=b;b=a$ 와 동일한 표현입니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

다음 4×2 의 결과로 알맞은 것은?

1 16

2 2

3 2.0

4 8



학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

다음 4×2 의 결과로 알맞은 것은?

☒ 16

☐ 2

☐ 2.0

☐ 8

정답

1번

해설

4의 2제곱을 의미합니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q4

다음 $a=a+1$ 과 동일한 문장은?

1 $a++$

2 $++a$

3 $a=+1$

4 $a+=a$



학습 평가

Q1

Q2

Q3

Q4

Q5

Q4

다음 $a=a+1$ 과 동일한 문장은?

1 $a++$

2 $++a$

3 $a=+1$

☒ $a+=a$

정답

4번

해설

1을 더하는 누적은 $a+=a$ 로 표현합니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

다음 문자형을 숫자형으로 바꿀 때의
표현으로 잘못 사용된 것은?

- 1 `a=int("12.3")`
- 2 `a=int("12")`
- 3 `a=float("12")`
- 4 `a=float("12.3")`



학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

다음 문자형을 숫자형으로 바꿀 때
잘못 사용된 것은?



a=int("12.3")

2

a=int("12")

3

a=float("12")

4

a=float("12.3")

정답

1번

해설

"12.3"을 변환하기 위하여 실수형 변환을 하여야
합니다.



정리하기

대입 연산자

- ✓ 단순 대입시 변수=값, 변수=수식의 형태로 사용됨
- ✓ 변수의 값을 집어넣는 순간 변수의 자료형이 결정되며 다시 다른 자료형을 대입 시 자료형이 변경됨
- ✓ $a, b = b, a$ 로 a, b 변수의 값을 서로 바꿀 수 있으며, 여러 변수의 값을 한번에 대입할 수 있음





정리하기

산술 연산자

- ✓ 사칙연산 순으로 계산
 - (+, -, *, /, **) : 덧셈, 뺄셈, 곱셈, 나눗셈, 거듭제곱
- ✓ 정수형 연산자
 - //(정수형 나눗셈), %(나머지)
- ✓ 누적, 누곱
 - $a = a + 1$ ➡ $a += a$ 로 표시
 - $a = a * 2$ ➡ $a *= 2$ 로 표시





정리하기

타입 캐스팅

- ✓ 숫자형의 문자형으로 변환할 때는 `str()` 함수를 사용함
- ✓ 문자형을 숫자형 (정수형)으로 바꾸는 경우 `int()` 함수를 사용함
- ✓ 문자형을 숫자형 (실수형)으로 바꾸는 경우 `float()` 함수를 사용함
- ✓ 반올림함수는 `round()`를 사용함
 - 옵션으로 양수는 소수점 이하 해당자리, 음수는 소수점 이상 해당자리에서 반올림함

