

Fastcampus Web Programming SCHOOL

Python

Introduce

최우영

- Co-founder, Developer at Disceptio
- Solution Architect, Web Developer, Instructor
- Skills: Python, Golang, Julia, Node.js, Google tag manager ...

Contacts

- blog: <https://ulgoon.github.io/>
- github: <https://github.com/ulgoon/>
- email: me@ulgoon.com

Computational Thinking

Computer Science and Engineering

- 컴퓨터의 소프트웨어를 다루는 학문
- 컴퓨터라는 물리적 기기를 연구하는 것이 아닌 Computer 의 개념과 구조를 이해하고 구현하는 학문

Computation vs Calculation

"calculation" implies a strictly arithmetic process,
whereas "computation" might involve applying rules in a systematic way

Computer vs Calculator

- **Stored Program** computer -> Computer
 - Stores and Executes instructions
- **Fixed Program** computer -> Calculator
 - just calculate

엇? 그럼 공학용 계산기는???

Computational Thinking

정답이 정해지지 않은 문제에 대한 해답을 일반화하는 과정

Process of Computational Thinking

1. 문제 조직화(추상화) - Problem Formulation (abstraction)
2. 솔루션 구현(자동화) - Solution Expression (automation)
3. 솔루션 실행 및 평가(분석) - Solution Execution & Evaluation (analyses)

Characteristics of Computational Thinking

- 문제 분해(decomposition)
- 패턴인지 / 데이터 표현(pattern recognition / data representation)
- 일반화 / 추상화(generalization / abstraction)
- 알고리즘(algorithms)

Computational Thinking Process

- 문제인지
 - 배가 고프다

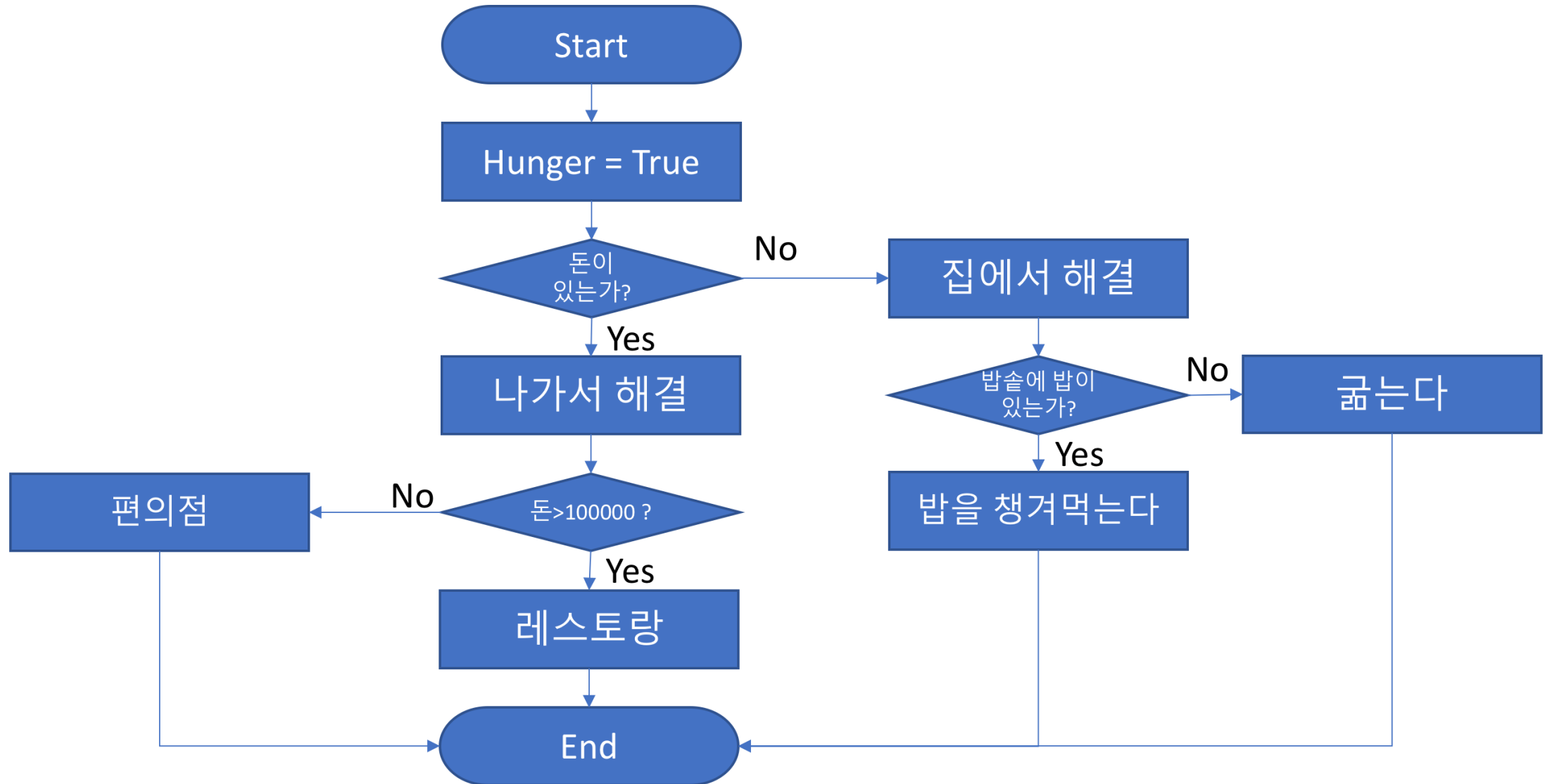
Computational Thinking Process

- 문제조직화
 - 문제분해
 - 뭘 먹긴 해야겠다
 - 집에서 해결함
 - 냉장고엔 뭐가있지? 밥은 해놨나? 라면이라도 먹을까? ...
 - 나가서 해결함
 - 편의점? 식당? 패스트푸드? 레스토랑??

Computational Thinking Process

- 패턴인지
 - 아! 배가고프면 어디서 뭔가를 먹음으로써 Hunger가 False가 되는구나
- 일반화/추상화
 - 추상화(간결하고 명확하게 단순화, 일반화, 개념화)
 - 배가 고프면 : {{어디}} 에서 {{어떻게}} 해결함
 - 알고리즘

Computational Thinking Process



Computational Thinking Process

- 솔루션구현
- 솔루션실행 및 평가
 - 솔루션대로 실행해서 나는 배고픔을 인지하고 해결하게 되었다.
 - 돈 보유량에 따라 다양한 선택지를 뒤야겠다
 - 집에서 밥이 없으면 굶지말고 밥을 해야겠다.

Python Basic

Python은?

1989년 크리스마스 연휴를 보내던 Guido van Rossum이 만든 고급 프로그래밍 언어

특징

- 인터프리터
- 객체지향
- 동적타이핑
- 엄격한 문법

C vs Python

```
int main(){  
    int i;  
    for(i=0;i<=10;i++){  
        if (i % 2 == 0){  
            printf(i);  
        }  
    }  
}
```

```
for i in range(1,10+1):  
    if i % 2 == 0:  
        print(i)
```

Python Basic

Python으로 할 수 있는 것들!

- System Programming
- Web Programming
- Data Analysis
- ...

Let's install Python!

<https://www.youtube.com/embed/AjGfUfW8njE>

MacOS

- install brew(brew.sh)
- `$ brew install python`

Windows

<https://www.python.org/ftp/python/3.9.1/python-3.9.1-amd64.exe>

Zen of Python - PEP 20

```
>>> import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better

Sparse is better than dense.

Readability counts.

Python Basic

REPL : Read - Eval - Print Loop

코드를 입력하면 바로 결과를 확인할 수 있음!!

```
>>> print("hello python!")  
hello python!
```

We'll use python3

difference of 2.x , 3.x

Short version: Python 2.x is legacy, Python 3.x is the present and future of the language

Jupyter Notebook

```
$ pip install jupyter  
$ pip list
```

```
$ jupyter notebook
```

Hello python!

So, let's try!!

```
print("hello python!")
```

Numbers & Math

<object> <operator> <object>

```
print(3 + 7)
print(10 - 3)
print(15 / 7)
print(34 * 100)
```


Numbers & Math

```
print(15 / 7)
print(15 / 5)
type(15 / 5)

print(15 // 5)
type(15 // 5)

print(7 % 3)

print(15 ** 3)

print(34 * 100)
print(3 * 2.5)
type(3 * 2.5)
```

Comparison

```
print(3 < 7)
print(10 < 3)
print(15 > 7)
print(3 >= 3)
print(3 <= 10)
print(34 == 100)
print(34 != 100)
```

Variable

```
print("hello python!")  
hello = "hello"  
python = "python!"  
print(hello, python)
```

```
num1 = 14  
num2 = 5  
  
print(num1+num2)  
print(num1-num2)  
print(num1*num2)  
print(num1/num2)
```

Practice(1)

반지름(`r=10`)을 선언한 뒤, 이를 이용하여 원의 지름, 둘레, 넓이, 구의 겉넓이, 부피를 각각 출력하는 파이썬 파일을 만들어보세요.(`pi=3.1415`)

sample output

```
r = 10 ==> print("r =", r)
d = 20
c = 62.830
a = 314.15
gnb = 1256.0000
v = 4188.666666666667
```

Toggl

<https://blog.toggl.com/wp-content/uploads/2016/12/toggl-it-jobs-explained-with-changing-lightbulb.jpg>

<https://assets.toggl.com/images/toggl-how-to-save-the-princess-in-8-programming-languages.jpg>

Let's Code PYTHONIC

Important Python Enhance Proposal

Layout

- 들여쓰기: 공백 4칸 or 탭(섞어쓰면 안됨)
- 한 줄은 79자(120자도 상관없음)
- 클래스정의와 최상위 함수는 두 줄을 띄움
- 클래스 내 메소드는 한 줄을 띄움

Important Python Enhance Proposal

Variables

- `_variable` : 내부적으로 사용되는 변수
- `print_` : 파이썬 키워드와 충돌 방지

Naming Convention

- 클래스 이름은 `CamelCase`
- 함수, 변수, 메소드 이름은 `snake_case`

파이썬에서 쓰이지 않는 네이밍 규칙

- `chHungarianNotation`
- `javaScriptStyleCamelCase`

Syntax

문법, 구조, 또는 언어 문장 내에 있는 구성요소의 순서

"나는 입니다 학생" (Syntax Error)

"나는 학생 입니다" (Syntactically Valid)

"Python"5 (Syntax Error)

3.6 * 12 (Syntactically Valid)

Data type

- int
- float
- long(0b, 0o, 0x)
- string
- boolean
- list, tuple, range
- set
- dictionary

operators

- arithmetic
 - `+, -, *, /, %, //, **`
- boolean
 - `and, or, not`
 - <https://docs.python.org/3/reference/expressions.html#boolean-operations>
- comparisons
 - `<, >, <=, >=`
 - `==, !=(value)`
 - `is, is not(object identity)`
 - `&, |, ~(bitwise) => 1000(8), 1010(10)`

Floating point

- 컴퓨터는 실수를 이진 부동소수점을 활용합니다.
- 0, 1!!
- $0.1 + 0.2 \neq 0.3$
→ `round(0.1+0.2, 1) == round(0.3, 1)`

type casting

`float(3) --> int to float`

`int(3.6) --> float to int`

`str(1) --> int to string`

`int("12") --> string to int`

input

```
name = input("What is your name? ")  
print("Hi, ", name)
```

input with evaluation

```
input("How old are you? ")  
eval(input("How old are you? "))
```

type casting with input

```
int(input("How old are you? "))
```


Refactoring Practice(1)

사용자의 입력을 받아 반지름(`r`)을 선언한 뒤, 이를 이용하여 원의 지름, 둘레, 넓이, 구의 겉넓이, 부피를 각각 출력하는 파이썬 파일을 만들어보세요. (`pi=3.1415`)

Strings

Strings

```
some_string = "python"  
len(some_string)
```

p	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

```
some_string[3:5] = "ho"  
some_string[1:5:2] = "yh"  
some_string[:] = some_string[0:len(some_string):1]  
some_string[::-1] = some_string[-1:-len(some_string):-1]  
some_string[::-1] = "nohtyp"
```

but, strings are **immutable**

```
>>> some_string = "python"

>>> some_string[0] = "c"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment

>>> some_string = "c" + some_string[1:]
```

String Functions

```
func = "python is easy programming language"
func.count('p')

func.find('p')

comma = ","
func = comma.join('python')

func.split(',')

python_is_easy = "python is easy"
python_is_easy.split()

python_is_easy.replace("python", "golang")
```

String Functions

```
some_string = "  computer  "  
some_string.strip()
```

```
some_string = ",,,Fastcampus..."  
some_string.strip(",")  
some_string.strip(".")
```

String Validator

```
str.isalnum()  
str.isalpha()  
str.isdigit()  
str.islower()  
str.isupper()
```

String Formatting - old way

```
print("I have a %s, I have an %s." % ("pen", "apple"))
```

```
%s - string  
%c - character  
%d - Integer(decimal)  
%f - floating-point  
%o - 8진수(Octal)  
%x - 16진수(hexadecimal)  
%% - %
```


String Formatting - New way

```
print("I have a {}, I have an {}".format("pen", "apple"))
```

```
print("I have a {0}, I have an {1}".format("pen", "apple"))
```

```
print("I have a {0}, I have an {0}".format("pen", "apple"))
```

padding and align

- `{:10}`
- `{:>10}`
- `{:^10}`
- `{:_^10}`

Final Practice

1. 사용자가 입력한 전화번호를 저장하려합니다. 전화번호의 구분자로 - 또는 를 혼합하여 사용할 때 이를 split과 join을 이용하여 - 로 통일하여 입력 받아 결과물을 출력하세요.
2. 사용자가 다음과 같은 입력을 할때, 그 자료가 알파벳 단독인지, 숫자 단독인지, 알파벳 숫자인지, 소문자인지, 대문자인지를 formating 하여 출력하세요.

```
FAST
campus12
FastCampus
1107
dynamite
```

이번 주 해야 할 일

1. <https://www.keybr.com/> 마스터하기
2. <https://www.hackerrank.com/> 가입하고 `python` 진도만큼 수행하기