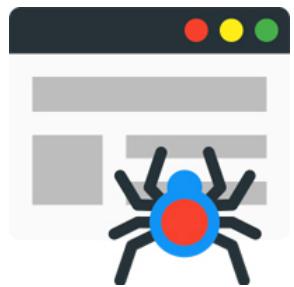


Intro to Computer Science

Local Laboratory

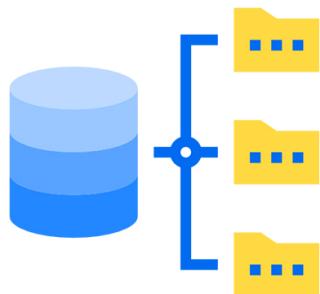
* Udacity – Intro to Computer Science

Build a Search Engine



Find Data

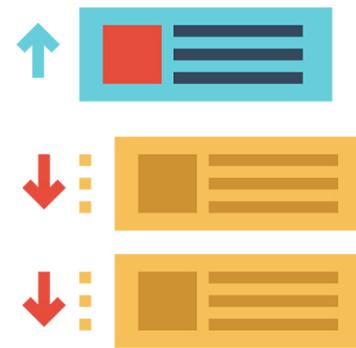
검색 엔진의 DB 데이터를
긁어올 웹크롤러를 만들어보자!



Build an index

자료를 저장하고 요청(query)에
빠르게 응답하는 구조 설계

자료구조

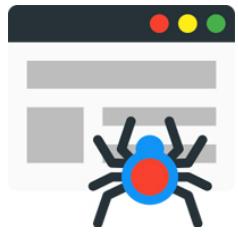


Rank Pages

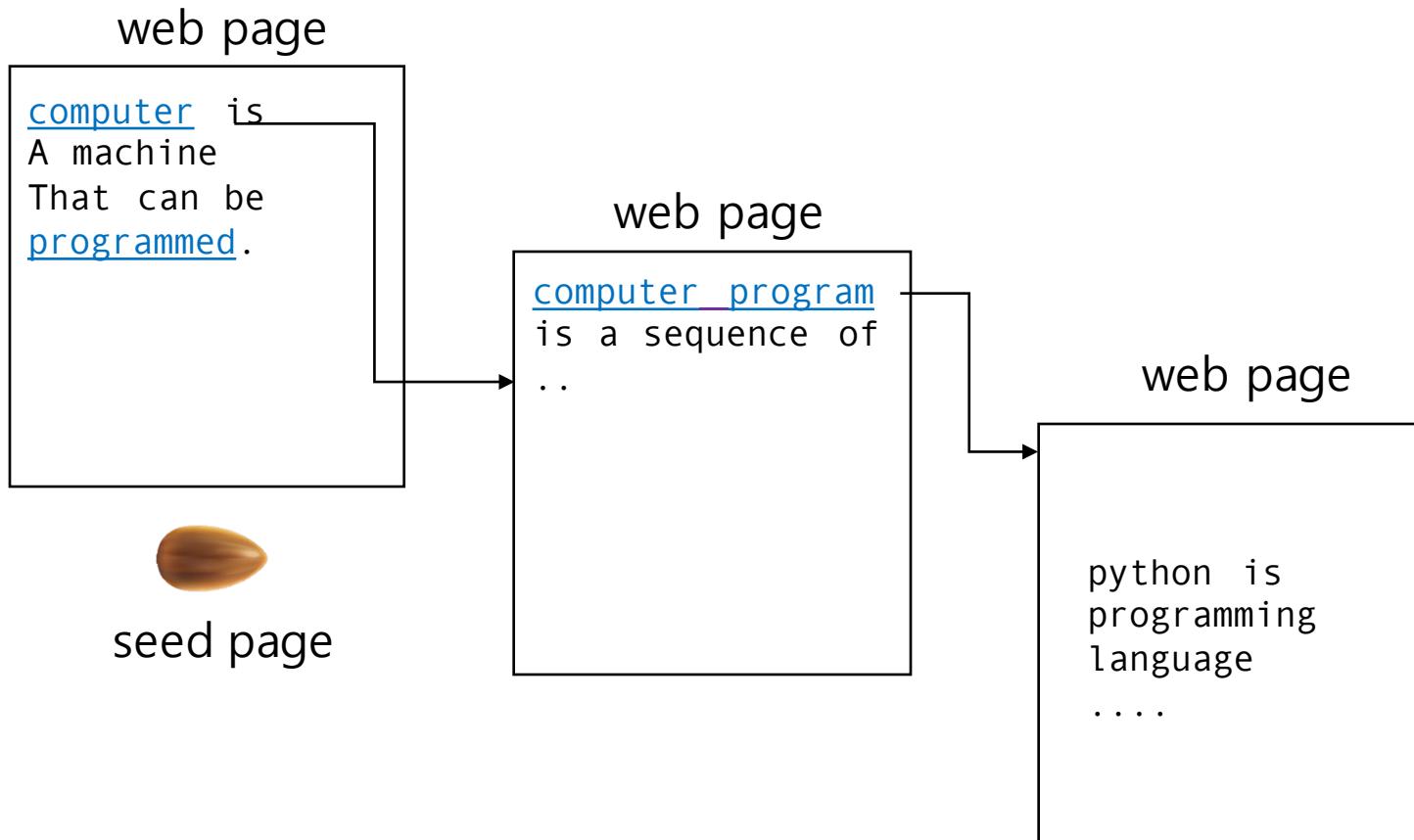
검색 결과에 정확도 등 순위를
매기는 방법론

알고리즘

Overview of the Unit



Find Data



Overview of the Unit

Unit 1:
Getting Started

HTML 코드에서 첫번째 링크 추출하기

What is Programming



Toster



Computer

Program

What is Programming



 python™ High Level Programming Language

Quiz : What is Programming

다음 중 컴퓨터 프로그램은 어떤 것인가?

1. a web browser
2. a slice of toast
3. the python interpreter
4. calendar app on a phone
5. the python code you will write in this class
`print("Hello!")`

First Programming

code

```
print( 3 )
print( 1 + 1 )
print( 52 * 3 + 12 * 9 )
print( (52 * 3) + (12 * 9) )
print( 52 * (3 + 12) * 9 )
print( 365 * 24 * 60 * 60 )
```

result

```
3
2
264
264
7020
31536000
```

Quiz : First Programming

[problemset/cs101_1_hello/1_first_programming.py]

7주의 기간이 몇 분인지 출력하는 파이썬 프로그램을 작성하시오.

```
# This is a Python comment. Lines that begin with a '#' are ignored by the
# Python interpreter. Comments are useful for documenting code or explaining
# quiz questions!
#
# Write a Python program that prints out the number of minutes in seven weeks.
# Remember: 7 weeks 7 days in a week, 24 hours in a day, and 60 mins in an hour.
# Multiplying these numbers together will give you the result
```

https://github.com/locallab-seoul/python_class

Congratulations



Dave – Udacity

Language Ambiguity

컴퓨터 프로그램을 만들 때,
왜 영어나 한국어 같은 자연언어를 쓰지 않고
Python과 같은 새로운 언어를 사용할까?

Ambiguity(모호성) Verbose(장황함)
biweekly?

Grammar

code

```
print ( 2 + 2 + )
```

result

```
File "test.py", line 1
  print(2+2+
        ^
SyntaxError: invalid syntax
```

Sentence → Subject Verb Object

Subject → Noun

Object → Noun

Verb → Eat

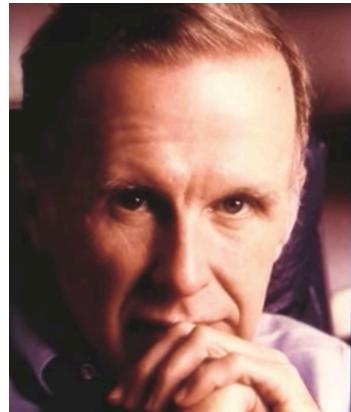
Verb → Like

Noun → I

Noun → Python

Noun → Cookies

Backus–Naur Form



John Backus
1924-2007

Lead designer of
Fortran programming
language (1950s)

Backus–Naur Form

Sentence → Subject Verb Object

Subject → Noun

Object → Noun

Verb → Eat

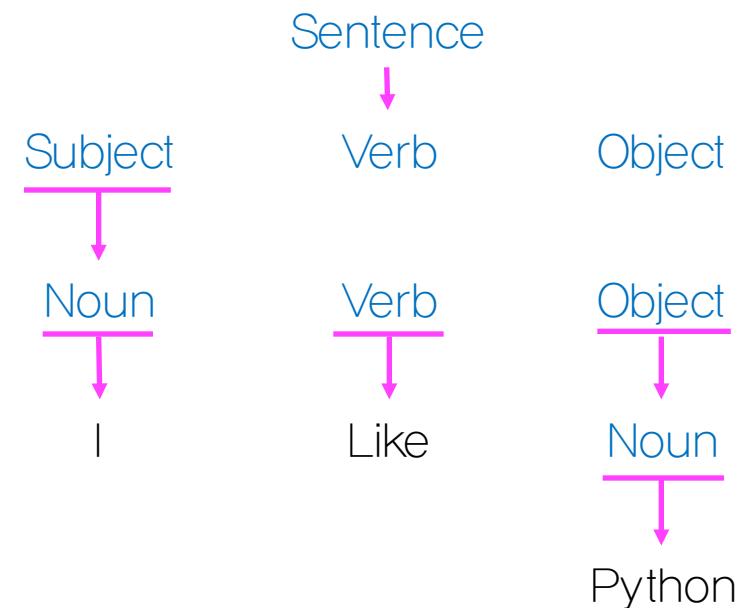
Verb → Like

Noun → I

Noun → Python

Noun → Cookies

Backus–Naur Form
⟨Non-terminal⟩ → replacement



Quiz : Backus–Naur Form

다음 문장 중 Sentence의 정의로 시작되어 생성될 수 있는 문장은?

1. Python Eat Cookies

2. Python Eat Python

3. I Like Eat

Sentence → Subject Verb Object

Subject → Noun

Object → Noun

Verb → Eat

Verb → Like

Noun → I

Noun → Python

Noun → Cookies

Python Expressions

수치 표현에 대한 파이썬 문법 (Python Grammar for Arithmetic Expressions)

Expression → Expression Operator Expression
Expression → Number
Operator → +
Operator → *
Number → 0, 1, ...
Expression → (Expression)

Quiz : Python Expressions

다음 표현 중 Expression의 정의로 시작되어 생성될 수 있는 올바른 파이썬 표현을 고르시오.

1. 3

2. (1 * (2 * (3 * 4)))

3. ((3)

4. + 33

5. (((7)))

Expression → Expression Operator Expression

Expression → Number

Operator → +

Operator → *

Number → 0, 1, ⋯

Expression → (Expression)

Quiz : Speed of Light

[problemset/cs101_1_hello/2_speed_of_light.py]

빛이 1 ns(nanosecond)에 몇 cm를 이동하는지를 출력하는 파이썬 코드를 작성하시오.

$$\text{빛의 속도} = 299,792,458 \text{ m / s}$$

$$1 \text{ m} = 100 \text{ cm}$$

$$1 \text{ ns} = 1 / 1000,000,000 \text{ s}$$

Processors

빛의 속도 = 299,792,458 m / s

meter = 100 cm

nanosecond = 1 / 1000,000,000 s

1 nanosecond에 빛이 이동 하는 거리 = 약 30 centimeters

2.7GHz = 2.7 Billion cycles per second
2.7 cycles per nanosecond

2.7GHz의 CPU가 한 cycle 동작할 때 빛이 이동하는 거리는?

code

```
print(299792458 * 100 * 1.0/1000000000 * 1/2.7)
```

result

```
11.1034243704
```

변수 (Variables)

Assignment Statement:

Name = Expression

speed_of_light = 299792458

speed_of_light → 299792458

code

```
speed_of_light = 299792458  
billionth = 1.0 / 1000000000  
nanostick = speed_of_light * billionth * 100  
print(nanostick)
```

result

```
29.9792458
```

Quiz : 변수 (Variables)

[problemset/cs101_1_hello/3_variables.py]

아래의 정의된 변수들을 이용하여, 프로세서가 1사이클 작동할 때 빛이 이동하는 거리를 cm 단위로 출력하는 파이썬 코드를 작성하시오.

```
speed_of_light = 299792458      # meters per second  
cycles_per_second = 2700000000. # 2.7 GHz
```

Variables can Vary

code

```
speed_of_light = 299792458      # meters per second
cycles_per_second = 27000000000 # 2.7 GHz

cycle_distance = speed_of_light / cycles_per_second

cycle_per_second = 2800000000. # 2.8 GHz
print(cycle_distance * 100)

cycle_distance = speed_of_light / cycles_per_second
print(cycle_distance * 100)
```

result

```
11.1034243704
10.7068735
```

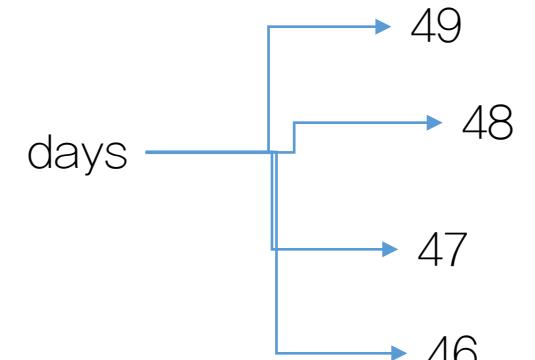
= assignment(할당)

days = 7 * 7

days = 48

days = days - 1

days = days - 1



Quiz : Varying Variables Quiz 1

아래의 코드를 실행 한 후 `hours`의 값은 무엇인가?

`hours = 9`

`hours = hours + 1`

`hours = hours * 2`

- | | |
|-------|----------|
| 1. 9 | 4. 10 |
| 2. 18 | 5. 20 |
| 3. 22 | 6. Error |

Quiz : Varying Variables Quiz 2

아래의 코드를 실행 한 후 seconds의 값은 무엇인가?

```
minutes = minutes + 1  
seconds = minutes * 60
```

- 1. 0 3. 120
- 2. 60 4. Error

Quiz : Spirit Age

[problemset/cs101_1_hello/4_spirit_age.py]

age라는 변수에 당신의 나이(년단위)를 정의하고,
당신이 살아온 날을 일단위로 출력하시오.

Intro to Computer Science

Local Laboratory

* Udacity – Intro to Computer Science

String – 문자열

↑
‘I am string’↑

↑
“I prefer double quotes!”↑

↑
“I’m happy I started with a double quote!”↑

String – 문자열

code

```
print( 'Hello' )  
print( "Hello" )  
print( Hello )
```

result

```
Hello  
Hello  
  
Traceback (most recent call last):  
  File "test.py", line 1, in <module>  
    print(Hello)  
NameError: name 'Hello' is not defined
```

String - 문자열

code

```
print( 'Hello' )  
print( "Hello" )  
Hello = "Howdy"  
print( Hello )
```

result

```
Hello  
Hello  
Howdy
```

Quiz : Valid String

아래 중 파이썬에서 유효한 문자열들을 고르시오.

- | | |
|----------|-----------|
| 1. “Ada” | 4. “Ada |
| 2. ‘Ada” | 5. Ada |
| 3. “Ada | 6. ““Ada’ |

Quiz : Hello!

[problemset/cs101_2_string/1_hello.py]

name이라는 변수를 정의하고, 이 변수에 자신의 이름을 문자열로 할당하시오.

$\langle \text{string} \rangle + \langle \text{string} \rangle \rightarrow$ concatenation of the two strings

Quiz : Strings and Numbers

code

```
name = 'Dave'  
print( 'Hello ' + name + '!' + '!' + '!' )  
print( 'My name is' + 9 )
```

result

Hello Dave!!!

Traceback (most recent call last):

File "test.py", line 1, in <module>
 print('My name is' + 9)

TypeError: must be str, not int

Strings and Numbers

code

```
name = 'Dave'  
print( 'Hello' + name + '!' + '!' + '!' )  
print( '!' * 12 )
```

result

```
Hello Dave!!!  
!!!!!!!!!
```

Indexing Strings – 문자열 색인

<string> [<expression>]

'udacity' [0] → 'u'
index : 0 1 2 3 4 5 6

'udacity' [1+1] → 'a'

name = 'Dave'
name[0] → 'D'

code

```
name = 'Dave'  
print( name[0] )
```

result

```
D
```

Indexing Strings – 문자열 색인

⟨string⟩ [⟨expression⟩]

‘udacity’
index : 0 1 2 3 4 5 6
[0] → ‘u’

‘udacity’ [1+1] → ‘a’

name = ‘Dave’
name[0] → ‘D’

code

```
name = 'Dave'  
print( name[3] )
```

result

```
e
```

Indexing Strings – 문자열 색인

⟨string⟩ [⟨expression⟩]

‘udacity’
index : 0 1 2 3 4 5 6
[0] → ‘u’

‘udacity’ [1+1] → ‘a’

name = ‘Dave’
name[0] → ‘D’

code

```
name = 'Dave'  
print( name[4] )
```

result

```
Traceback (most recent call last):  
File "test.py", line 3, in <module>  
    print(name[4])  
IndexError: string index out of range
```

Indexing Strings – 문자열 색인

⟨string⟩ [⟨expression⟩]

‘udacity’
index : 0 1 2 3 4 5 6
[0] → ‘u’

‘udacity’ [1+1] → ‘a’

name = ‘Dave’
name[0] → ‘D’

code

```
name = 'Dave'  
print( name[-1] )
```

result

```
e
```

Indexing Strings – 문자열 색인

⟨string⟩ [⟨expression⟩]

‘udacity’
index : 0 1 2 3 4 5 6
[0] → ‘u’

‘udacity’ [1+1] → ‘a’

name = ‘Dave’
name[0] → ‘D’

code

```
name = 'Dave'  
print( name[-2] )
```

result

↙

Quiz : Same Value

주어진 변수

$s = \langle \text{any string} \rangle$

다음의 두 쌍들 중에서 서로 완전히 같은 값을 가지는 쌍을 모두 고르시오.

1. $s[3]$, $s[1 + 1 + 1]$
2. $s[0]$, $(s + s)[0]$
3. $s[0] + s[1]$, $s[0 + 1]$
4. $s[1]$, $(s + \text{'ity'})[1]$
5. $s[-1]$, $(s + s)[-1]$

Selecting Sub Sequences – 문자열 부분집합

`<string>[<expression>]` → one-character
Number string

`<string>[<expression> : <expression>]`
`s start Number stop Number`

→ `start` index에서 시작하여,
`stop-1` index에서 끝나는 string 변수 `s` 의
부분집합을 가지는 string 타입의 값

Selecting Sub Sequences – 문자열 부분집합

code

```
word = 'assume'  
print( word[3] )  
print( word[3:4] )
```

result

```
u  
u
```

Selecting Sub Sequences – 문자열 부분집합

code

```
word = 'assume'  
print( word[3] )  
print( word[4:6] )
```

result

```
u  
me
```

Selecting Sub Sequences – 문자열 부분집합

code

```
word = 'assume'  
print( word[3:3] )
```

result

Selecting Sub Sequences – 문자열 부분집합

code

```
word = 'assume'  
print( word[3] )  
print( word[4:6] )  
print( word[4:] )  
print( word[:2] )  
print( word[:] )
```

result

```
u  
me  
me  
as  
assume
```

Quiz : Capital Udacity

[problemset/cs101_2_string/2_capital_udacity.py]

다음 변수를 이용하여 Udacity(대문자 U)를 출력하는 파이썬 코드를 작성하시오.

s = 'audacity'

code

```
s = 'audacity'  
print('U' + s[2:])
```

result

```
Udacity
```

Quiz : Understanding Selection

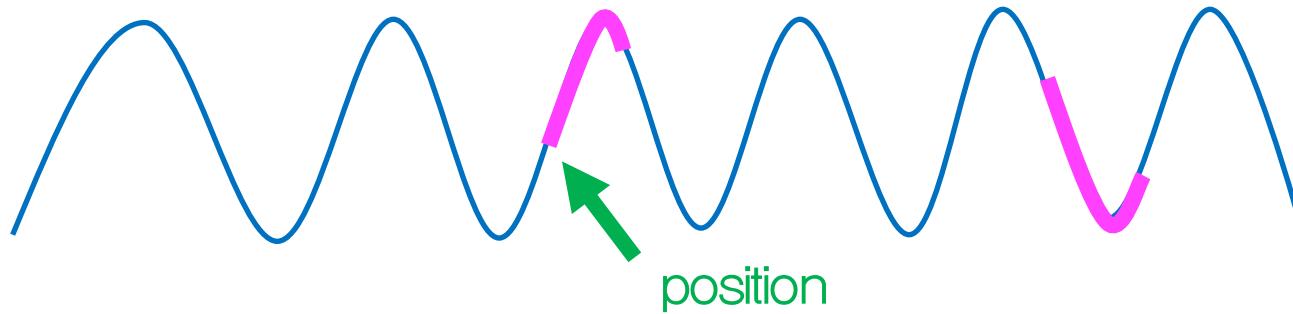
어떤 문자열 `s`에 대하여

`s = ‘⟨any string⟩’`

아래의 보기 중 `s`와 언제나 동일한 것들을 고르시오.

1. `s[:]`
2. `s + s[0:-1+1]`
3. `s[0:]`
4. `s[:−1]`
5. `s[:3] + s[3:]`

Finding Strings in Strings – 문자열 Find 메소드



`<string>.find(<string>)`

→ 타깃 문자열(target string)
이 검색 문자열(search string)에서 첫
번째로 나타나는 index인 position을
숫자 타입으로 반환

만약 타깃 문자열(target string)

검색되지 않으면,

-1 을 반환

Finding Strings in Strings – 문자열 Find 메소드

code

```
pythagoras = 'There is geometry in the humming of the strings, there is music in the spacing of the spheres.  
print( pythagoras.find( 'string' ) )  
print( pythagoras[40:] )  
print( pythagoras.find( 'T' ) )  
print( pythagoras.find( 'sphere' ) )  
print( pythagoras[86:] )  
print( pythagoras.find( 'algebra' ) )
```

result

```
40  
strings, there is music in the spacing of the spheres.  
0  
86  
spheres.  
-1
```

Quiz : Testing

다음에서 결과값이 **-1**이 되는 것을 모두 고르시오.

1. ‘test’.find(‘t’)
2. “test”.find(‘st’)
3. “Test”.find(‘te’)
4. ‘west’.find(‘test’)

Quiz : Testing 2

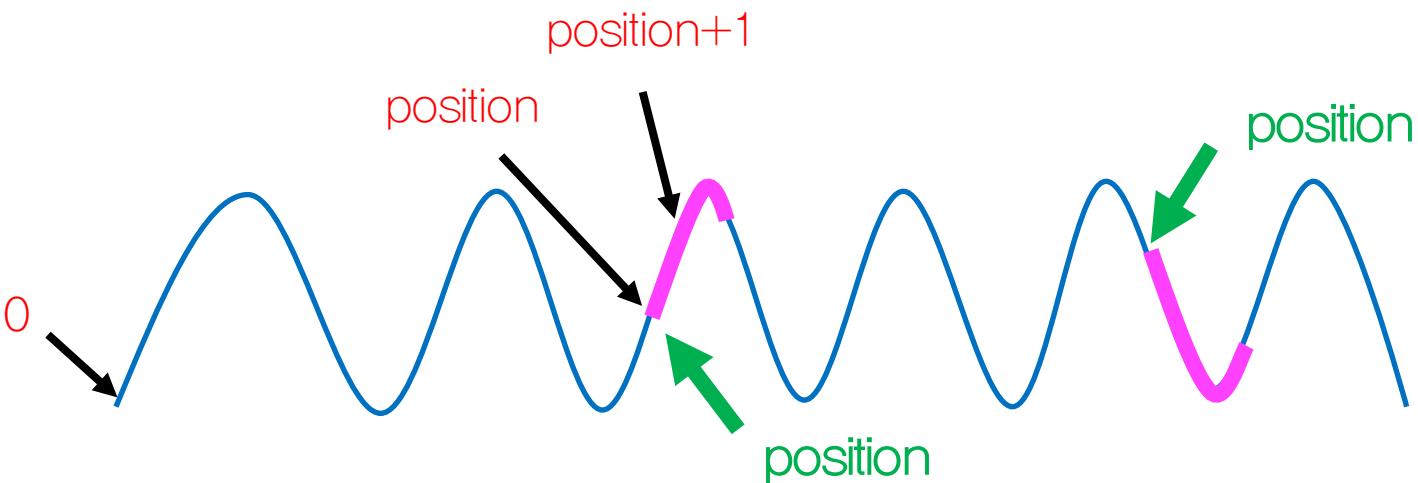
어떤 문자열 **s**에 대하여

s = ‘<any string>’

아래의 보기 중 항상 **0**의 값을 가지는 것을 모두 고르시오.

1. `s.find(s)`
2. `s.find('s')`
3. `'s'.find('s')`
4. `s.find("")`
5. `s.find(s + '!!!') + 1`

Finding with Numbers – 문자열 Find 메소드 2



`<string>.find(<string>) <number>`

→ 타깃 문자열(target string)
이 검색 문자열(search string)에서 첫 번째로 나타나는 위치 (position) 를 숫자 타입으로 반환

(<number> 값의 위치 또는 그 이후부터 검색)

만약 타깃 문자열(target string)
검색되지 않으면,

-1 을 반환

Finding with Numbers – 문자열 Find 메소드 2

code

```
danton = "De l' audace, encore de l' audace, toujours de l' audace."  
print( danton.find( 'audace' ) )  
print( danton.find( 'audace' , 0 ) )  
print( danton.find( 'audace' , 5 ) )  
print( danton.find( 'audace' , 6 ) )  
print( danton[6:] )  
print( danton[25:] )
```

result

```
5  
5  
5  
25  
udace, encore de l' audace, toujours de l' audace.  
audace, toujours de l' audace.
```

Finding with Numbers – 문자열 Find 메소드 2

code

```
danton = "De l' audace, encore de l' audace, toujours de l' audace."  
print( danton.find( 'audace' , 25 ))  
print( danton.find( 'audace' , 26 ))  
print( danton[47:] )  
print( danton.find( 'audace' , 48 ))
```

result

```
25  
47  
audance.  
-1
```

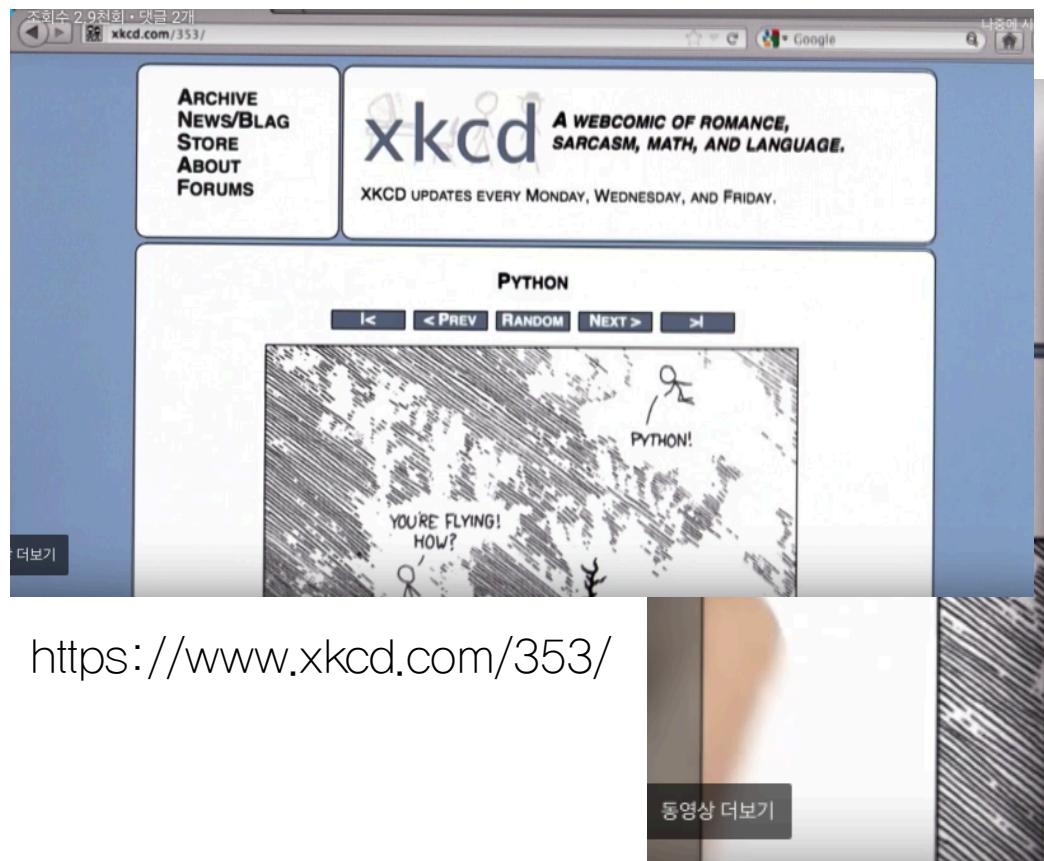
String Theory



Sebastian Thrun – Udacity President

David Evans – Udacity founder VP of Education

Extracting Links



<https://www.xkcd.com/353/>

```
Source of: http://xkcd.com/353/
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="http://imgs.xkcd.com/s/c40a9f8.css" type="text/css" media="scr
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>xkcd: Python</title>
  <link rel="stylesheet" type="text/css" href="http://imgs.xkcd.com/s/c40a9f8.css" me
<!--{if IE}--<link rel="stylesheet" type="text/css" href="http://imgs.xkcd.com/s/ecb
<link rel="alternate" type="application/atom+xml" title="Atom 1.0" href="/atom.xml"
<link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="/rss.xml" />
<link rel="icon" href="http://imgs.xkcd.com/s/919f273.ico" type="image/x-icon" />
<link rel="shortcut icon" href="http://imgs.xkcd.com/s/919f273.ico" type="image/x-i
</head>
<body>
  <div id="container">
    <div id="topContainer">
      <div id="topLeft" class="dialog">
        <div class="hd"><div class="c"></div></div>
        <div class="bd">
          <div class="c">
            <div class="s">
              <ul>
                <li><a href="/archive/">Archive</a><br /></li>
                <li><a href="http://blag.xkcd.com/">News/Blag</a><br /></li>
                <li><a href="http://store.xkcd.com/">Store</a><br /></li>
                <li><a href="/about/">About</a><br /></li>
                <li><a href="http://forums.xkcd.com/">Forums</a><br /></li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Extracting Links

... <a href = “<url>” > ...

page = web page의 소스코드의
string 값

Quiz : Extracting Links

[problemset/cs101_2_string/3_extracting_links.py]

page 변수 내용 중 첫번째 '<a href=' 의 위치값을 start_link 변수에 정의하는 파이썬 코드를 작성하시오.</p>

code

```
page = "<div id=top_bin> <div id=top_content class=width960>
    <div class=udacity float-left> <a href="/">""
start_link =
```

Quiz : Final Quiz

page 문자열에서 link 태그에 속한 첫번째 URL의 값을 url 변수에 할당하는 파이썬 코드를 작성하시오.

page = contents of a web page

start_link = page.find('<a href=')

print(url) → http://udacity.com

... <a href = “<url>” ...

↑
start_link

Quiz : Final Quiz

[problemset/cs101_2_string/4_final_quiz.py]

page 문자열에서 link 태그에 속한 첫번째 URL의 값을
url 변수에 할당하는 파이썬 코드를 작성하시오.

print(url) → http://udacity.com

code

```
page = "<div id='top_bin'> <div id='top_content' class='width960'>\n    <div class='udacity float-left'> <a href='http://udacity.com'>'</a>\n    </div>\n</div>"
```

```
print( url )
```

Great Job!



Intro to Computer Science

Local Laboratory

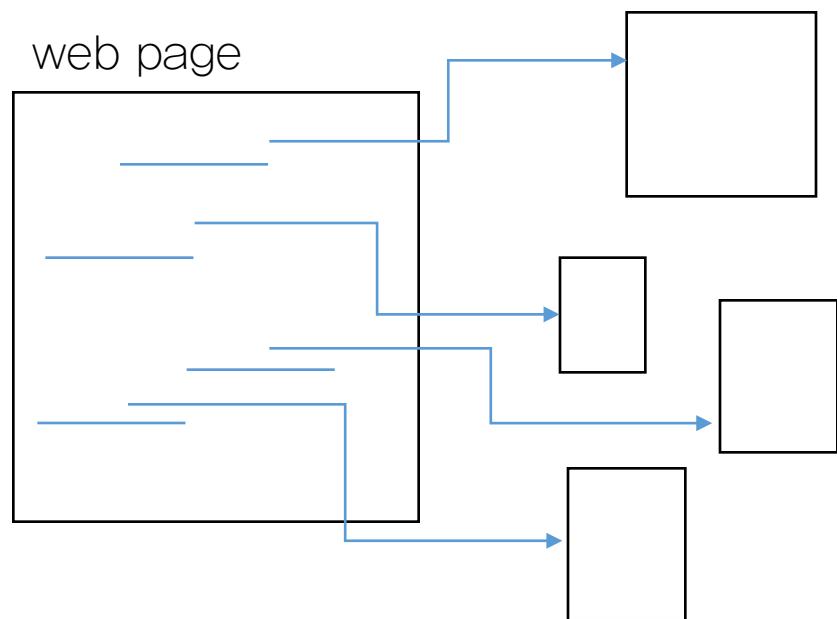
* Udacity – Intro to Computer Science

Unit 2:

How to Repeat

Procedures & Control

함수와 조건문, 반복문



```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
```

```
page = page[end_quote:]
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
```

```
page = page[end_quote:]
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
```

Motivating Procedures

```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
```

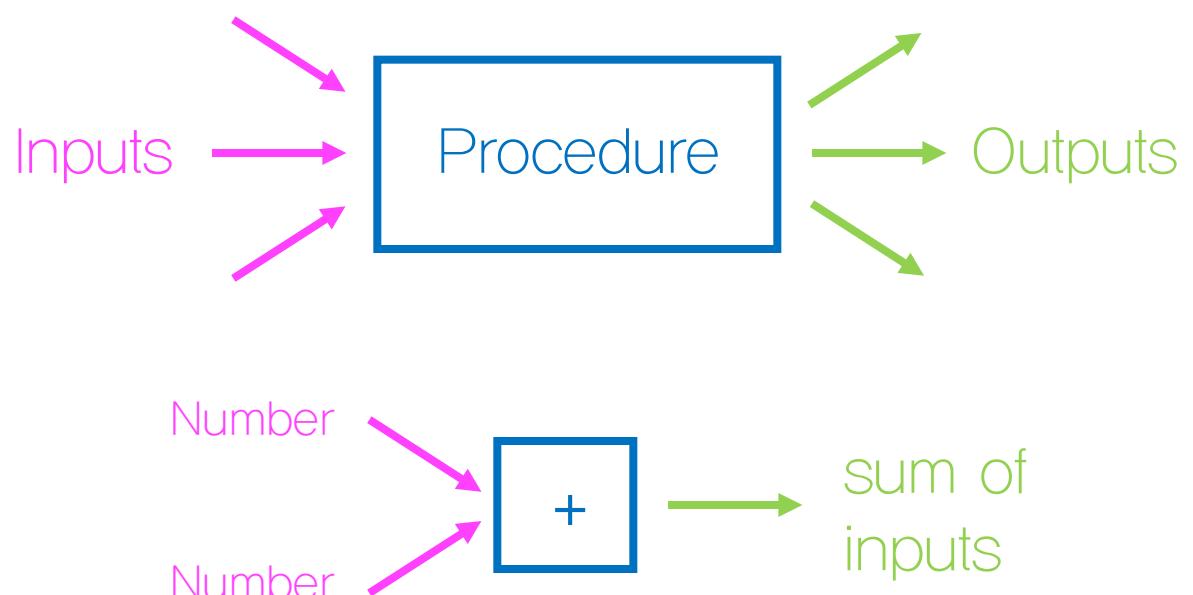
```
page = page[end_quote:]
```

```
start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
```

```
page = page[end_quote:]
```

```
start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

Introducing Procedures – 함수 소개 및 정의하기



```
def <name> (<parameters>):  
    <block>  
    ▲▲▲ tab ▲▲▲
```

inputs
()
(page)
(a, b, c, d, e)
→ <name>, <name>, ...

Procedure Code

procedure



```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find("“", start_link)
end_quote = page.find("”", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
start_link = page.find('<a href=')
start_quote = page.find("“", start_link)
end_quote = page.find("”", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
start_link = page.find('<a href=')
start_quote = page.find("“", start_link)
end_quote = page.find("”", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

Quiz : Procedure Code

아래 보기 중 `get_next_target` 프로시저를 위한 input 값은 어떤 것들이 되어야하는가?

1. 링크의 시작 위치에 대한 number 값
2. 마지막 따옴표의 시작 위치에 대한 number 값
3. 웹페이지 코드에서 이전 링크를 찾은 이후에 남은 내용에 대한 string 값
4. 링크의 시작 위치에 대한 number 값, **procedure** ← 페이지 내용에 대한 string 값

```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

Quiz : Output

아래 보기 중 `get_next_target` 프로시저를 위한 output 값들은 어떤 것들이 되어야하는가?

1. 다음 타켓 url에 대한 string 값(`url`)
2. `url, page`
3. `url, end_quote`
4. `url, start_link`

```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

procedure ←

```
start_link = page.find('<a href=')
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

Return Statement – 리턴문

```
page = ... contents of some web page ...
start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
[start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
[start_link = page.find('<a href=')
start_quote = page.find(" ", start_link)
end_quote = page.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

procedure

def get_next_target(s):

```
start_link = s.find('<a href=')
start_quote = s.find(" ", start_link)
end_quote = s.find(" ", start_quote + 1)
url = page[start_quote + 1 : end_quote]
url = page[start_quote + 1 : end_quote]
return <expression>, <expression>
```



Quiz : Return Statement

빈칸을 채우시오.

```
def get_next_target(page):
    start_link = page.find( '<a href=' )
    start_quote = page.find( '"', start_link)
    end_quote = page.find( '"', start_quote + 1)
    url = page[start_quote + 1 : end_quote]

    return [ ] , [ ]
```

Using Procedures – 함수 사용법

<procedure> (<input>, <input>, ...)

operands
arguments
parameters

code

```
def rest_of_string(s):
    return s[1:]
print(rest_of_string('udacity'))
```

The diagram illustrates the execution flow of the `rest_of_string` function. It shows the function definition and its call within a code block. Red arrows indicate the flow from the call to the definition and from the definition to the return statement. Blue boxes highlight specific parts of the code, such as the function name and arguments, and a variable assignment. Labels in pink text provide additional context for the highlighted terms.

Quiz : Inc Procedure

아래에 정의된 inc 프로시저는 보기 중 어떤 작업을 수행하는가?

```
def inc(n):  
    return n + 1
```

1. Nothing!
2. number 값을 입력으로 받아, 그 number 값에 1을 더한 값을 리턴한다.
3. number 값을 입력으로 받아, 동일한 number 값을 리턴한다.
4. 두개의 number 값을 입력으로 받아, 두 값의 합을 리턴한다.

Quiz : Sum Procedure

아래에 정의된 `sum` 프로시저는 보기 중 어떤 작업들을 수행하는가?

```
def sum(a, b):  
    a = a + b
```

1. Nothing!!
2. 두개의 number 값을 입력으로 받아, 그 값들의 합을 리턴한다.
3. 두 string 값을 입력으로 받아, 두 string 값의 결합 값을 리턴한다.
4. 두개의 number 값을 입력으로 받아, 첫번째 입력 값을 두 number 값의 합으로 교체한다.

Sum Procedure

code

```
def sum(a, b):  
    a = a + b  
  
print( sum(1, 1) )
```

code

```
def sum(a, b):  
    print( "enter sum!" )  
    print( "a is " , a )  
    a = a + b  
    print( "a is " , a)  
  
print( sum(2, 123) )
```

code

```
def sum(a, b):  
    a = a + b  
    return a  
  
print( sum(2, 123) )
```

result

```
None
```

result

```
enter sum!  
a is 2  
a is 125  
None
```

result

```
125
```

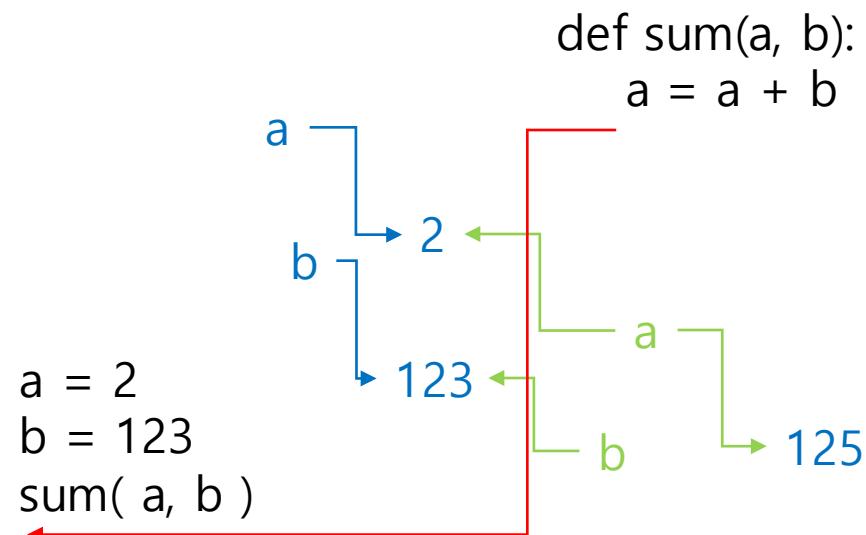
Sum Procedures

code

```
def sum(a, b):  
    a = a + b  
  
a = 2  
b = 123  
print( sum(a, b) )  
print( a )
```

result

```
None  
2
```



Quiz : Sum Procedure with Return Statement

아래에 정의된 `sum` 프로시저는 보기 중 어떤 작업들을 수행하는가?

```
def sum(a, b):  
    a = a + b  
    return a
```

- 1. Nothing
- 2. 두개의 number 값을 입력으로 받아, 그 값들의 합을 출력한다.
- 3. 두 string 값을 입력으로 받아, 두 string 값의 결합 값을 출력한다.
- 4. 두개의 number 값을 입력으로 받아, 첫번째 입력 값을 두 number 값의 합으로 교체한다.

Sum Procedure with Return Statement

code

```
def sum(a, b):  
    a = a + b  
    return a  
  
s = 'Hello '  
t = 'Dave!'  
print( sum(s, t) )  
print( s )
```

result

```
Hello Dave!  
Hello
```

Quiz : Square

[problemset/cs101_3_procedure/1_sqre.py]

`square` 라는 이름의 프로시저를 정의하여, 하나의 숫자 입력을 받아서, 그 숫자의 제곱을 출력하게 하시오.

`print(square(5)) → 25`

```
# Define a procedure, square, that takes one number
# as its input, and returns the square of that
# number (result of multiplying
# the number by itself).
# To help you out, the code for sum(a,b) is below.

def sum(a, b):
    c = a + b
    return c

# print(square(5))
# >>> 25
```

Square

code

```
def square(n):  
    return n * n  
  
x = 37  
print( square(x) )
```

result

```
1369
```

Square

code

```
def square(n):  
    return n * n  
  
x = 37  
y = square(x)  
print( square(y) )
```

result

```
1874161
```

Sum Procedure

code

```
def square(n):  
    return n * n  
  
x = 37  
print( square(square(x)) )  
procedure composition
```

result

```
1874161
```

Quiz : Sum of Three

[problemset/cs101_3_procedure/2_sum_of_three.py]

sum3라는 이름의 프로시저를 정의하여, 세개의 입력을 받아서,
세개의 숫자들의 합을 출력하게 하시오.

```
print( sum3(1,2,3) ) → 6
```

Quiz : Abbaize

[problemset/cs101_3_procedure/3_abbaize.py]

`abbaize`라는 이름의 프로시저를 정의하여, 두개의 문자열을 입력으로 받고, 첫번째 입력값, 두번째 입력값의 두번 반복, 다시 첫번째 입력값 순서대로 합친 문자열을 출력하도록 하시오.

`abbaize('a' , 'b') → 'abba'`
`abbaize('dog' , 'cat') → 'dogcatcatdog'`

Quiz : Find Second

[problemset/cs101_3_procedure/4_find_second.py]

`find_second`라는 이름의 프로시저를 정의하여,
두개의 문자열(검색 대상 문자열, 검색할 문자열)을 입력으로 받으시오.
검색할 문자열이 검색 대상 문자열에서 두번째로 나타나는 위치를 출력하시오.

code

```
danton = "De l' audace, encore de l' audace, toujours de l' audace"  
print( find_second(danton, 'audace' ) )
```

result

```
25
```

Intro to Computer Science

Local Laboratory

* Udacity – Intro to Computer Science

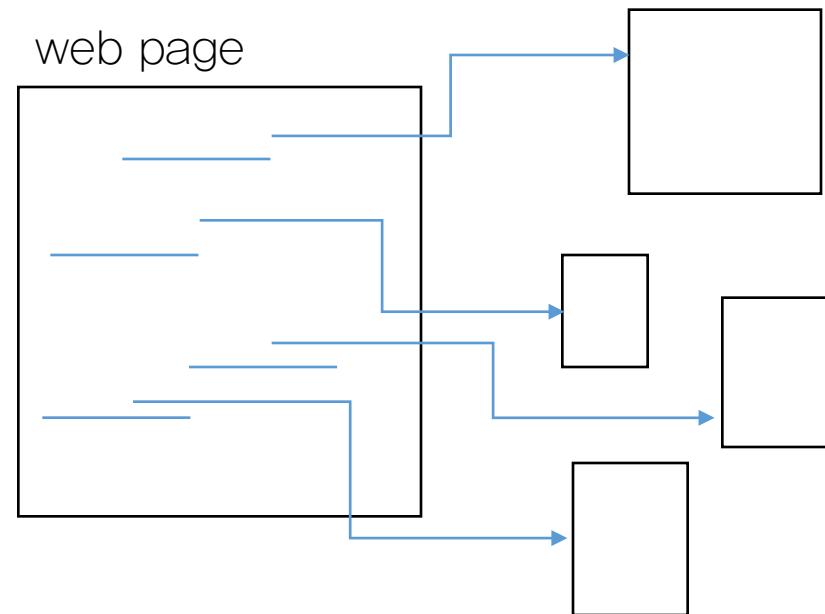
INTRODUCTION

Unit 2:

How to Repeat

Procedures & Control

함수와 조건문, 반복문



Comparisons – 비교 표현

< > <= ...

⟨Number⟩ ⟨Operator⟩ ⟨Number⟩
⟨숫자⟩ ⟨비교연산자⟩ ⟨숫자⟩

Boolean value : True False

code

```
print( 2 < 3 )
print( 21 < 3 )
print( 7 * 3 < 21 )
```

```
print( 7 * 3 != 21 )
print( 7 * 3 == 21 )
```

result

```
True
False
False
False
True
```

Quiz : Equality Comparisons

왜 동일 비교 시, `=` 대신 `==`를 사용할까?

1. `=` 는 대략적으로 같다는 것을 뜻하기 때문이다.
2. 같지 않다는 것을 `!=`로 표현하기 때문이다.
3. Guido가 `=`를 좋아하지 않기 때문이다.
4. `=`는 할당의 뜻을 의미하기 때문이다.
5. 사실 `==` or `=` 둘다 사용이 가능하다.

`i = 21`

→ 변수 i에 숫자 값 21을 할당한다

`i == 21`

→ True or False

If Statements – 조건문

if <비교 표현 or Boolean>:
 <Block>

```
def absolute( x ):  
    if x < 0:  
        x = -x  
    return x
```

[problemset/cs101_4_control/1_if_statements.py]

두 숫자를 입력받아, 둘 중 더 큰 숫자를 출력하는
`bigger`라는 프로시저를 정의하시오.

`bigger(2, 7) → 7`
`bigger(3, 2) → 3`
`bigger(3, 3) → 3`

If Statements – 조건문

```
if <비교 표현 or Boolean>:  
    <Block>  
else:  
    <Block>
```

Quiz : Is Friend

[problemset/cs101_4_control/2_is_friends.py]

문자열을 입력(친구의 이름)받아 Boolean을 출력(친구가 맞는지 아닌지)하는
`is_friend` 라는 프로시저를 정의하시오. (이름이 ‘D’로 시작하는 사람을 친구로 가정)

```
print( is_friend( 'Diane' ) ) → True
print( is_friend( 'fred' ) ) → False
```

Quiz : More Friends

[problemset/cs101_4_control/3_more_friends.py]

문자열을 입력(친구의 이름)받아 Boolean을 출력(친구가 맞는지 아닌지)하는
`is_friend` 라는 프로시저를 정의하시오. (이름이 ‘D’ 또는 ‘N’ 시작하는 사람을 친구로 가정)

```
print( is_friend( 'Diane' ) ) → True  
print( is_friend( 'Ned' ) ) → True
```

More Friends

code

```
def is_friend(name):
    if name[0] == 'D' :
        return True
    if name[0] == 'N' :
        return True
    return False

print( is_friend( 'Doug' ) )
print( is_friend( 'Nicole' ) )
print( is_friend( 'Fred' ) )
```

result

```
True
True
False
```

code

```
def is_friend(name):
    if name[0] == 'D' :
        return True
    else:
        if name[0] == 'N' :
            return True
        else:
            return False

print( is_friend( 'Doug' ) )
print( is_friend( 'Nicole' ) )
print( is_friend( 'Fred' ) )
```

result

```
True
True
False
```

Or - 논리연산자

code

```
def is_friend(name):
    return name[0] == 'D' or name[0] == 'N'

print( is_friend( 'Doug' ) )
print( is_friend( 'Nicole' ) )
print( is_friend( 'Fred' ) )
```

code

```
print( True or False )
print( False or True )
print( True or True )
print( False or False )
```

result

```
True
True
False
```

result

```
True
True
True
False
```

Quiz : Biggest

[problemset/cs101_4_control/4_biggest.py]

3개의 숫자를 입력으로 받아서, 세 숫자 중 가장 큰 수를 출력하는
`biggest`라는 프로시저를 정의하시오.

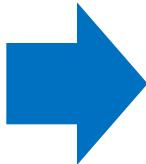
`biggest(6, 2, 3) → 6`

`biggest(6, 2, 7) → 7`

`biggest(6, 9, 3) → 9`

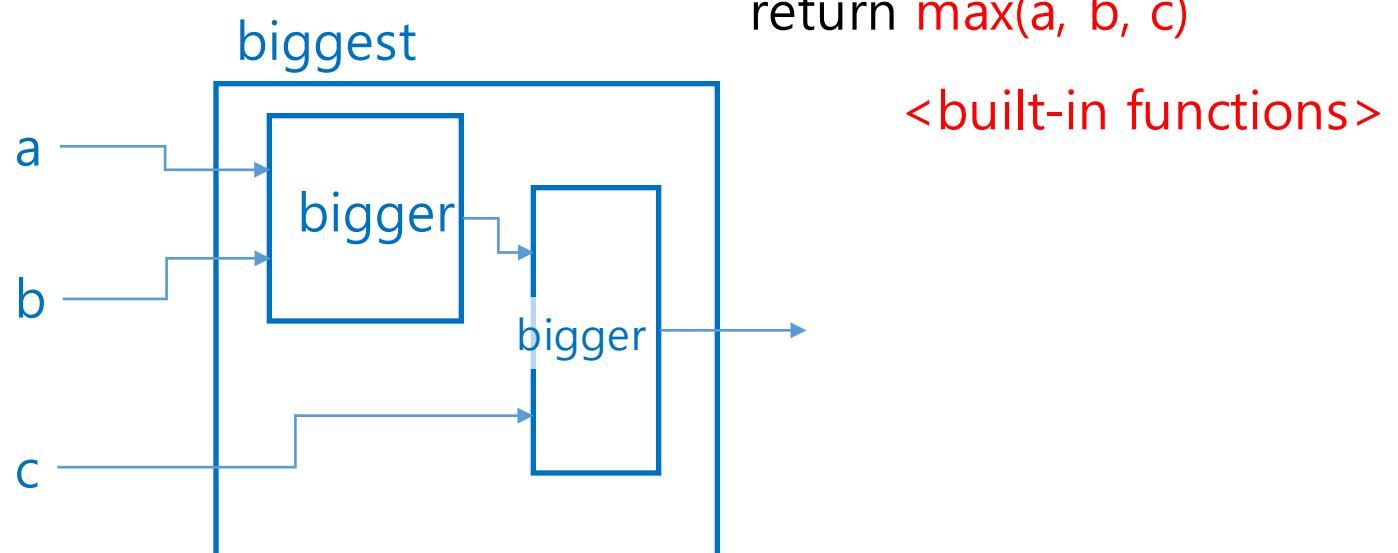
Biggest

```
def biggest(a, b, c):
    if a > b:
        if a > c:
            return a
        else:  
c: >= a > b
            return c
    else:  
b: >= a
        if b > c:
            return b
        else:  
c: >= b >= a
            return c
```



```
def bigger(a, b):
    if a > b:
        return a
    return b
```

```
def biggest(a, b, c):
    return bigger( bigger(a, b), c )
```



Biggest

arithmetic
comparisons
procedures
if

Alan Turing 1912-1954

Developed abstract model
of a computer (1936)
("Turing Machine")

Proved that a machine with
a few simple operations
could simulate any other
machine



You now know enough
(in theory!)
to write every possible
computer program!

Loops

`while <Boolean>:` 0, 1, 2, ... times
 `<Block>`

`if <Boolean>:` 0 or 1 times
 `<Block>`

Quiz : While Loops

아래 프로그램은 어떤 동작을 할지 보기에서 고르시오.

```
i = 0  
while i != 10:  
    i = i + 1  
    print( i )
```

1. 에러 발생
2. 0부터 9까지의 숫자를 출력한다
3. 1부터 9까지의 숫자를 출력한다
4. 1부터 10까지의 숫자를 출력한다
5. 영원히 실행된다

Quiz : While Loops 2

아래 프로그램은 어떤 동작을 할지 보기에서 고르시오.

```
i = 1  
while i != 10:  
    i = i + 2  
    print( i )
```

1. 에러 발생
2. 숫자 2, 4, 6, 8 출력
3. 숫자 1, 3, 5, 7, 9 출력
4. 숫자 3, 5, 7, 9 출력
5. 영원히 실행

Quiz : Print Numbers

[problemset/cs101_4_control/5_print_numbers.py]

모든 양의 숫자 입력에 대해, 1부터 입력 숫자까지의 모든 숫자를 출력하는
`print_numbers`라는 프로시저를 정의하시오.

`print_numbers(3)`

1

2

3

Quiz : Print Numbers

code

```
def print_numbers(n):
    i = 1
    while i <= n:
        print( i )
        i = i + 1

print_numbers(3)
```

result

```
1
2
3
```

code

```
def print_numbers(n):
    i = 0
    while i < n:
        i = i + 1
        print( i )

print_numbers(3)
```

result

```
1
2
3
```

code

```
def print_numbers(n):
    i = 0
    while i < n:
        i = i + 1
        print( i )

print_numbers(0)
```

result

Quiz : Factorial

[problemset/cs101_4_control/6_factorial.py]

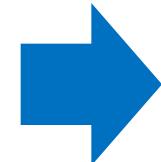
하나의 숫자를 입력받고 그 수의 팩토리얼 값을 출력하는
`factorial` 이라는 프로시저를 정의하시오.

$$\text{factorial}(n) = n * (n-1) * (n-2) * \dots * 2 * 1$$

Break

```
while <Boolean>:  
    <Code>  
    if <Boolean>:  
        break  
    <More Code>  
  
<After While>
```

```
def print_numbers(n):  
    i = 1  
    while i <= n:  
        print( i )  
        i = i + 1
```



```
def print_numbers(n):  
    i = 1  
    while True:  
        if i > n:  
            break  
        print( i )  
        i = i + 1
```

Quiz : Break

보기 중 다음의 표현과 언제나 같은 표현을 가지는 것을 모두 고르시오.

while <T>:
 ⟨S⟩

while <T>:
 if False:
 break
 ⟨S⟩

while True:
 if <T>:
 break
 ⟨S⟩

while <T>:
 ⟨S⟩
 if <T>:
 ⟨S⟩
 else:
 break

while <T>:
 ⟨S⟩
 break

Multiple Assignment – 다중 할당

```
def get_next_target(page):
    start_link = page.find( '<a href=' )
    start_quote = page.find( '"', start_link )
    end_quote = page.find( '"', start_quote + 1 )
    url = page[start_quote + 1 : end_quote]
    return url, end_quote
```

〈변수명〉 = 〈Expression〉

Multiple Assignment

`<변수_1>, <변수_2>, ... = <Expression_1>, <Expression_2>, ...`

Quiz : Multiple Assignment

다음 표현의 결과는 어떻게 될지 예측해보시오.

$s, t = t, s$
2 1

$s \rightarrow 1$
 $t \rightarrow 2$

$s = t$
 $t = s$

- Nothing
- s와 t 모두를 원래의 t 값으로 할당한다
- s와 t값을 변경한다
- Error

Quiz : No Links

[problemset/cs101_4_control/7_no_links.py]

아래의 코드는 이전에 정의했던 get_next_target 함수이다. 타당한 링크가 문자열에 포함되었을 경우, 이 함수는 링크와 마지막 따옴표의 위치를 리턴한다.

**문자열에 링크가 없을 경우를 검출하기 위해 함수를 수정하시오.
(만약 링크가 문자열에 없다면, 함수는 아래 내용을 출력해야한다.)**

None, 0

code

```
def get_next_target(page):
    start_link = page.find('<a href=')

    #Insert your code below here

    start_quote = page.find('"', start_link)
    end_quote = page.find('"', start_quote + 1)
    url = page[start_quote + 1:end_quote]
    return url, end_quote
```

Print All Links

```
page = ... contents of some web page ...
start_link = page.find( '<a href=' )
start_quote = page.find( '"', start_link)
end_quote = page.find( '"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
start_link = page.find( '<a href=' )
start_quote = page.find( '"', start_link)
end_quote = page.find( '"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
start_link = page.find( '<a href=' )
start_quote = page.find( '"', start_link)
end_quote = page.find( '"', start_quote + 1)
url = page[start_quote + 1 : end_quote]
print(url)
page = page[end_quote:]
```

```
def get_next_target(page):
    start_link = page.find('<a href=')
```

```
if start_link == -1:
    return None, 0
```

```
start_quote = page.find('"', start_link)
end_quote = page.find('"', start_quote + 1)
url = page[start_quote + 1:end_quote]
return url, end_quote
```

Quiz : Print All Links

[problemset/cs101_4_control/8_print_all_links.py]

`print_all_links`는 `page`의 모든 링크를 출력하는 프로시저이다.
`page`의 모든 링크를 출력할 수 있도록 빈칸을 채우시오.

```
def print_all_links(page):
    while :
        url, endpos = get_next_target(page)
        if url:
            print( url )
            page = page[endpos:]
        else:
            
```