

Web/Python Programming

웹/파이썬 프로그래밍

```
21 <?php language_attributes(); ?>
22 <?php bloginfo( 'charset' ); ?>
23 <?php bloginfo( 'width=device-width' ); ?>
24 <?php wp_title( '|', true, 'right' ); ?>
25 <?php rel="profile" href="http://gmpg.org/xfn/11" ?>
26 <?php rel="pingback" href="http://gmpg.org/xfn/11" ?>
27 <?php fruitful_get_favicon(); ?>
28 <?php echo get_template_directory_uri(); ?>
29 <?php wp_head(); ?>
30 </head>
31 <?php body_class(); ?>
32 <div id="page-header" class="hfeed site">
33     $theme_options = fruitful_get_theme_options();
34     $logo_pos = $menu_pos = '';
35     if (isset($theme_options['logo_position']))
36         $logo_pos = esc_attr($theme_options['logo_position']);
37     if (isset($theme_options['menu_position']))
38         $menu_pos = esc_attr($theme_options['menu_position']);
39     $logo_pos_class = fruitful_get_class($logo_pos);
40     $menu_pos_class = fruitful_get_class($menu_pos);
41     $responsive_menu_type = fruitful_get_class($menu_pos);
42     $responsive_menu_type = fruitful_get_class($menu_pos);
```

Welcome to SWCon104

- Course objective

- Introduce basic concept of programming and computational thinking
- Help students understand how to map problems into a computational framework
- Prepare freshmen with no prior programming experience for entry into other programming based courses
- Help students get familiar with web programming using Python
- Web programming
- Web server and client

Course info

- Lecture + Practice
- Full time exam
 - Mid-term
 - Final-term
- Scoring (= Syllabus)

- Textbook
 - ***Think Python 2nd Edition by Allen B. Downey***
 - Free under CCL license
 - <http://greenteapress.com/wp/think-python-2e/>
 - 박응용, “점프 투 파이썬”
 - 정인용, “자바스크립트+제이쿼리 입문”

Course homepage

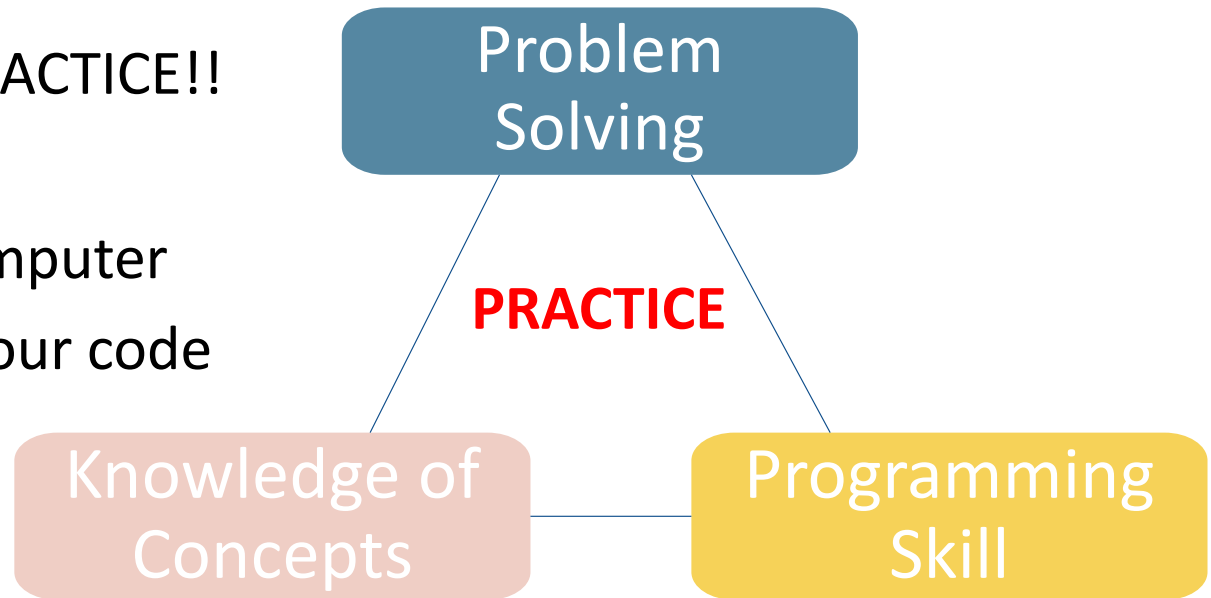
- Dropbox
 - Posted in KHUIS Syllabus
- Web/Python BBS
 - <http://mobilelab.khu.ac.kr/webpythonbbs/>
- On-line Lecture (2018-2H)
 - <http://mobilelab.khu.ac.kr/webpythonbbs/>

Today

- Course introduction
- What does a computer do?
- What is programming?
- Computational thinking
- Python installation
- Intro to Python (Author, Python.org, Others)

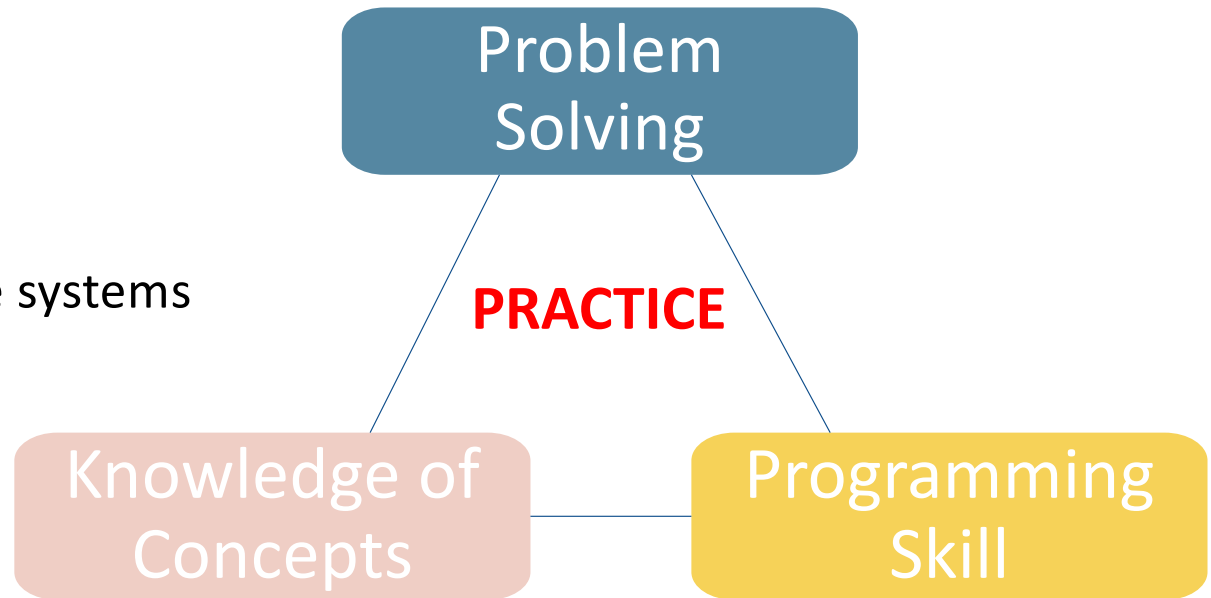
Fast paced course

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



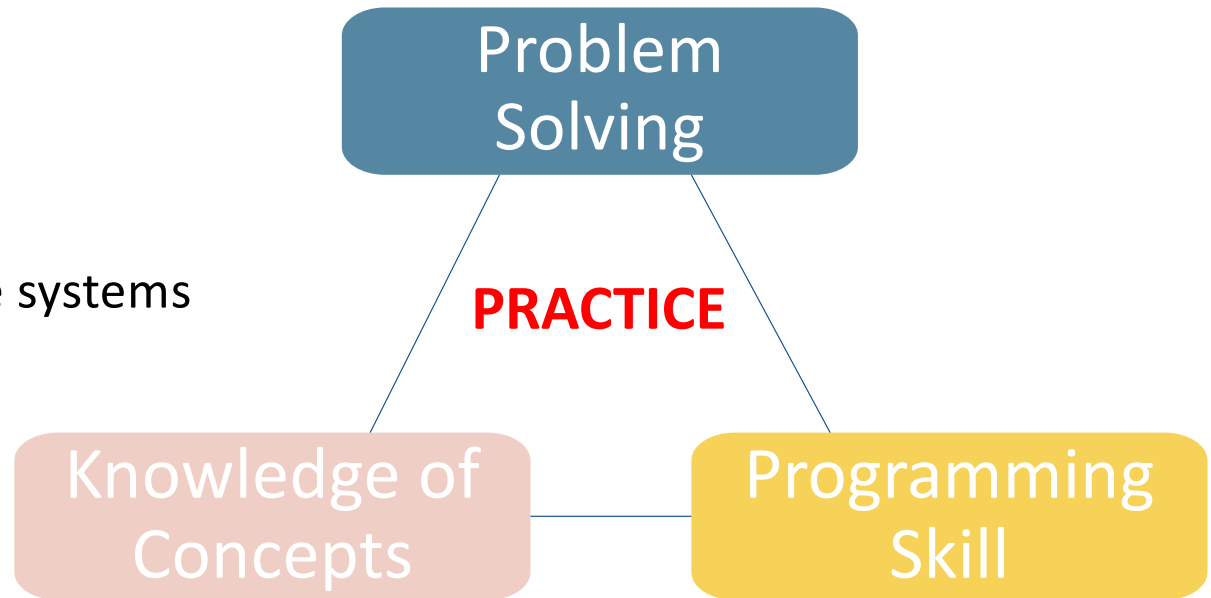
Topics

- How to program
 - Data structures
 - Iteration and recursion
- How to write good code
 - Organize and modularize systems
 - Classes and methods
- How to evaluate
 - Different algorithms
 - Complexity



Topics

- How to program
 - Data structures
 - Iteration and recursion
- How to write good code
 - Organize and modularize systems
 - Classes and methods
- How to evaluate
 - Different algorithms
 - Complexity

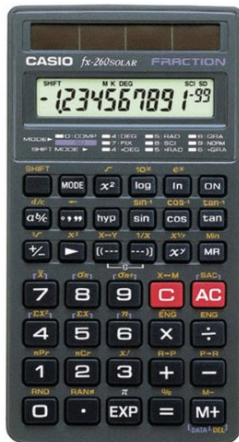


What does a computer do?

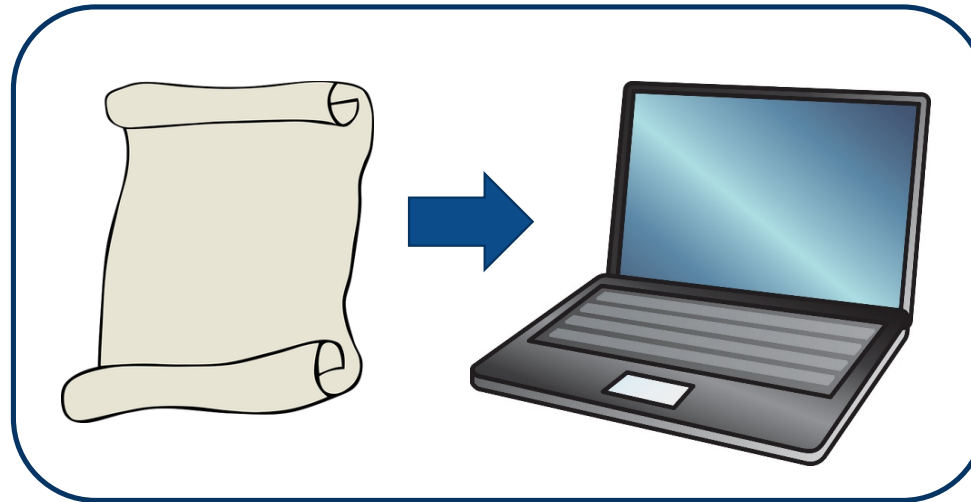
- Fundamentally:
 - Performs calculations
 - Remembers results
- What kinds of calculations?
 - Built-in to the language
 - Ones that you define as the programmer
- Computers only do what you tell them to do

What is programming?

- A program is a set of instructions



VS.



- You can “teach” a computer new operations

Computational thinking

- Computer science is the study of computation
 - What can be computed and how to compute it
- Characteristics of computational thinking
 - Conceptualizing, not programming
 - A fundamental skill
 - A way that humans think
 - Complements and combines mathematical and engineering thinking
 - Ideas
 - For everyone, everywhere
- One can major in computer science/software convergence and do anything!

Programming language

- There are many programming languages

English

Add 3 and 4

Python

3 + 4

Scheme

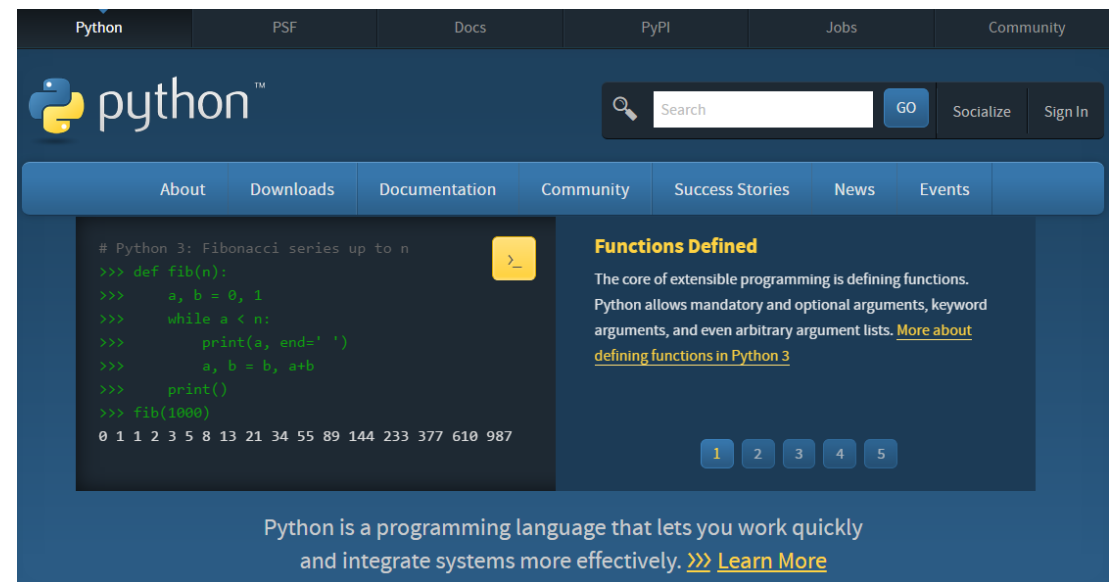
(+ 3 4)

- Mathematical expressions (add, subtract, multiply, divide...)
- Repeat a list of instructions a number of times (loop operations)
- Choose which of two instructions to do based on the current information you have (conditional 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



Intro to Python

- Author: https://en.wikipedia.org/wiki/Guido_van_Rossum
- Author: <https://gvanrossum.github.io//>
- Author: https://www.youtube.com/results?search_query=Guido+van+Rossum
- Overview: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- Python.org: https://en.wikipedia.org/wiki/Python_Software_Foundation
- Source code: <https://github.com/python/cpython>
- Open course: <https://www.coursera.org/courses?query=python&>
- Open course: <https://edu.goorm.io/lecture/44/%EB%B0%94%EB%A1%9C%EC%8B%A4%EC%8A%B5-%EC%83%9D%ED%99%9C%EC%BD%94%EB%94%A9-%ED%8C%8C%EC%9D%B4%EC%8D%AC-python>
- Open course: <https://programmers.co.kr/learn/courses/2>

Positioning of Python

- Tiobe index: <https://www.tiobe.com/tiobe-index/>
- GitHub user rank: <https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>
- GitHub repository statistics: <https://githut.info/>
- Source codes: <https://github.com/collections/programming-languages>

Open Sources?

- FYI: <http://mobilelab.khu.ac.kr/opensource/reference/>