Web/Python Programming

웹/파이썬 프로그래밍

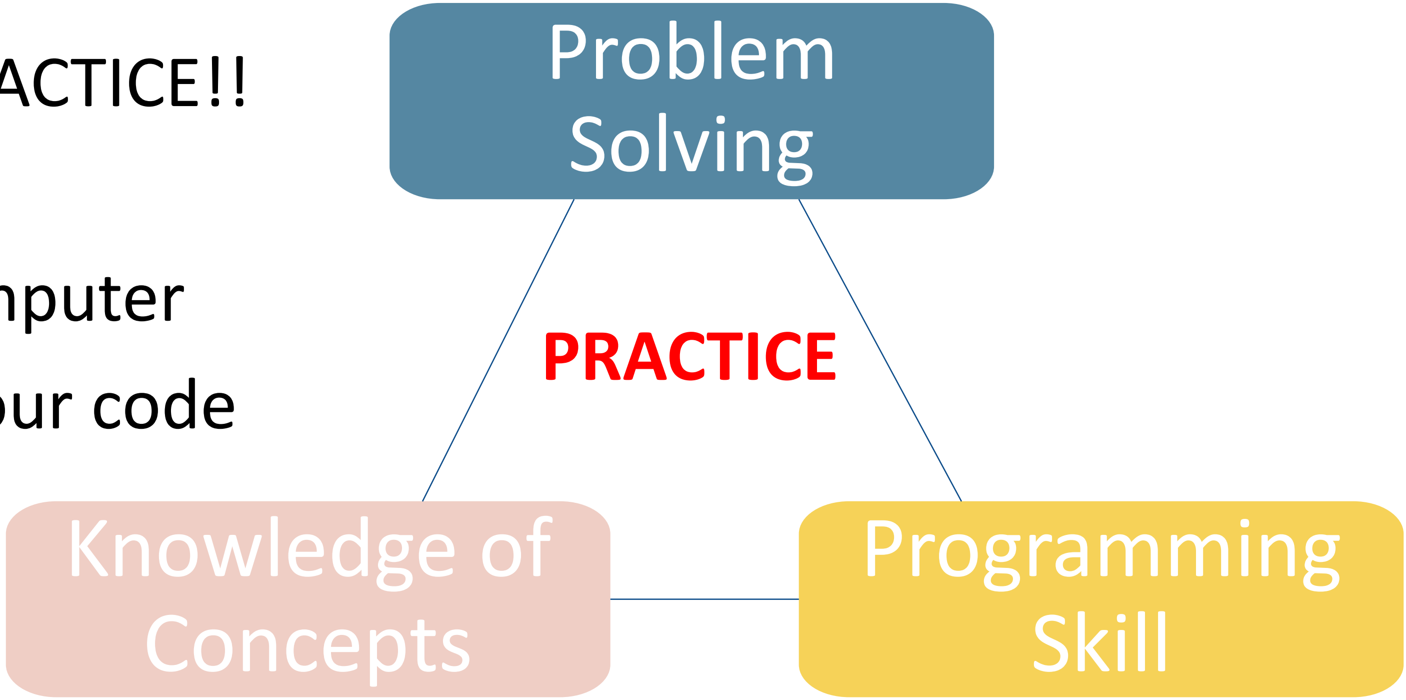
Today

- Python basics
- Variables and computer memory
- Assignment statement
- Augmented statement
- Covers Chapter 2 of your textbook

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



Problem Solving

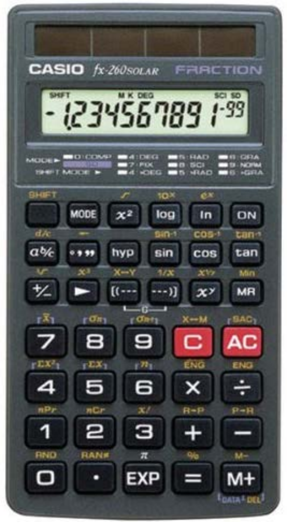
PRACTICE

Knowledge of Concepts

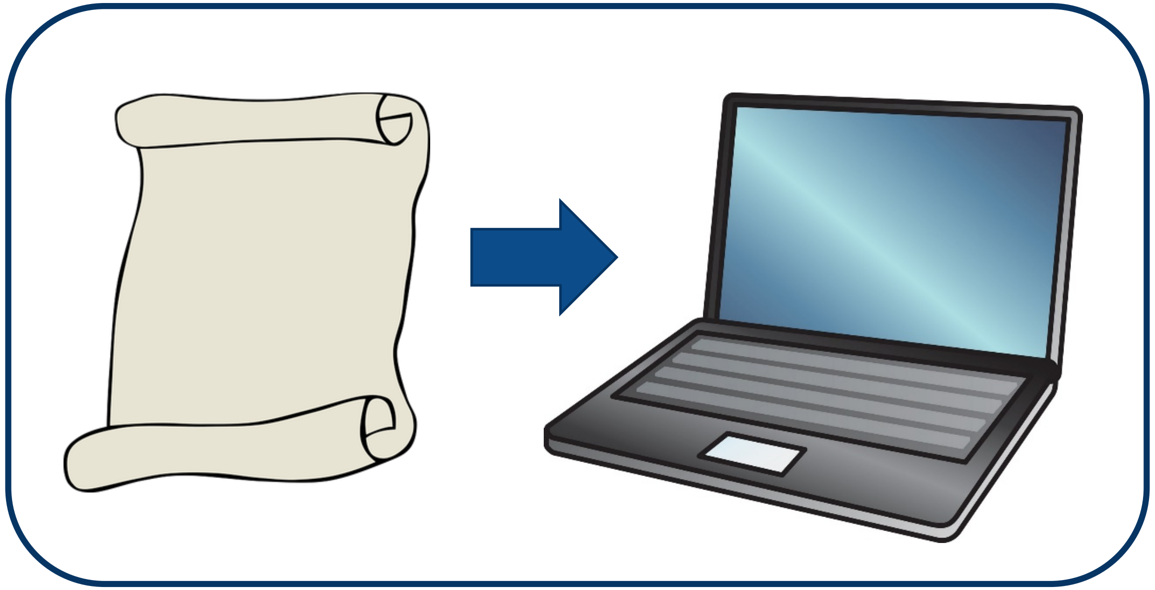
Programming Skill

What is programming?

- A program is a set of instructions



VS.

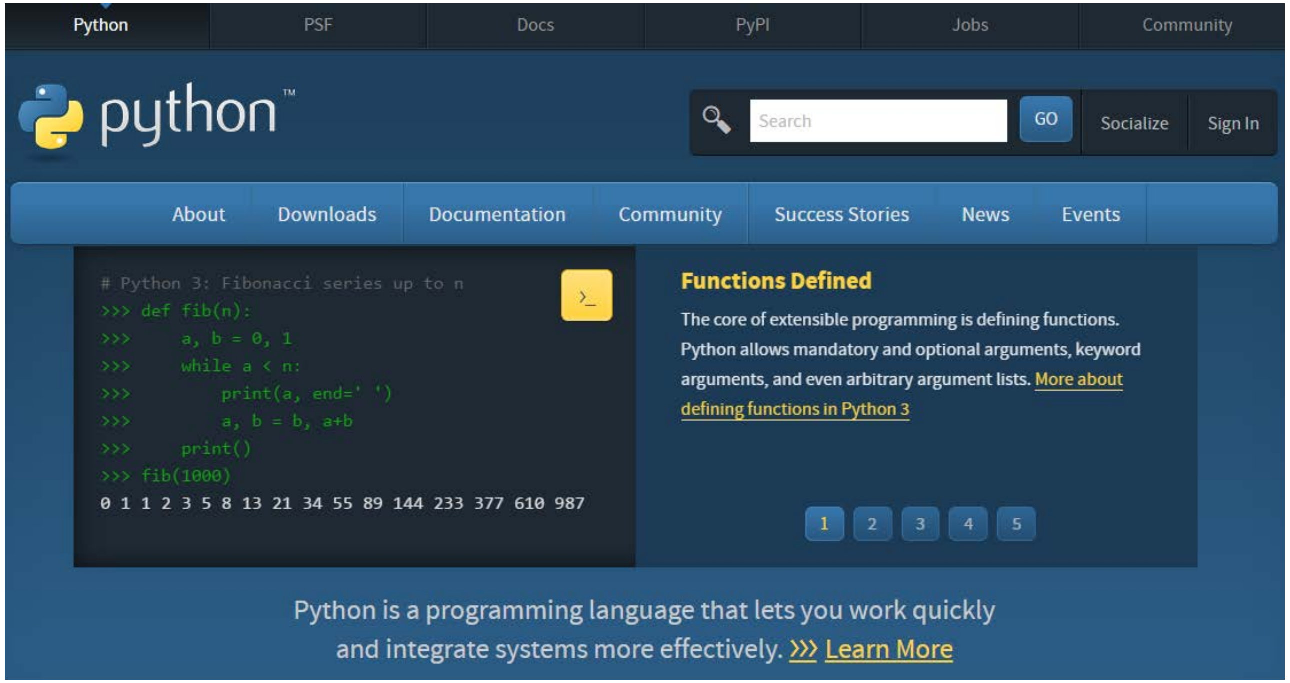


- You can “teach” a computer new operations

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`



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

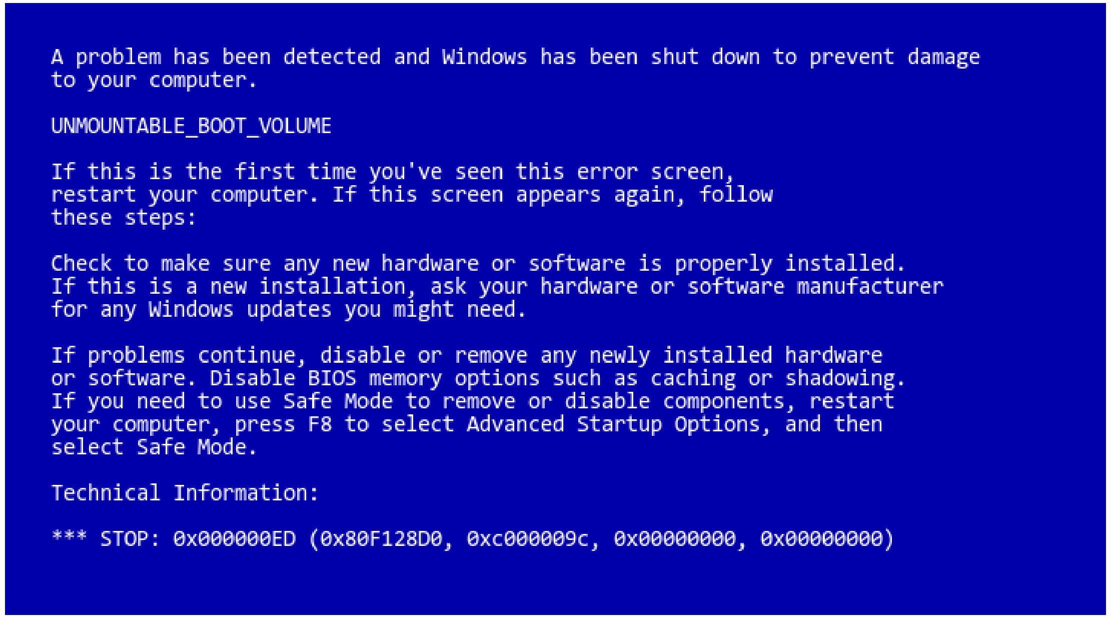
Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

What is a Bug?

- May cause a program crash
- May give incorrect results



```
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)
```

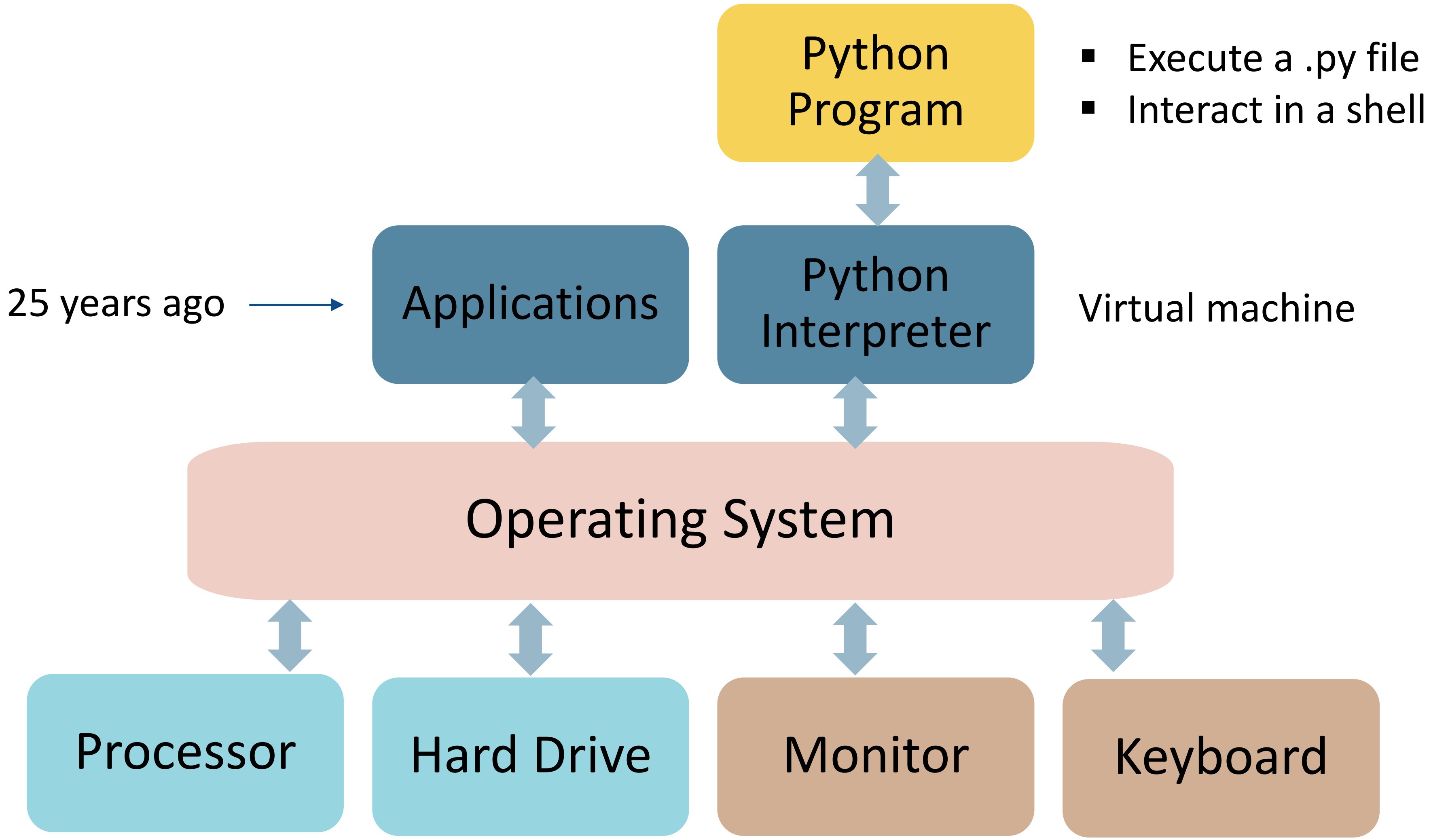
- Every program has bugs!
- Kinds of errors
  - **Syntax error**: Interpreter cannot understand your code and refuses to execute it
  - **Runtime error**: When executing your program (at runtime), your program suddenly terminates with an error message
  - **Semantic error**: Your program runs without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

Python basics

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

- IDLE programming environment

How does a computer run a python program?

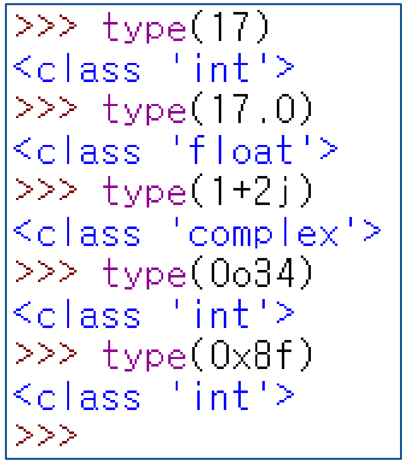


- Execute a .py file
- Interact in a shell

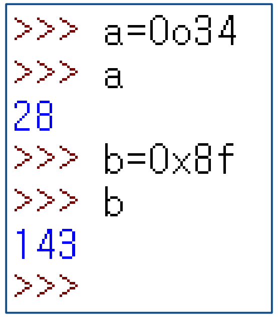
Virtual machine

Interact in a Python shell

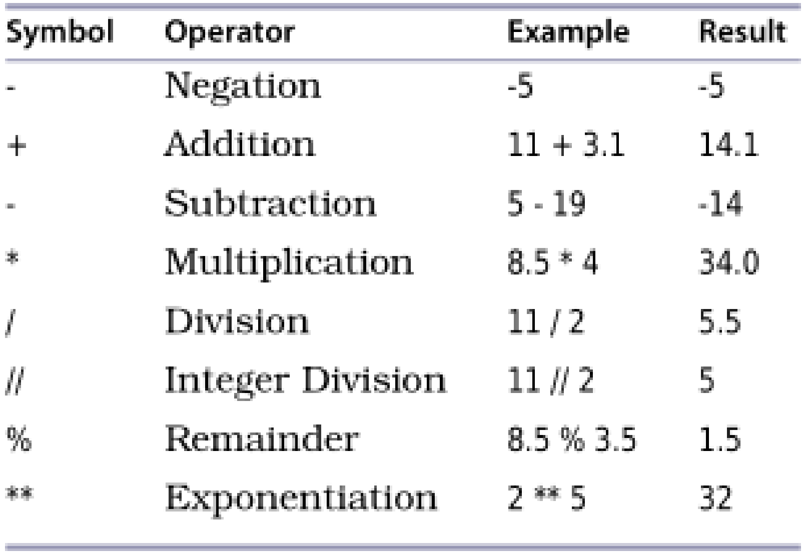
- Arithmetic in Python
  - Addition, subtraction, multiplication, division
- Types
  - `int, float, complex`



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



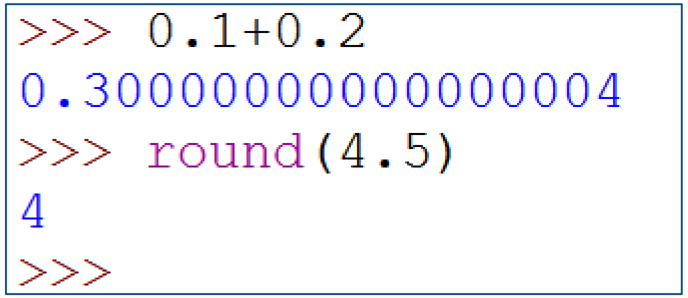
```
>>> a=0o34
>>> a
28
>>> b=0x8f
>>> b
143
>>>
```



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

Question



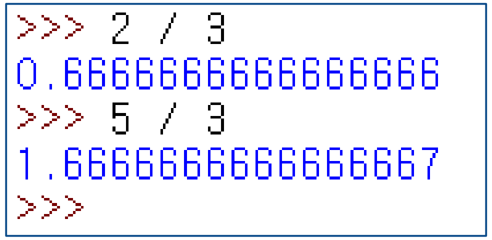
```
>>> 0.1+0.2
0.30000000000000004
>>> round(4.5)
4
>>>
```

- Read: https://docs.python.org/2/tutorial/floatingpoint.html

- Floating-point numbers are represented in computer hardware as base 2 (binary) fractions.
- Decimal fraction vs. binary fraction

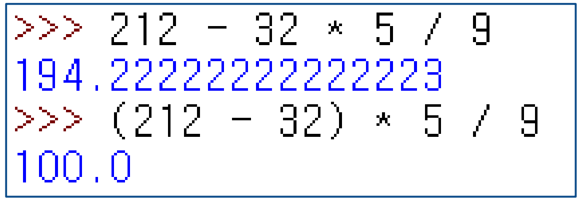
Finite precision

- 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 ^{o}F = 100 ^{o}C



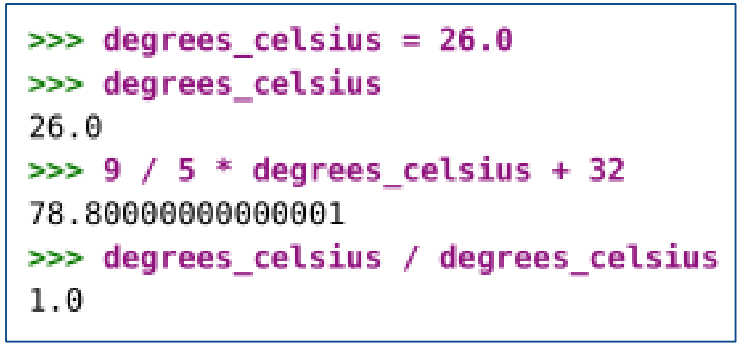
```
>>> 212 - 32 * 5 / 9
194.22222222222223
>>> (212 - 32) * 5 / 9
100.0
```

Variables

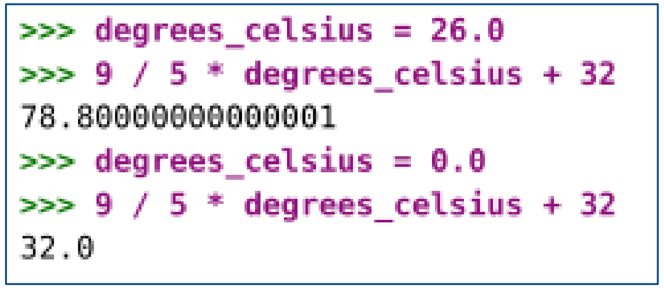
- Let’s give a name to a value
  - `X, species5618, degrees_celsius`
  - `777obj(X), no-way(X), hello!(X)`
- Assignment statement

```
>>> 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
```



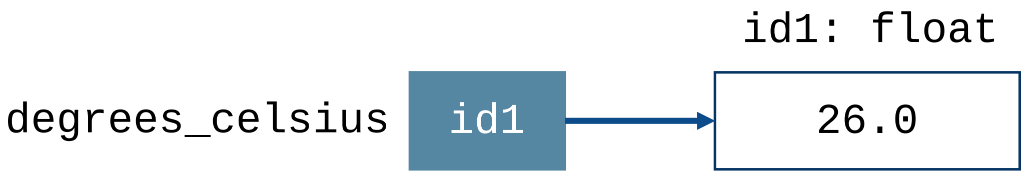
```
>>> degrees_celsius = 26.0
>>> 9 / 5 * degrees_celsius + 32
78.80000000000001
>>> degrees_celsius = 0.0
>>> 9 / 5 * degrees_celsius + 32
32.0
```

- 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

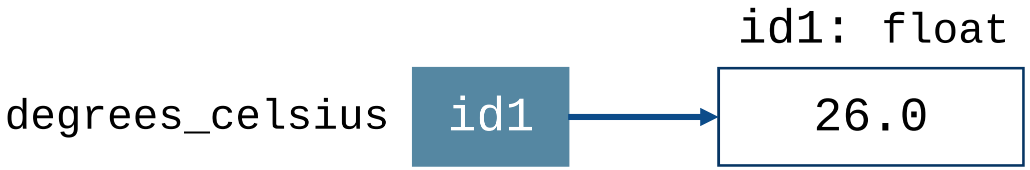
26.0 id1 float



- **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

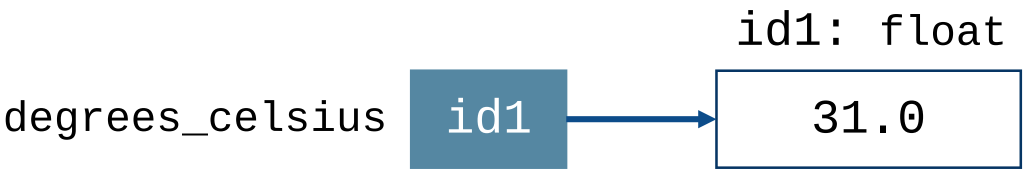
- **Variable** contains the memory address of the object

degrees_celsius

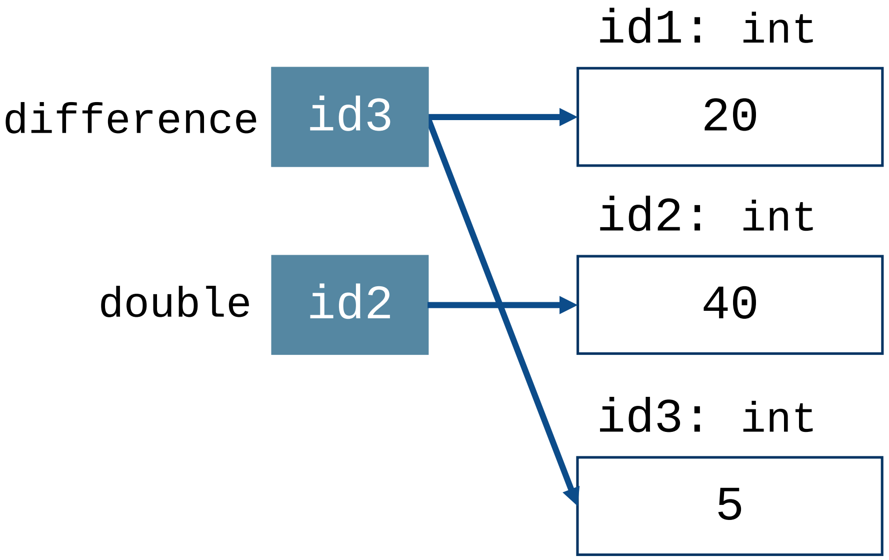
  - Value `26.0` has the memory address `id1`.
  - The object at the memory address `id1` has type `float` and the value `26.0`
  - Variable `degree_celsius` contains the memory address `id1`.

Assignment statement

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

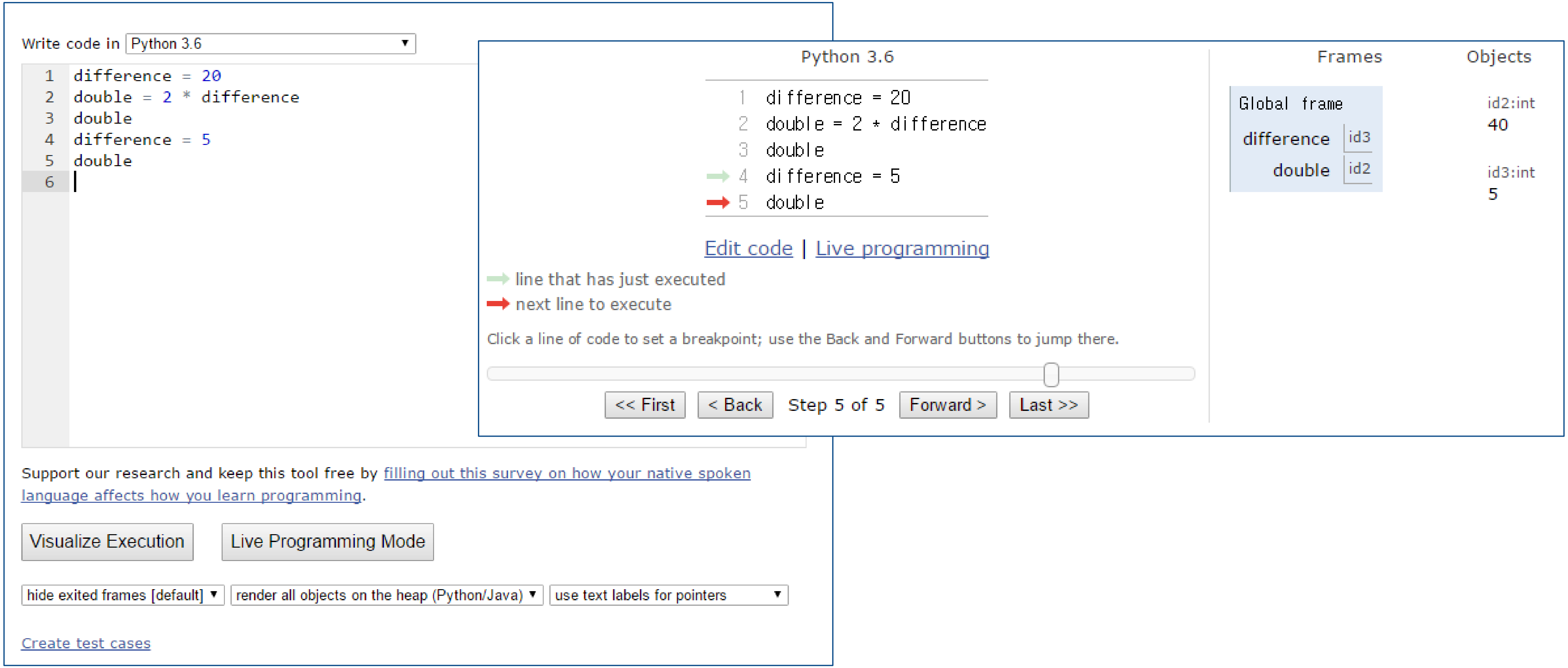


```
>>> difference = 20
>>> double = 2 * difference
>>> double
40
>>> difference = 5
>>> double
40
```



Memory visualization

- `http://pythontutor.com/visualize.html`



Write code in Python 3.6

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

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

hide exited frames [default] | render all objects on the heap (Python/Java) | use text labels for pointers

Create test cases

Python 3.6

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

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.

Frames | Objects

Global frame
difference id3
double id2

id2:int
40

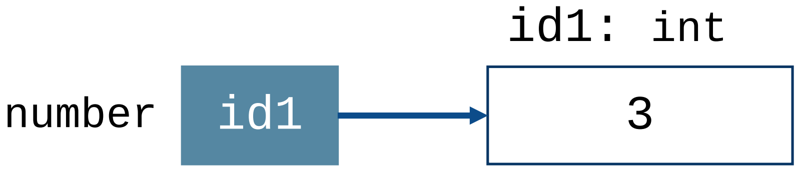
id3:int
5

Memory visualization

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

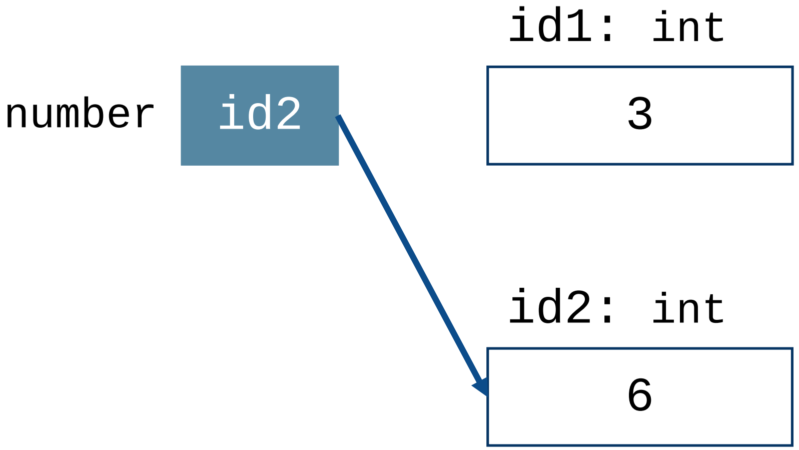
Memory visualization

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



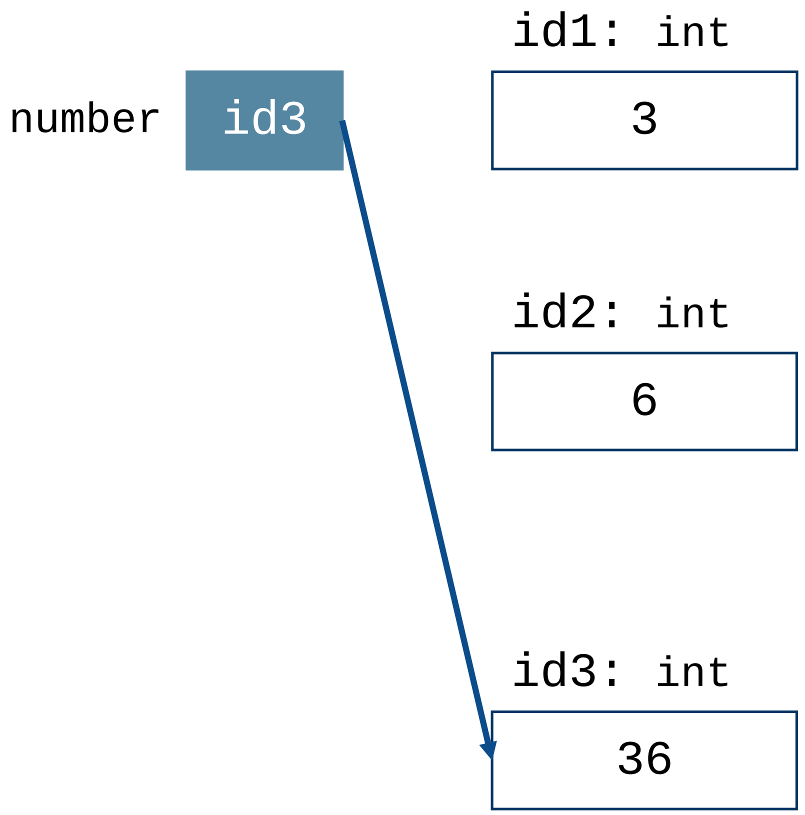
Memory visualization

```
>>> 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

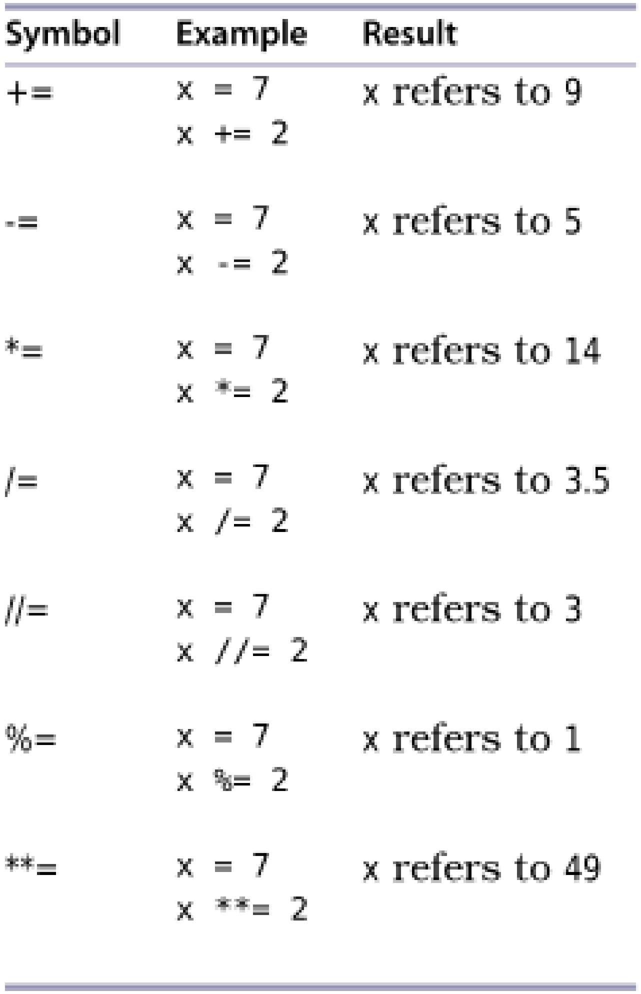
```
>>> score = 50
>>> score
50
>>> score = score + 20
>>> score
70
```

```
>>> 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

More information in your textbook

- Read Chapters 2.8 and 2.9
- Do Exercises 2.10