

3. 변수 *Variable*

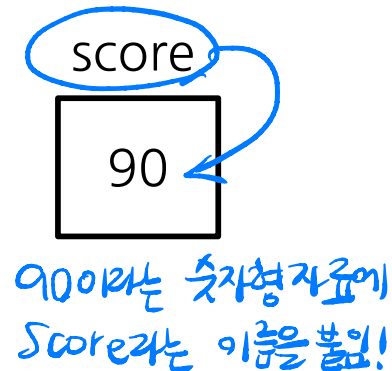
컴퓨터에 값을 저장하려면?

변수란

◆ 값을 저장하는 저장 공간

- 의미 있는 값은 저장해야 한다.
- 값을 저장해야 필요로 할 때 사용할 수 있고,
- 기존 값으로부터 새로운 값을 만들 수 있다.
- 저장 공간에 이름을 붙인 것이 변수이다. 이름을 붙여야 쓰기 편하므로.

값만 저장하고 끝나는 것이 아니라,
이름을 붙이는 것!



◆ 파이썬에서의 변수

- 자료형에 얽매이지 않는다.
- 필요로 할 때 언제든지, 얼마든지 만들어 사용할 수 있다.
- 따로 변수 이름이나 자료형을 선언할 필요 없다. (장점? 단점?)

파이썬만의 독특한 특징!
C/C++ 언어와 큰 차이

↳ 변수에 이름을 넣는 순간 바로 만들어짐!

동적 타이핑 : 그때 그때 자료형(type)을 결정하는 것

동적 타이핑

- C언어 -

```
int score;  
  
score = 10;  
  
score = 90.1; // error
```

score 변수의 자료형을 미리 선언

- Python -

```
score = 10      score는 정수형  
  
score = 90.1    score는 실수형  
  
score = 'hello' score는 문자형  
↳ 그때그때 무엇을 넣느냐에 따라 type 변화
```

↳ 이게 "동적타이핑"!

score 변수가 무엇인지 미리 선언하지 않는다.
왜냐하면 어차피 바뀔 수 있으므로

- ◆ 동적(Dynamic) : 실행하는 도중에
- ◆ 타이핑(Typing) : 자료형이 결정된다.

변수의 이름

◆ 변수 이름의 규칙

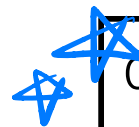
- 문자나 숫자, _를 이용하여 이름을 짓는다.
- 첫 글자는 문자 또는 _ → 첫글자는 숫자로 할수없음 !
- 한글 이름도 쓸 수 있다. 하지만,,,
- 공백이나 기호를 쓸 수 없다.
- 예약어를 쓸 수 없다.
- 이름은 이해하기 쉬운 것으로.

예약어(reserved word)란?

◆ 변수 이름의 예

- a a123 my_name MyName _pri

↳ 변수의 특성을 이해하기쉬움!



이름을 잘 짓는 것은 아주 중요하다 !

변수의 자료형

◆ 변수는 값을 넣을 때 변수도 만들어지고 자료형이 결정된다.

- 변수는 하나의 자료형을 갖는다.
- 변수는 자료형이 바뀔 수 있다.

```
a = 10
```

```
print(a)      # 변수 a는 정수다.
```

```
a = a + 1.1    # 변수 a는 실수(float)로 바뀌었다.
```

```
print(a)
```

```
a = a + b      # error : b는 값을 넣지 않았다.-> b는 만들어 지지도 않았다.
```

```
a = 10.0      # a의 자료형은? float
```

```
b = 10        # b의 자료형은? int
```

→ 변수 a의 자료형을
안다면!

type(a)

변수에 값 넣기



=

- 오른쪽의 결과값을 왼쪽의 변수에 넣는다.

score = 100 → 변수 생성!

ave = score / 3

score = score + 10

30 = score # 잘못된 사용 : =의 왼쪽에 변수

= 는 가장 마지막에 수행하는 연산자(우선순위 낮음)

확장된 사용

- c, d = 3, 4

- x = y = z = 10 x, y, z = 10, 10, 10으로도 가능

- e, f = f, e

e = f
f = e

→ 두 변수의 값을
change 하겠다!

변수에 값을 넣는 순간
변수가 생성되고 변수의 자료
형이 결정된다.

변수에 값 넣기

◆ 확장된 연산자 (간결한 표현)

◆ $+=$ $-=$ $*=$ $/=$ $//=$ $\%=$ $**=$ (우선순위는 $=$ 와 같다)

- $x += y$ 는 $x = x + y$

- $z += 1$ 는 $z = z + 1$

① $+=$

$a += 1 \longrightarrow a = a + 1$

② $-=$

$a -= 1 \longrightarrow a = a - 1$

③ $*=$

$a *= 1 \longrightarrow a = a * 1$

⋮

굉장히 많이 사용!

프로그램에는 이런 형식의 연산이 많다.

실습1

◆ 다음을 작성하라.

- ① – 반지름을 radius 라는 변수에 넣어라. 반지름 = 3.5 가정!
– 원의 면적을 계산하여 area 라는 변수에 넣어라.
– 원의 면적을 화면에 표시하라.
- ② – 원금을 money 변수에 넣어라. (원금 = 100만원)
– 이자율을 rate 변수에 넣어라. (이자율 = 0.25 가정!)
– 3년간 이자를 합한 총액을 1년 단위로 화면에 표시하라.
- Q. 총액 변수는 무엇으로 하면 더 알기 쉽고 편하게 쓸수 있을까?

◆ input()

- 키보드로부터(사용자로부터) 데이터를 입력 받는 함수(부품)

- 프롬프트 사용

- 키보드로부터 입력 받은 문자열 반환

```
name = input('your name ?')
```

```
print(name)
```

너 이름이 뭐니? **홍길동**

프롬프트

- 숫자 데이터 입력은 문자열을 숫자로 변환 ☆ 주의

```
age = int(input('your age?'))
```

```
age = input('your age?')
```

↳ 두개의 차이를 알기위해 IDLE에 작성해볼것!

이렇게 int()로
전환시키는 것이
매우 중요!

'23'과 23은 다르다.

↳ 어떻게 다를까?
무슨 차이일까?

◆ print()

- 문자열, 변수의 값, 수식을 화면에 표시
- , 로 구분하여 자료를 나열

```
print('4 + 5 = ', 4+5, '4 * 5 = ', 4*5)
```

이렇게 하면 어떻게 나올까?

4개의 값을 표시

- print는 기본으로 줄 바꿈을 한다.

```
print('4 + 5 = ', 4+5) # 4 + 5 = 9를 표시한 후 자동으로 줄 바꿈
```

```
print('4 + 5 = ', 4+5, end = '')
```

이것과의 차이는 무엇일까? 어떻게 표시될까?

end 를 이용하여 기본인 줄 바꿈을 다른 것으로 변경한다.

print를 여러 번 쓰면 인지를 바꿔서
이렇게 나오지 않음!
print를 여러 번 쓸 때

- print는 기본으로 항목 사이에 공백을 출력한다.

```
print(1, 2, 3, 4, 5)
```

```
print(1, 2, 3, 4, 5, sep = ',')
```

항목 사이에 공백 대신 , 를 사용

이들의 차이는 무엇일까?

◆ 다음의 프로그램을 작성하시오.

당신의 생년, 월, 일을 차례로 입력하세요.

생년? 1995
월? 3
일? 3

— 사용자가 입력한 값.

당신은 1995년 3월 3일에 태어났습니다.

생년월일을 합하면 2001 입니다.

- ◆ 이차방정식 $ax^2 + bx + c = 0$ ($a \neq 0$)의 두 근을 각각 표시하라. 단 두 근은 실근($b^2 - 4ac \geq 0$)이다.
- ◆ $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ 를 참고하라.
- ◆ a, b, c는 사용자로부터 입력 받는다.
- ◆ $\sqrt{a} = \text{sqrt}(a)$ # sqrt는 math 모듈에 있다

- ◆ 사용자로부터 원의 반지름을 입력 받아 원주의 길이($2\pi r$)와 원의 면적(πr^2)을 화면에 표시하라. 결과는 소수점 2자리까지만 표시하라.

– 실행 예)

반지름을 입력하세요 : 2

원주 : 12.56

면적 : 12.56