

Web/Python Programming

웹/파이썬 프로그래밍

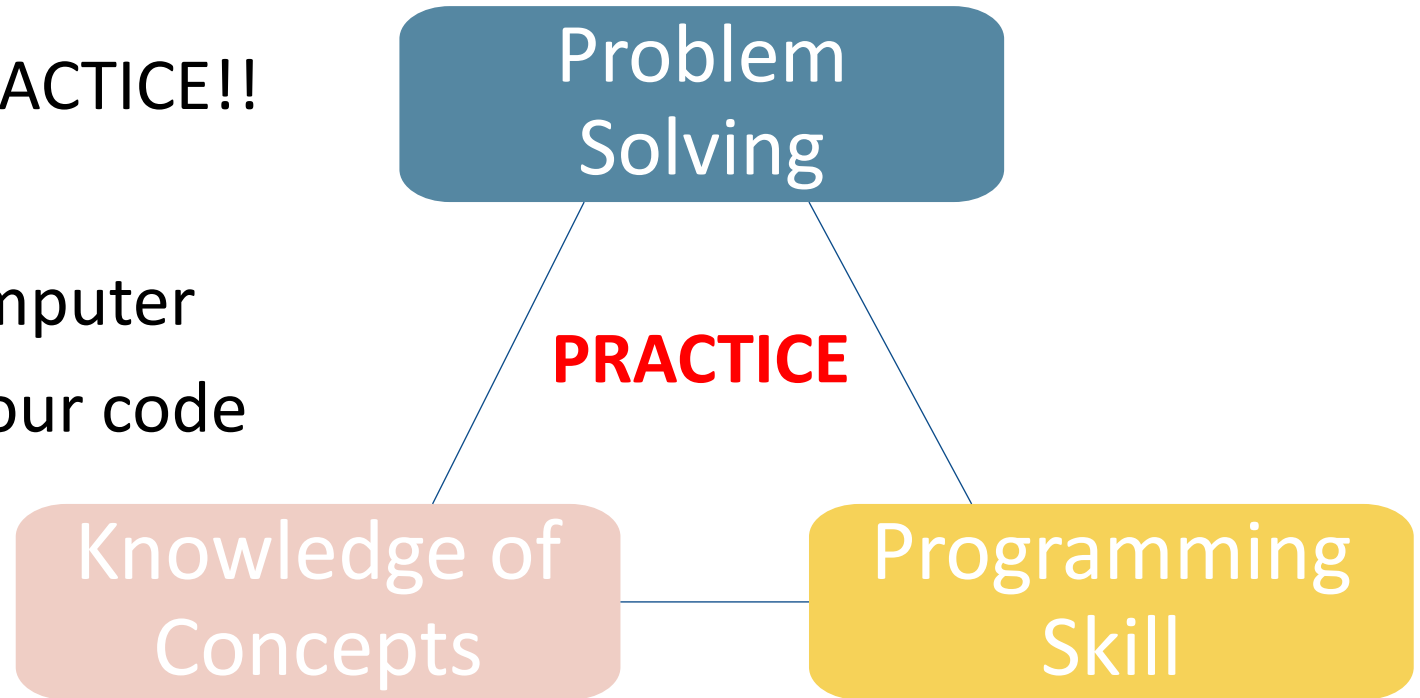
```
1 <?php language_attributes(); ?>
2
3 <?php charset = "<?php bloginfo( 'charset' ); ?>" ?>
4 <?php name = "viewport" content="width=device-width" ?>
5 <?php wp_title( '|', true, 'right' ); ?>
6 <?php rel = "profile" href="http://gmpg.org/xfn/11" ?>
7 <?php rel = "pingback" href="<?php bloginfo( 'pingback_url' ); ?>" ?>
8 <?php fruitful_get_favicon(); ?>
9 <!--[if IE 9]><script src="<?php echo get_template_directory_uri(); ?>/js/html5.js"></script></if></head>
10
11 <?php wp_head(); ?>
12
13 <?php body_class(); ?>
14 <div id="page-header" class="hfeed site">
15
16 <?php
17 $theme_options = fruitful_get_theme_options();
18 $logo_pos = $theme_options['logo_position'];
19 if (isset($theme_options['logo_position']))
20     $logo_pos = esc_attr($theme_options['logo_position']);
21
22 if (isset($theme_options['menu_position']))
23     $menu_pos = esc_attr($theme_options['menu_position']);
24
25 $logo_pos_class = fruitful_get_theme_option('logo_pos_class');
26 $menu_pos_class = fruitful_get_theme_option('menu_pos_class');
27
28 $responsive_menu_type = fruitful_get_theme_option('responsive_menu_type');
29 $responsive_menu_type = esc_attr($responsive_menu_type);
30
31 </div>
32
33
34
35
```

Today

- Python basics
- Variables and computer memory
- Assignment statement
- Augmented statement

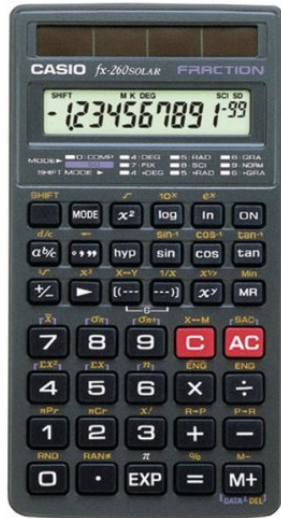
Fast paced course

- New to programming?
- PRACTICE PRACTICE PRACTICE!!
- You can't break your computer
- Don't be afraid to test your code
- Worst case: reboot

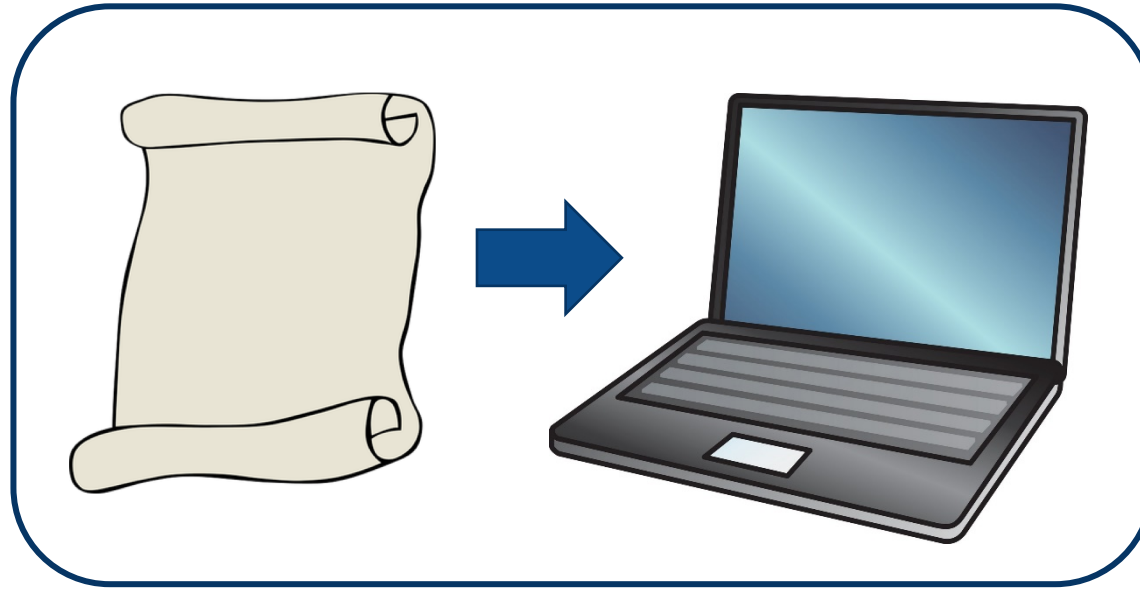


What is programming?

- A program is a set of instructions



VS.

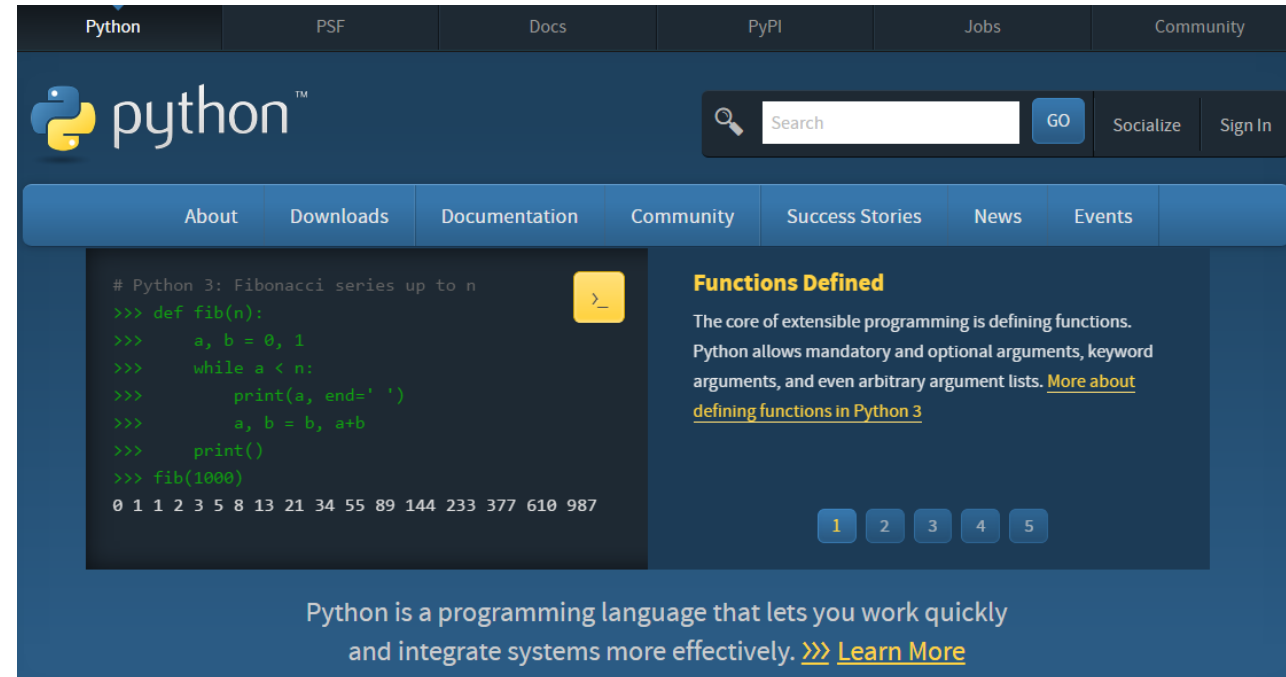


- You can “teach” a computer new operations

Why Python?

- It is free and well documented
- It runs everywhere
 - supports multiple platforms
- It has a clean syntax
- It is relevant
 - many companies use it every day
- It is well supported by tools
 - IDLE, PyCharm, etc.
 - Jupyter Notebook

■ `www.python.org`



What is a Bug?

- May cause a program crash
- May give incorrect results
- Every program has bugs!
- Kinds of errors
 - **Syntax error:** Interpreter cannot understand your code and refuses to execute it
 - **Runtime error:** When executing your program (at runtime), your program suddenly terminates with an error message
 - **Semantic error:** Your program runs without error messages, but does not do what it is supposed to do

A problem has been detected and Windows has been shut down to prevent damage to your computer.

UNMOUNTABLE_BOOT_VOLUME

If this is the first time you've seen this error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

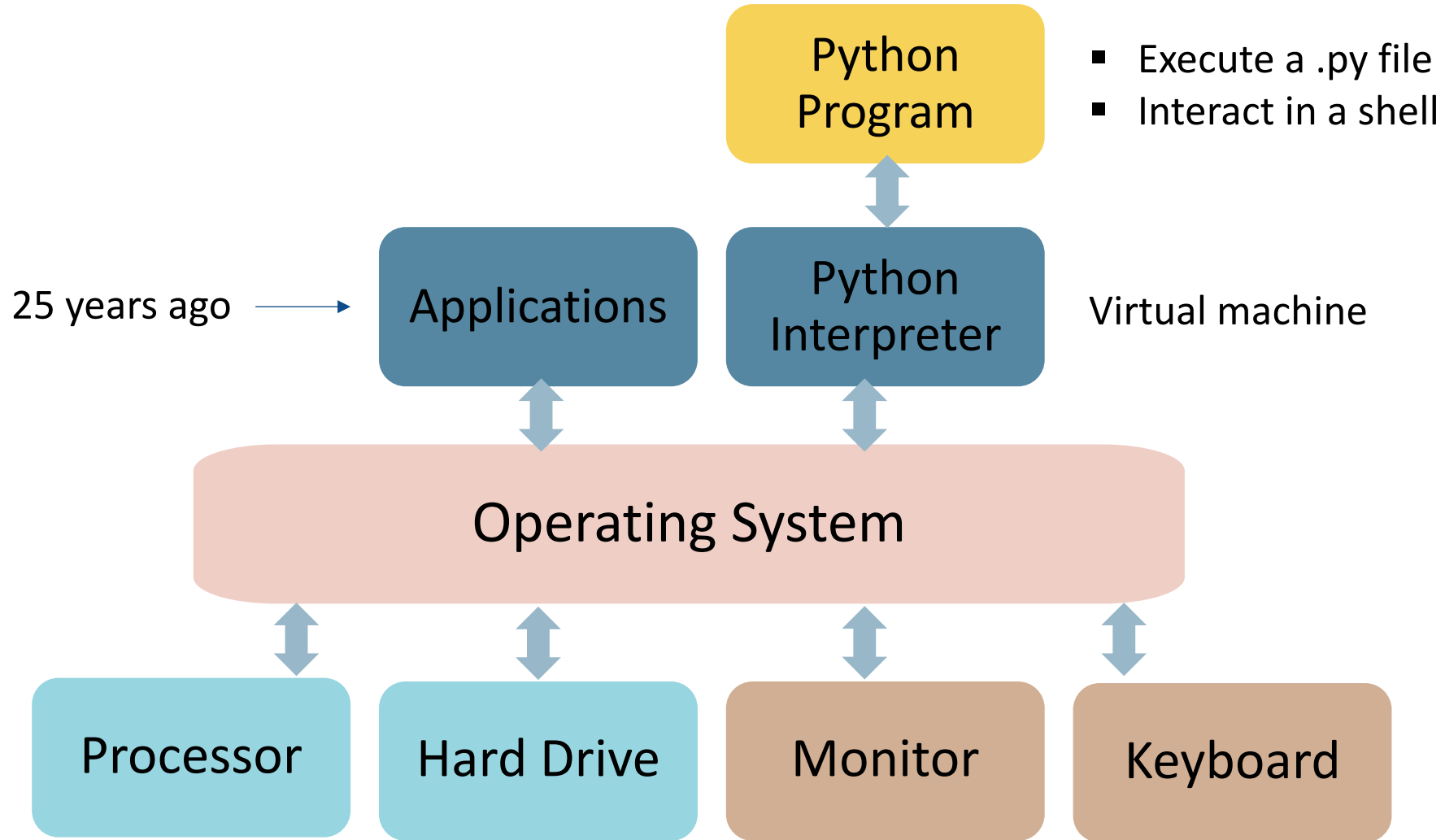
Technical Information:

*** STOP: 0x000000ED (0x80F128D0, 0xc000009c, 0x00000000, 0x00000000)

Python basics

- () Parentheses (소괄호)
- [] Brackets (대괄호)
- { } Braces (중괄호)
- IDLE programming environment

How does a computer run a python program?



Interact in a Python shell

- Arithmetic in Python
 - Addition, subtraction, multiplication, division
- Types
 - int, float, complex

```
>>> type(17)
<class 'int'>
>>> type(17.0)
<class 'float'>
>>> type(1+2j)
<class 'complex'>
>>> type(0o34)
<class 'int'>
>>> type(0x8f)
<class 'int'>
>>>
```

```
>>> a=0o34
>>> a
28
>>> b=0x8f
>>> b
143
>>>
```

Symbol	Operator	Example	Result
-	Negation	-5	-5
+	Addition	11 + 3.1	14.1
-	Subtraction	5 - 19	-14
*	Multiplication	8.5 * 4	34.0
/	Division	11 / 2	5.5
//	Integer Division	11 // 2	5
%	Remainder	8.5 % 3.5	1.5
**	Exponentiation	2 ** 5	32

Table 1—Arithmetic Operators

Finite precision

- Computers have a finite amount of memory

```
>>> 2 / 3
0.6666666666666666
>>> 5 / 3
1.6666666666666667
>>>
```

- Operator precedence

- Ex) Fahrenheit to Celsius: $(F - 32) * 5/9$
- Ex) $212^{\circ}\text{F} = 100^{\circ}\text{C}$

```
>>> 212 - 32 * 5 / 9
194.22222222222223
>>> (212 - 32) * 5 / 9
100.0
```

Variables

- Let's give a name to a value

- X, species5618, degrees_celsius
 - 777obj(X), no-way(X), hello!(X)

- Assignment statement

```
>>> degrees_celsius = 26.0
```

- You can assign a new value to the existing variable

```
>>> degrees_celsius = 26.0
>>> degrees_celsius
26.0
>>> 9 / 5 * degrees_celsius + 32
78.80000000000001
>>> degrees_celsius / degrees_celsius
1.0
```

```
>>> degrees_celsius = 26.0
>>> 9 / 5 * degrees_celsius + 32
78.80000000000001
>>> degrees_celsius = 0.0
>>> 9 / 5 * degrees_celsius + 32
32.0
```

- Note that = means “assignment”, not “equality”

Values, variables, and computer memory

- Every location in the computer's memory has a **memory address**
- **Object**: a value at a memory address with a type

26.0

id1

float



- **Variable** contains the memory address of the object

`degrees_celsius`

Values, variables, and computer memory



- **Object:** a value at a memory address with a type

`26.0`

`id1`

`float`

- **Variable** contains the memory address of the object

`degrees_celsius`

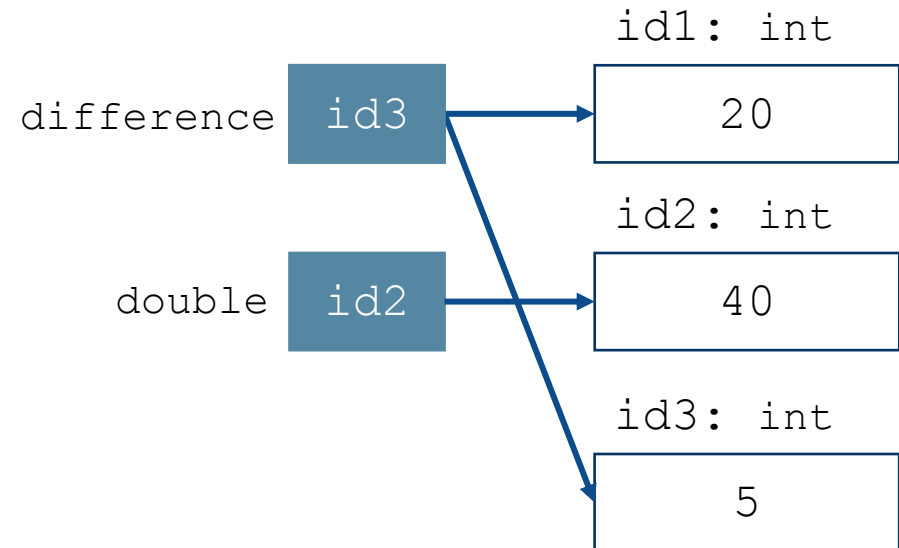
- Value `26.0` has the memory address `id1`.
- The object at the memory address `id1` has type `float` and the value `26.0`
- Variable `degree_celsius` contains the memory address `id1`.

Assignment statement

```
>>> degrees_celsius = 26.0 + 5
>>> degrees_celsius
31.0
```



```
>>> difference = 20
>>> double = 2 * difference
>>> double
40
>>> difference = 5
>>> double
40
```



Memory visualization

- <http://pythontutor.com/visualize.html>

Write code in Python 3.6

```
1 difference = 20
2 double = 2 * difference
3 double
4 difference = 5
5 double
6 |
```

Python 3.6

1 difference = 20
2 double = 2 * difference
3 double
→ 4 difference = 5
→ 5 double

[Edit code](#) | [Live programming](#)

→ line that has just executed
→ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First

< Back

Step 5 of 5

Forward >

Last >>

Frames

Global frame
difference id3
double id2

Objects

id2:int
40

id3:int
5

Support our research and keep this tool free by [filling out this survey on how your native spoken language affects how you learn programming](#).

Visualize Execution

Live Programming Mode

hide exited frames [default]

render all objects on the heap (Python/Java)

use text labels for pointers

[Create test cases](#)

Memory visualization

```
>>> number = 3
```

```
>>> number
```

```
3
```

```
>>> number = 2 * number
```

```
>>> number
```

```
6
```

```
>>> number = number * number
```

```
>>> number
```

```
36
```

Memory visualization

```
>>> number = 3
```

```
>>> number
```

```
3
```

```
>>> number = 2 * number
```

```
>>> number
```

```
6
```

```
>>> number = number * number
```

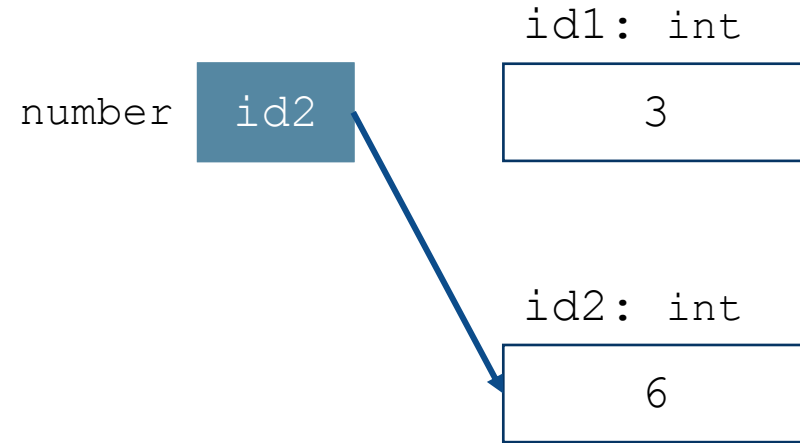
```
>>> number
```

```
36
```



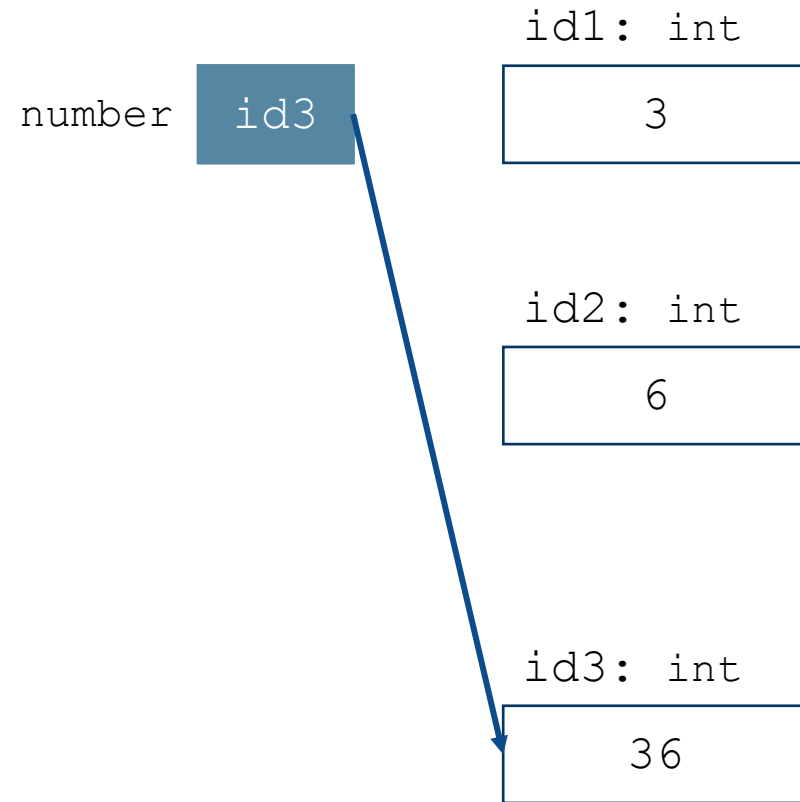
Memory visualization

```
>>> number = 3
>>> number
3
>>> number = 2 * number
>>> number
6
>>> number = number * number
>>> number
36
```



Memory visualization

```
>>> number = 3
>>> number
3
>>> number = 2 * number
>>> number
6
>>> number = number * number
>>> number
36
```



Augmented assignment

```
>>> score = 50
```

```
>>> score
```

```
50
```

```
>>> score = score + 20
```

```
>>> score
```

```
70
```

```
>>> score =50
```

```
>>> score
```

```
50
```

```
>>> score += 20
```

```
>>> score
```

```
70
```

Augmented assignment

```
>>> d = 2
```

```
>>> d *= 3 + 4
```

```
>>> d
```

```
14
```

```
>>> number = 10
```

```
>>> number *= number
```

```
>>> number
```

```
100
```

Symbol	Example	Result
+=	x = 7 x += 2	x refers to 9
-=	x = 7 x -= 2	x refers to 5
*=	x = 7 x *= 2	x refers to 14
/=	x = 7 x /= 2	x refers to 3.5
//=	x = 7 x //= 2	x refers to 3
%=	x = 7 x %= 2	x refers to 1
**=	x = 7 x **= 2	x refers to 49

Table 3—Augmented Assignment Operators