이번주엔 뭐하나?

채팅!!

채팅!!

1일차 : 회원가입/로그인

채팅！！

1일차 ： 회원가입/로그인

2일차 ： 로그인 유지 （세션）

채팅！！

1일차 ： 회원가입/로그인

2일차 ： 로그인 유지 （세션）

3일차 ： 채팅 구현

채팅！！

1일차 ： 회원가입/로그인

2일차 ： 로그인 유지 （세션）

3일차 ： 채팅 구현

코드카데미

채팅!!

1일차 : 회원가입/로그인

2일차 : 로그인 유지 (세션)

3일차 : 채팅 구현

코드카데미 - 자바스크립트

그럼 오늘은 뭐하나?

패턴 매칭

cmd & Termianl

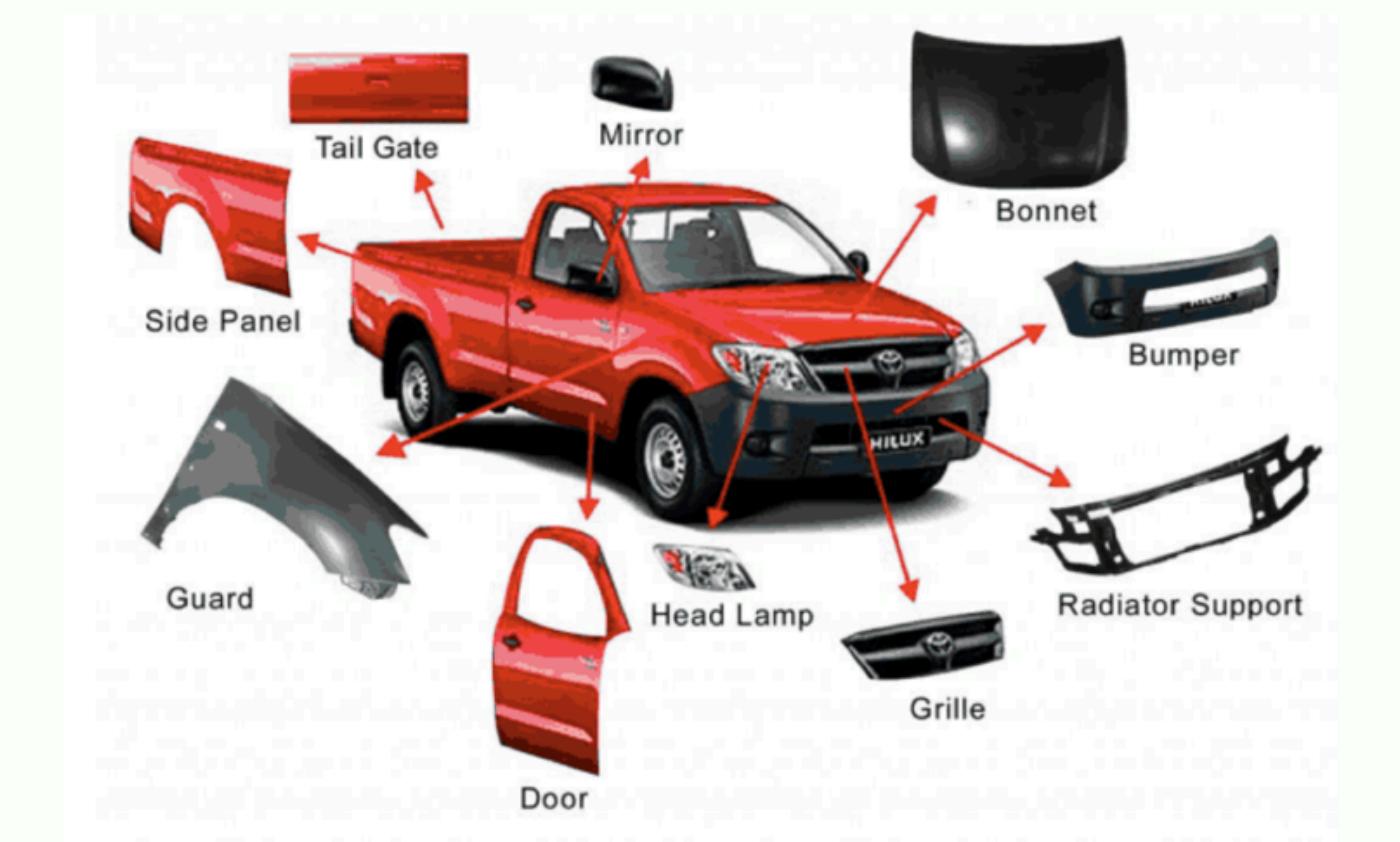
Flask Tutorial

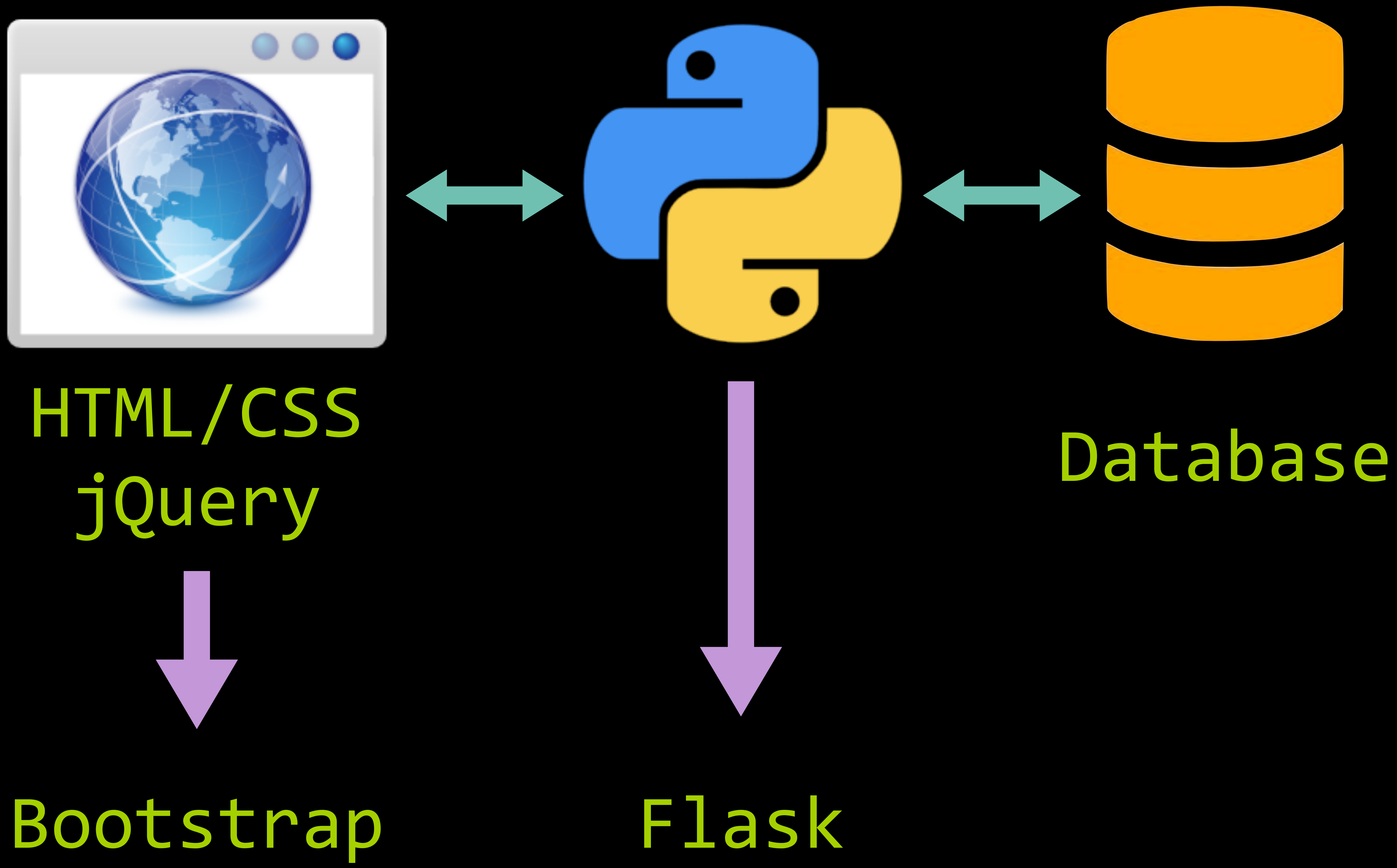
HTTP Error

GET & POST

Flask?

Flask?
어따쓰는 물건이고?


Tail Gate
Mirror
Bonnet
Side Panel
Bumper
HILUX
Guard
Radiator Support
Head Lamp
Grille
Door

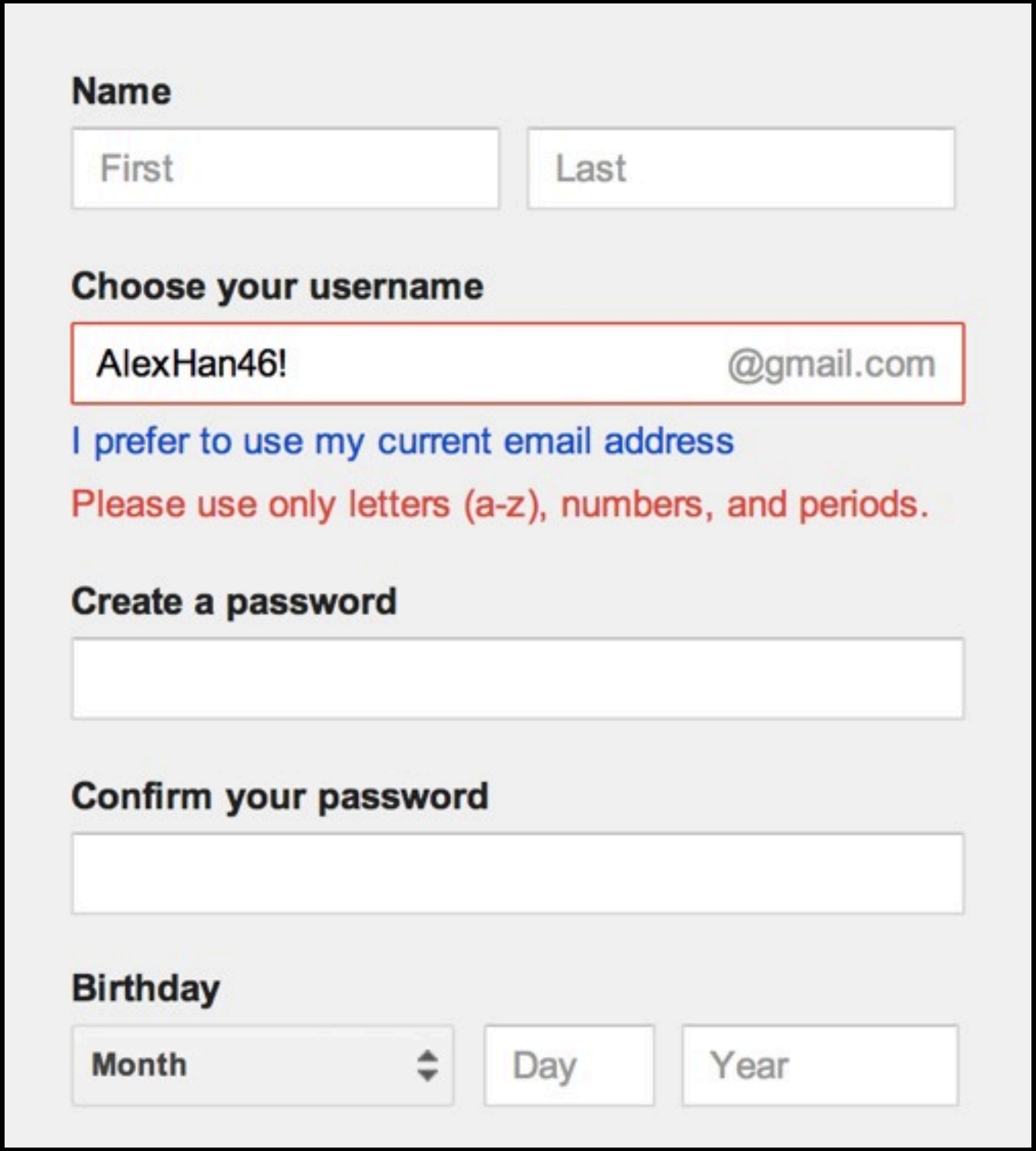


HTML/CSS
jQuery

Bootstrap

Flask

Database



Name

First | Last

Choose your username

AlexHan46! @gmail.com

I prefer to use my current email address

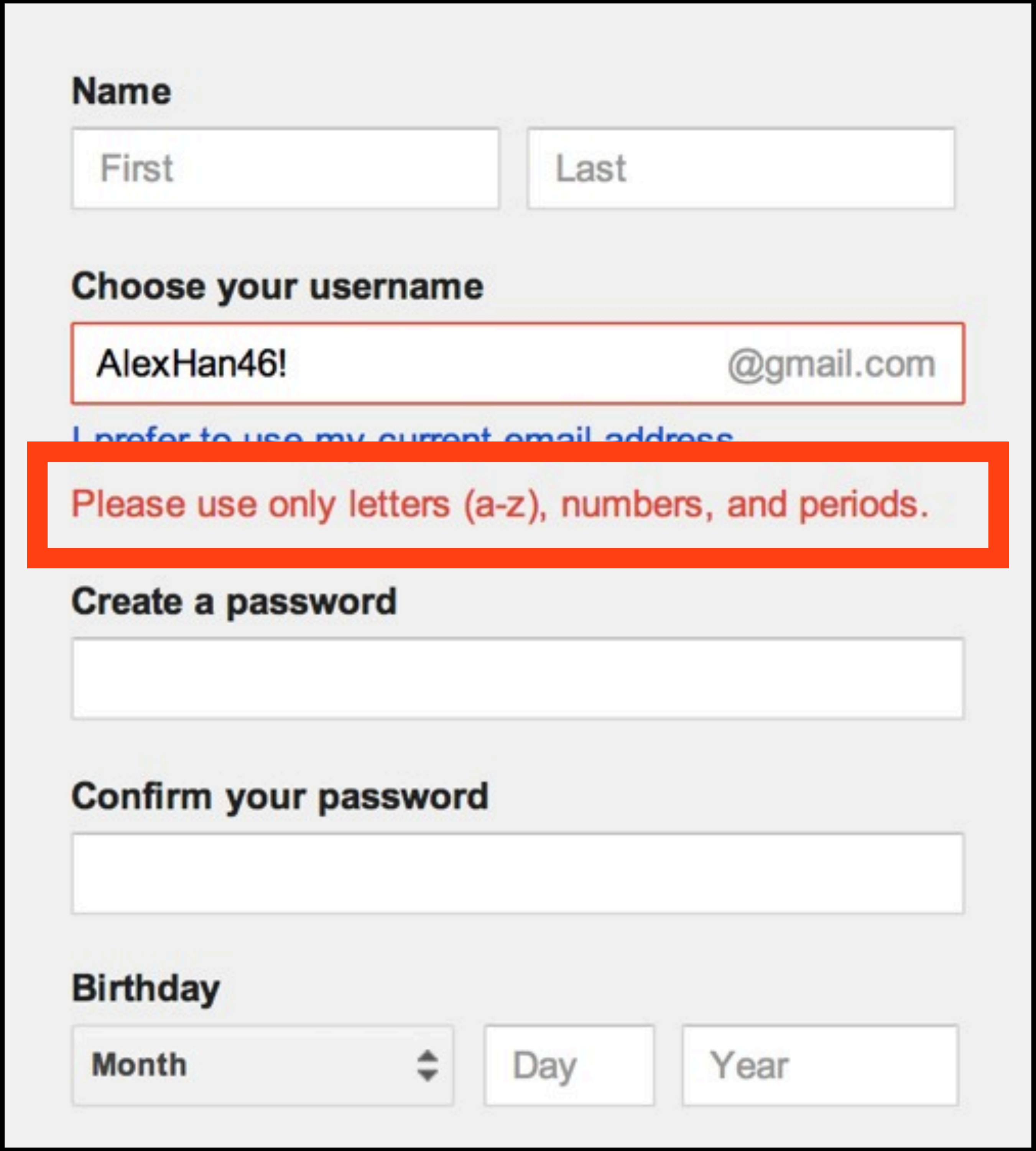
Please use only letters (a-z), numbers, and periods.

Create a password

Confirm your password

Birthday

Month | Day | Year



Name

First Last

Choose your username

AlexHan46! @gmail.com

I prefer to use my current email address

Please use only letters (a-z), numbers, and periods.

Create a password

Confirm your password

Birthday

Month Day Year

Between 3 ~ 20 characters

Between 3 ~ 20 characters

Use letters, numbers, and periods

Between 3 ~ 20 characters

Use letters, numbers, and periods

No special characters

Between 3 ~ 20 characters

Use letters, numbers, and periods

No special characters

No uppercase

```
username = “AlexHan46!”
username = “AlexHan46!”

for i in username:
```

```
username = “AlexHan46!”

for i in username:

    if not i.isalnum()
```

```
username = “AlexHan46!”

for i in username:

    if not i.isalnum()

        if i != “.”
```

```
username = “AlexHan46!”

for i in username:

    if not i.isalnum()

        if i != “.”

        if i.isupper()
```

```
re.match("[a-z0-9.]{6,30}", username)
```

```
[a-z0-9.]{6,30}
```

```
[a-z0-9.]{6,30}
```

Regex

```
[a-z0-9.]{6,30}
```

RegEx

= Reg Ex

```
[a-z0-9.]{6,30}
```

RegEx

= Regular Expression

```
[a-z0-9.]{6,30}
```

RegEx

= Regular Expression

= 정규표현식

```
[a-z0-9.]{6,30}
```

RegEx

```
[a-z0-9.]{6,30}
```

RegEx

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

[ ]: 괄호안의 조건을 만족하는 문자 하나

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

[ ]: 괄호안의 조건을 만족하는 문자 하나

a-z : a 에서 z 까지

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

```
[     ]: 괄호안의 조건을 만족하는 문자 하나
a-z    : a 에서 z 까지
0-9    : 0 에서 9 까지
```

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

[ ]	괄호안의 조건을 만족하는 문자 하나
a-z	a 에서 z 까지
0-9	0 에서 9 까지
.	[]안에서는 그냥 “.”

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

```
[a-z0-9.]{6,30}
```

```
[    ]: 괄호안의 조건을 만족하는 문자 하나
a-z   : a 에서 z 까지
0-9   : 0 에서 9 까지
.     : []안에서는 그냥 “.”
{m,n} : 선행 패턴 m회 이상 n회 이하
```

= 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어

Between 3 ~ 20 characters

Use letters, numbers, and periods

No special characters

No uppercase

뭐라고 하는지 모르겠다.
그냥 존나 가만있어야겠다.

http://regex101.com/

Literal Matching

Special Characters

Anchors

Quantifiers

Class

Group

Literal Matching   match this

Special Characters

Anchors

Quantifiers

Class

Group

Literal Matching   match this

Special Characters   -\^$.|?*+()[]

Anchors

Quantifiers

Class

Group

<table>
<tr><td>Literal Matching</td><td>match this</td></tr>
<tr><td>Special Characters</td><td>-\^$.|?*+()[]</td></tr>
<tr><td>Anchors</td><td>^I want, I want$</td></tr>
<tr><td>Quantifiers</td><td></td></tr>
<tr><td>Class</td><td></td></tr>
<tr><td>Group</td><td></td></tr>
</table>

<table>
<tr><td>Literal Matching</td><td>match this</td></tr>
<tr><td>Special Characters</td><td>-\^$.|?*+()[]</td></tr>
<tr><td>Anchors</td><td>^I want, I want$</td></tr>
<tr><td>Quantifiers</td><td>? + {n} {m,n} *</td></tr>
<tr><td>Class</td><td></td></tr>
<tr><td>Group</td><td></td></tr>
</table>

<table>
<tr><td>Literal Matching</td><td>match this</td></tr>
<tr><td>Special Characters</td><td>-\^$.|?*+()[]</td></tr>
<tr><td>Anchors</td><td>^I want, I want$</td></tr>
<tr><td>Quantifiers</td><td>? + {n} {m,n} *</td></tr>
<tr><td>Class</td><td>[   ]</td></tr>
<tr><td>Group</td><td></td></tr>
</table>

Literal Matching	match this
Special Characters	-\^$.|?*+()[]
Anchors	^I want, I want$
Quantifiers	? + {n} {m,n} *
Class	[ ]
Group	( ) |

```
[A-Za-z0-9_]
```

```
[A-Za-z0-9_]        \w
```

```
[A-Za-z0-9_]        \w

[^A-Za-z0-9_]
```

```
[A-Za-z0-9_]        \w

[^A-Za-z0-9_]       \W
```

```
[A-Za-z0-9_]        \w

[^A-Za-z0-9_]       \W

[0-9]
```

[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d

[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d
[^0-9]	

[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d
[^0-9]	\D

[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d
[^0-9]	\D
[\n\r\t\f]	

[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d
[^0-9]	\D
[\n\r\t\f]	\s

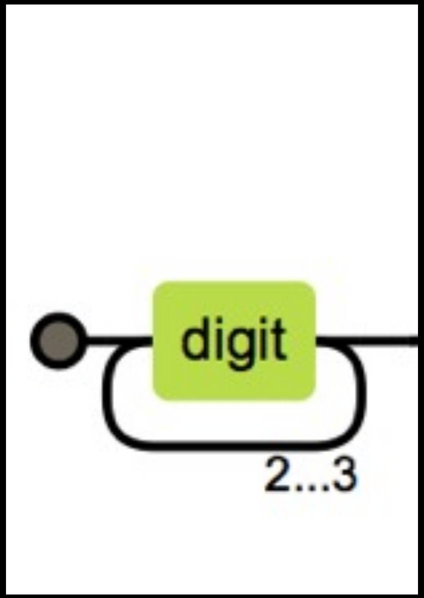
[A-Za-z0-9_]	\w
[^A-Za-z0-9_]	\W
[0-9]	\d
[^0-9]	\D
[\n\r\t\f]	\s
[^\n\r\t\f]	

`[A-Za-z0-9_]`	`\w`
`[^A-Za-z0-9_]`	`\W`
`[0-9]`	`\d`
`[^0-9]`	`\D`
`[\n\r\t\f]`	`\s`
`[^\n\r\t\f]`	`\S`

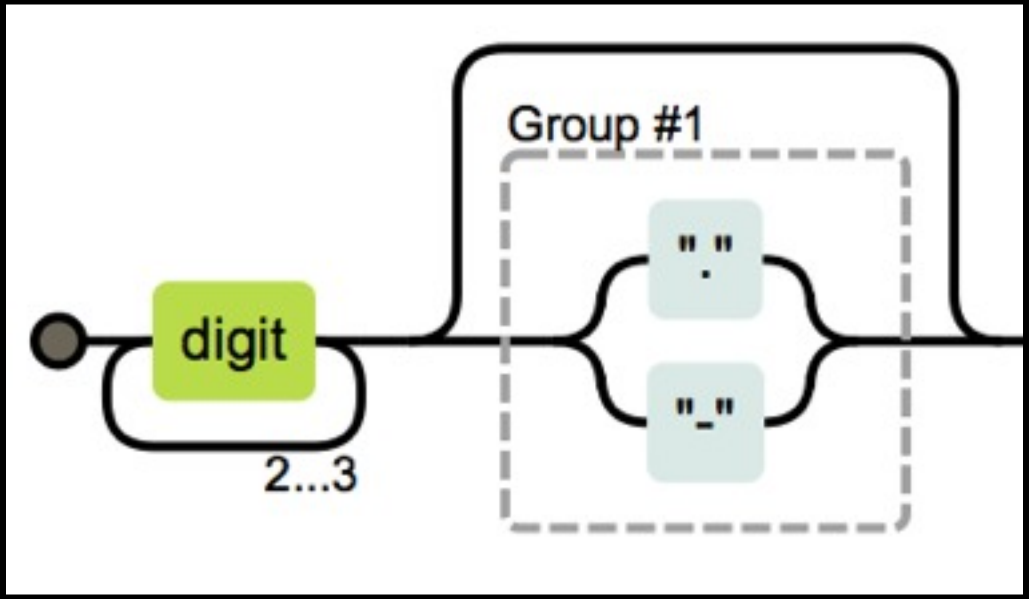
```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```

```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```

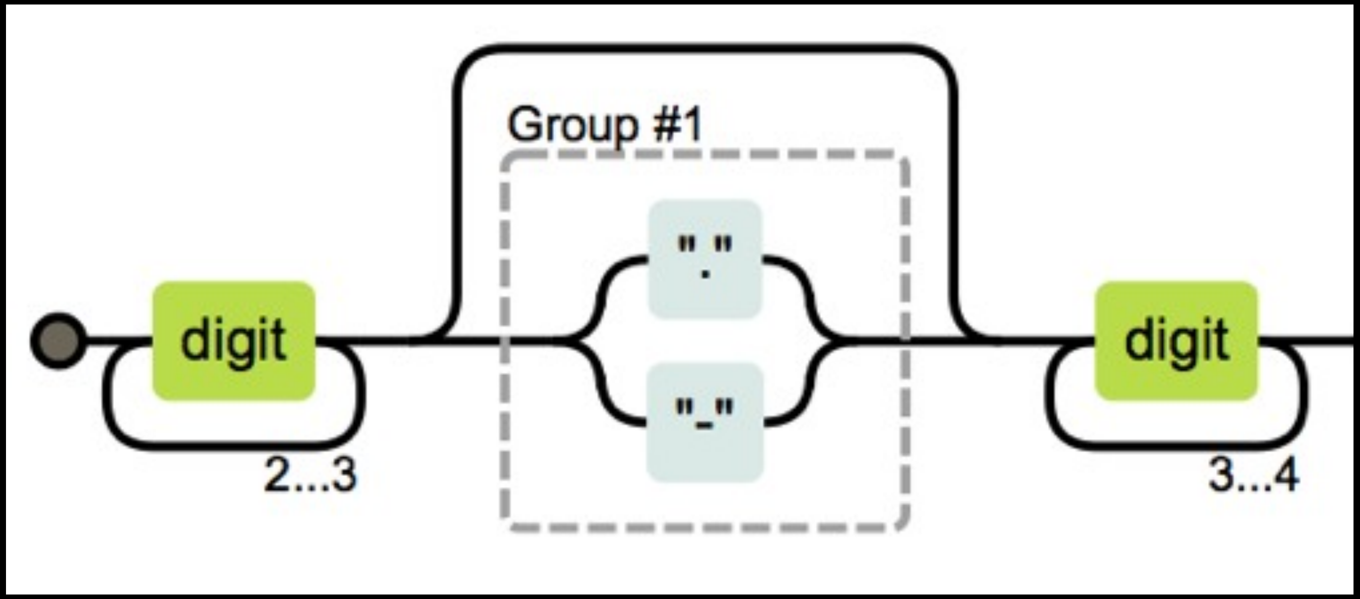
```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```



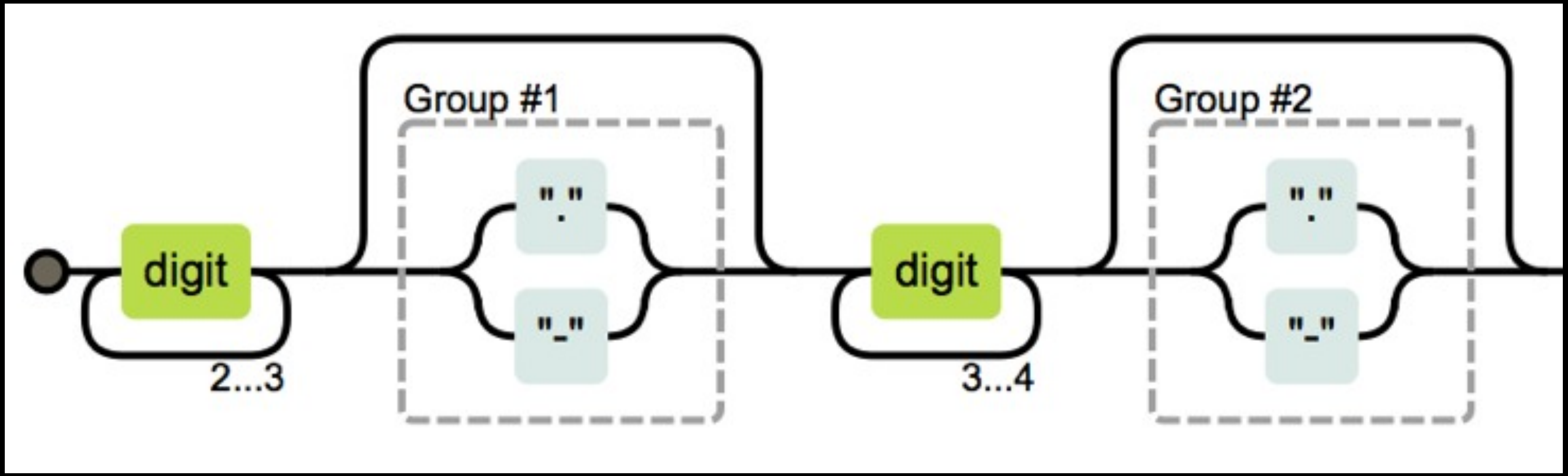
```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```



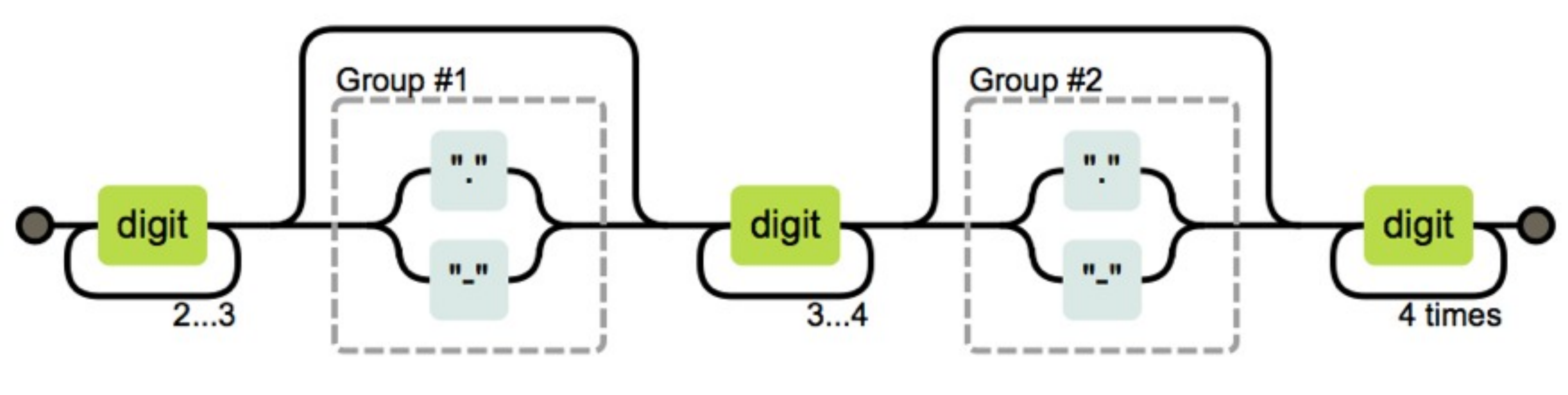
```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```



```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```

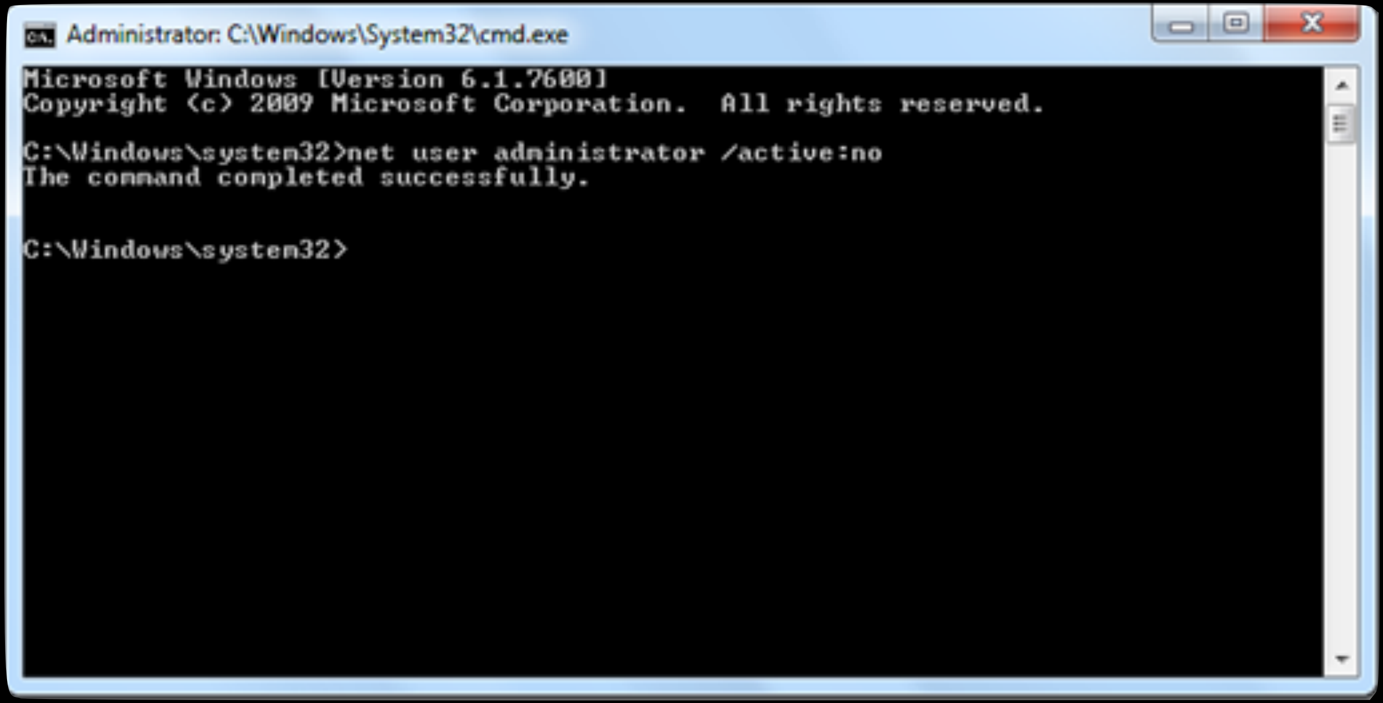


```
\d{2,3}(\.|\-)?\d{3,4}(\.|\-)?\d{4}
```

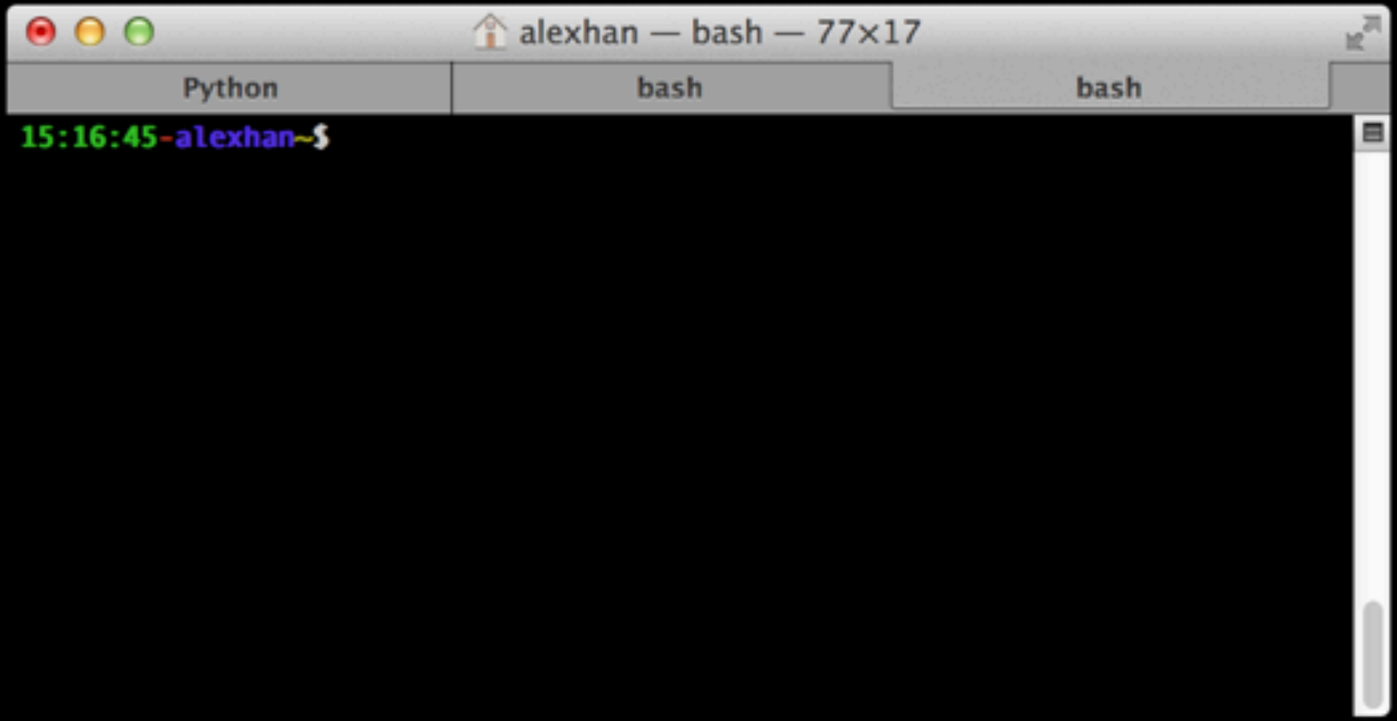


그래서
어쩌라고.

cmd & Terminal



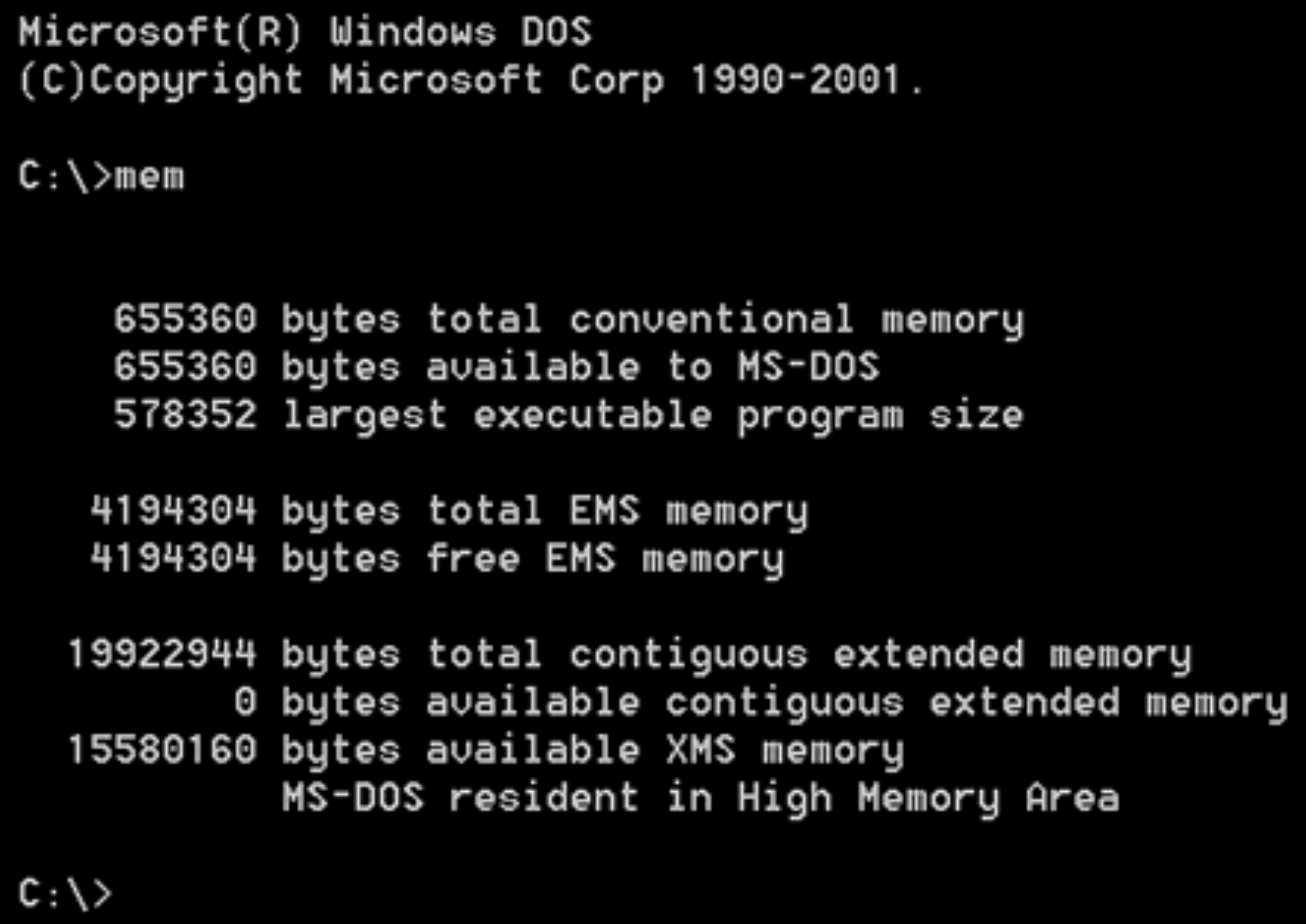
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>net user administrator /active:no
The command completed successfully.


C:\Windows\system32>
```



```
15:16:45-alexhan~$
```

MS-DOS

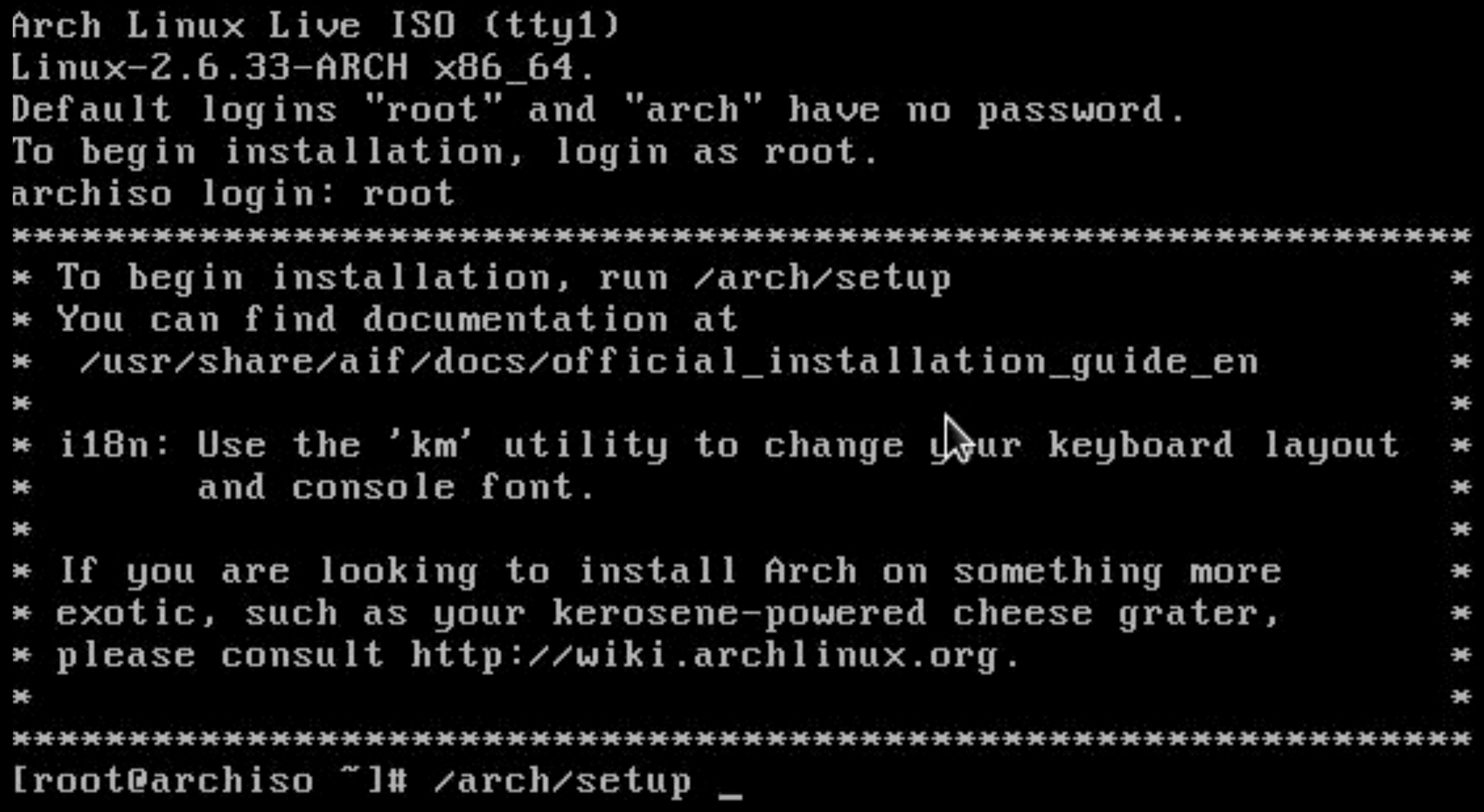
MS-DOS



```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

C:\>mem


    655360 bytes total conventional memory
    655360 bytes available to MS-DOS
    578352 largest executable program size

   4194304 bytes total EMS memory
   4194304 bytes free EMS memory

  19922944 bytes total contiguous extended memory
         0 bytes available contiguous extended memory
  15580160 bytes available XMS memory
           MS-DOS resident in High Memory Area

C:\>
```

Linux

Linux



```
Arch Linux Live ISO (tty1)
Linux-2.6.33-ARCH x86_64.
Default logins "root" and "arch" have no password.
To begin installation, login as root.
archiso login: root
***************************************************************
* To begin installation, run /arch/setup                      *
* You can find documentation at                               *
*  /usr/share/aif/docs/official_installation_guide_en         *
*                                                             *
* i18n: Use the 'km' utility to change your keyboard layout   *
*       and console font.                                     *
*                                                             *
* If you are looking to install Arch on something more        *
* exotic, such as your kerosene-powered cheese grater,        *
* please consult http://wiki.archlinux.org.                   *
*                                                             *
***************************************************************
[root@archiso ~]# /arch/setup _
```

명령어

명령어

```
pip
```

```
명령어 [옵션]

 pip
```

```
명령어 [옵션]

 pip install
```

```
명령어 [옵션] ([옵션] [옵션]..)

 pip install
```

```
명령어 [옵션] ([옵션] [옵션]..)

pip install BeautifulSoup4
```

```
cd [dir]
cd [dir]

cd ..
cd [dir]

cd ..

python [file.py]
```

```
cd [dir]

cd ..

python [file.py]

Mac : ls  | Windows : dir
```

HTTP Error

HTTP Error

404

HTTP Error

404 : Page Not Found

404 ERROR
*** GIRLFRIEND NOT FOUND ***

HTTP Error

404 : Page Not Found

200

HTTP Error

404 : Page Not Found

200 : OK

HTTP Error

404 : Page Not Found

200 : OK

500

HTTP Error

404 : Page Not Found

200 : OK

500 : Internal Server Error

```python
from flask import Flask

app = Flask(__name__)
```

```
@app.route(‘/’)
def index():
    return render_template(‘index.html’)
```

```
@app.route(‘/’)
def index():
    return render_template(‘index.html’)
```

http://____.appspot.com/

```
@app.route(‘/’)
def index():
    return render_template(‘index.html’)
```

http://____.appspot.com/

```
@app.route(‘/idea’)
def idea():
    return render_template(‘idea.html’)
```

```
http://____.appspot.com/idea
```

```
if __name__ == ‘__main__’:
    app.run()
```

```
from flask import Flask
app = Flask(__name__)

@app.route(‘/’)
def index():
    return render_template(‘index.html’)

if __name__ == ‘__main__’:
    app.run()
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
{{ url_for(‘static’, filename=‘__’) }}
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
{{ url_for(‘static’, filename=‘__’) }}
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
[flaskr]
   |-[static]
   |-[templates]
   |      |-layout.html
   |      |-login.html
   |      |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
   render_template(‘___.html’)
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
render_template(‘___.html’)
```

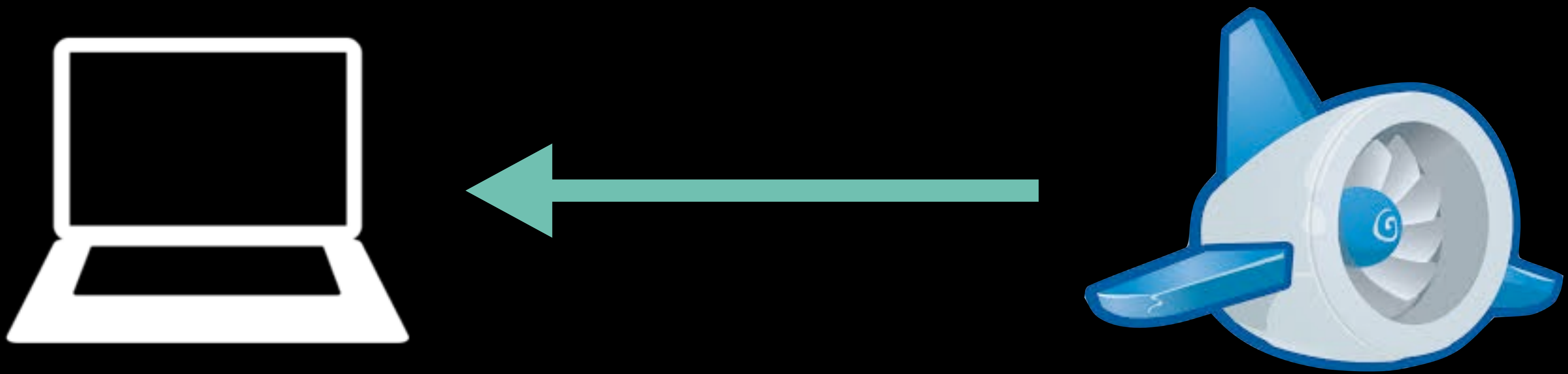
```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

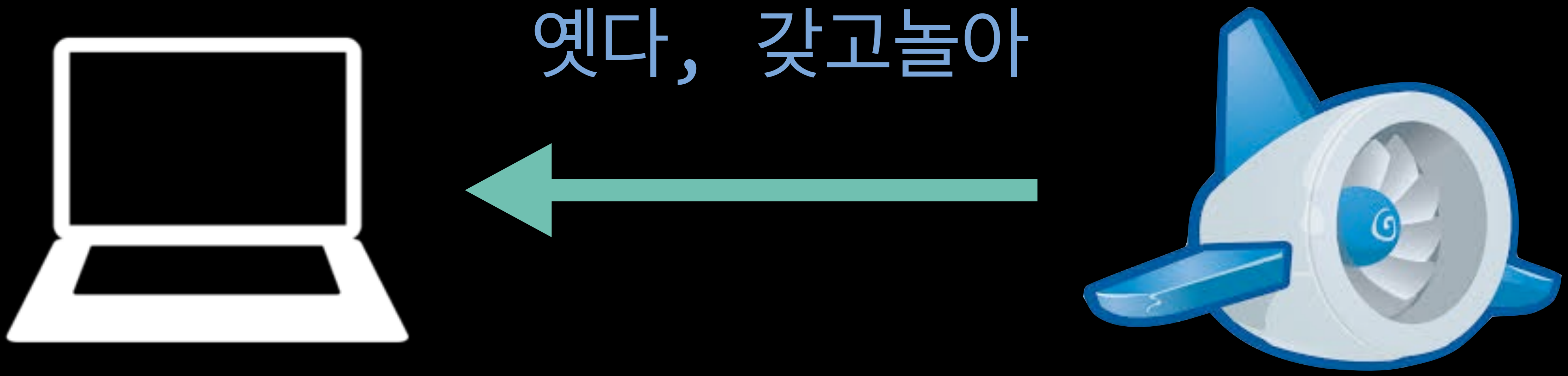
```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

```
[flaskr]
   |-[static]
   |-[templates]
   |     |-layout.html
   |     |-login.html
   |     |-show_entries.html
   |-flaskr.py
   |-schema.sql
```

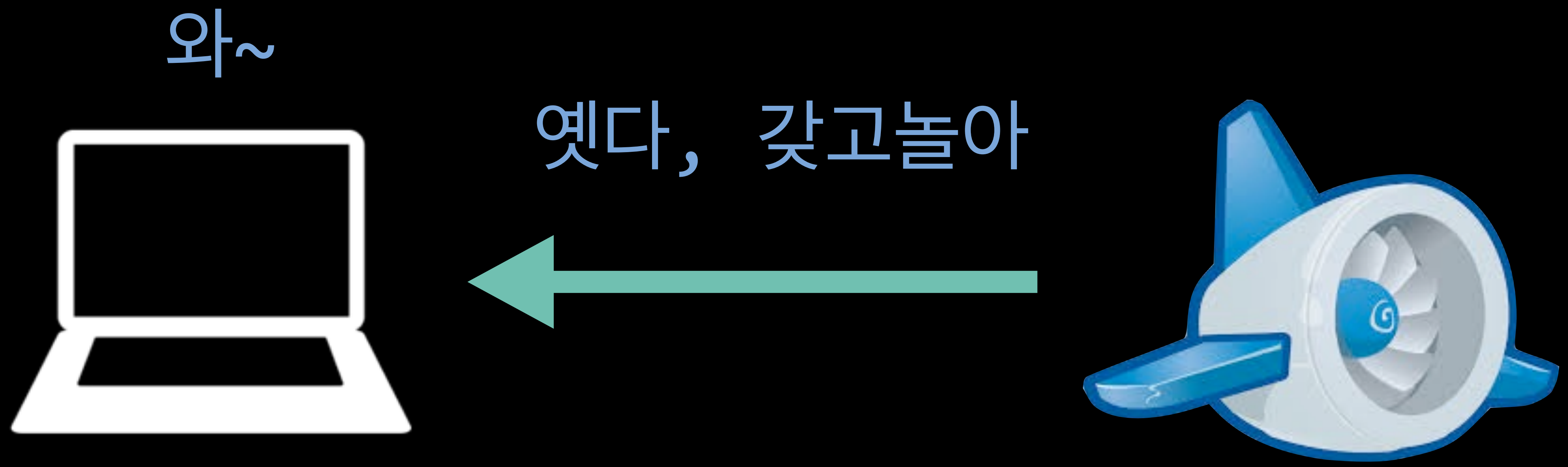
Login

뭐가 다를까?



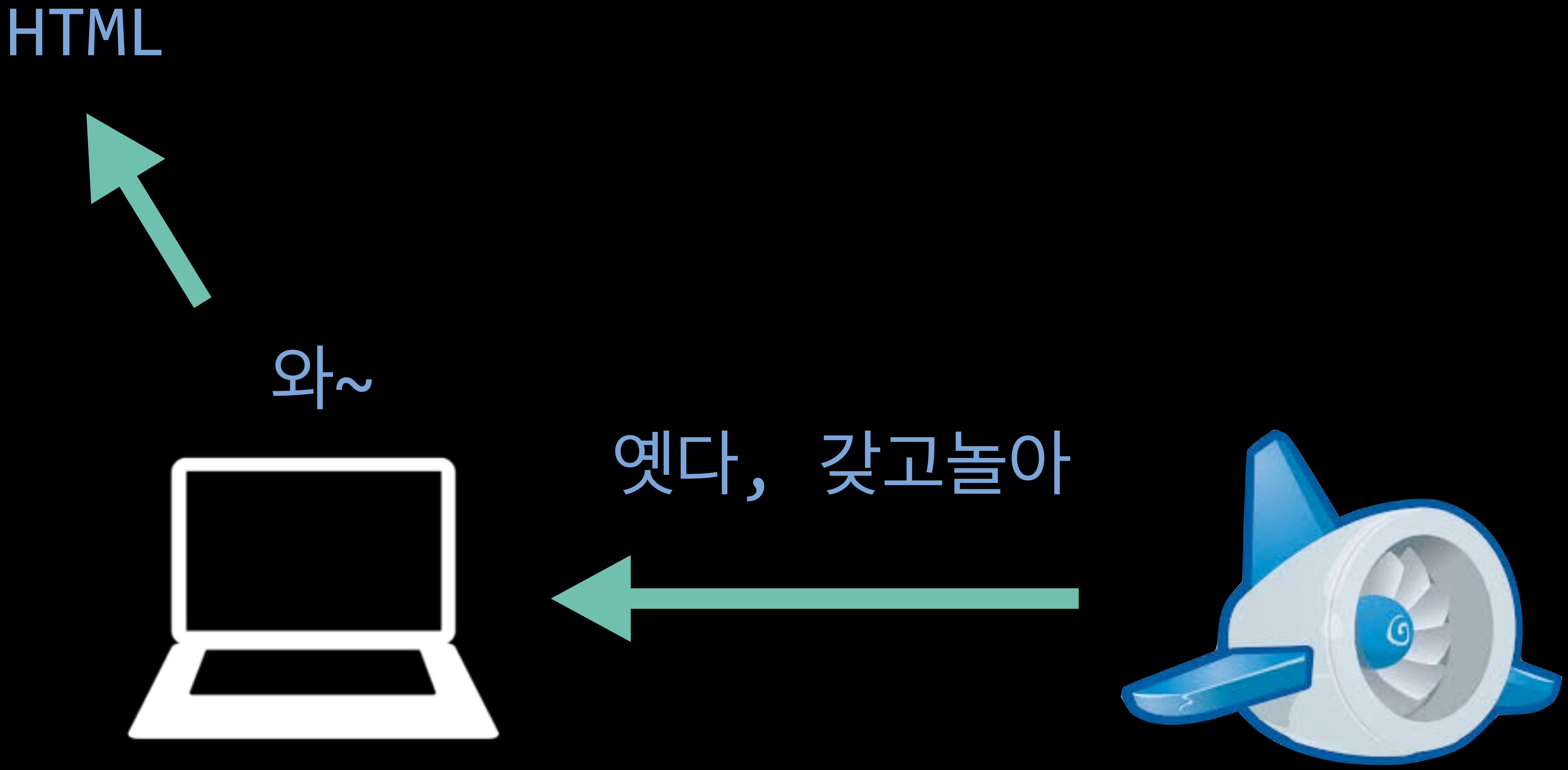


옛다, 갖고놀아



와~

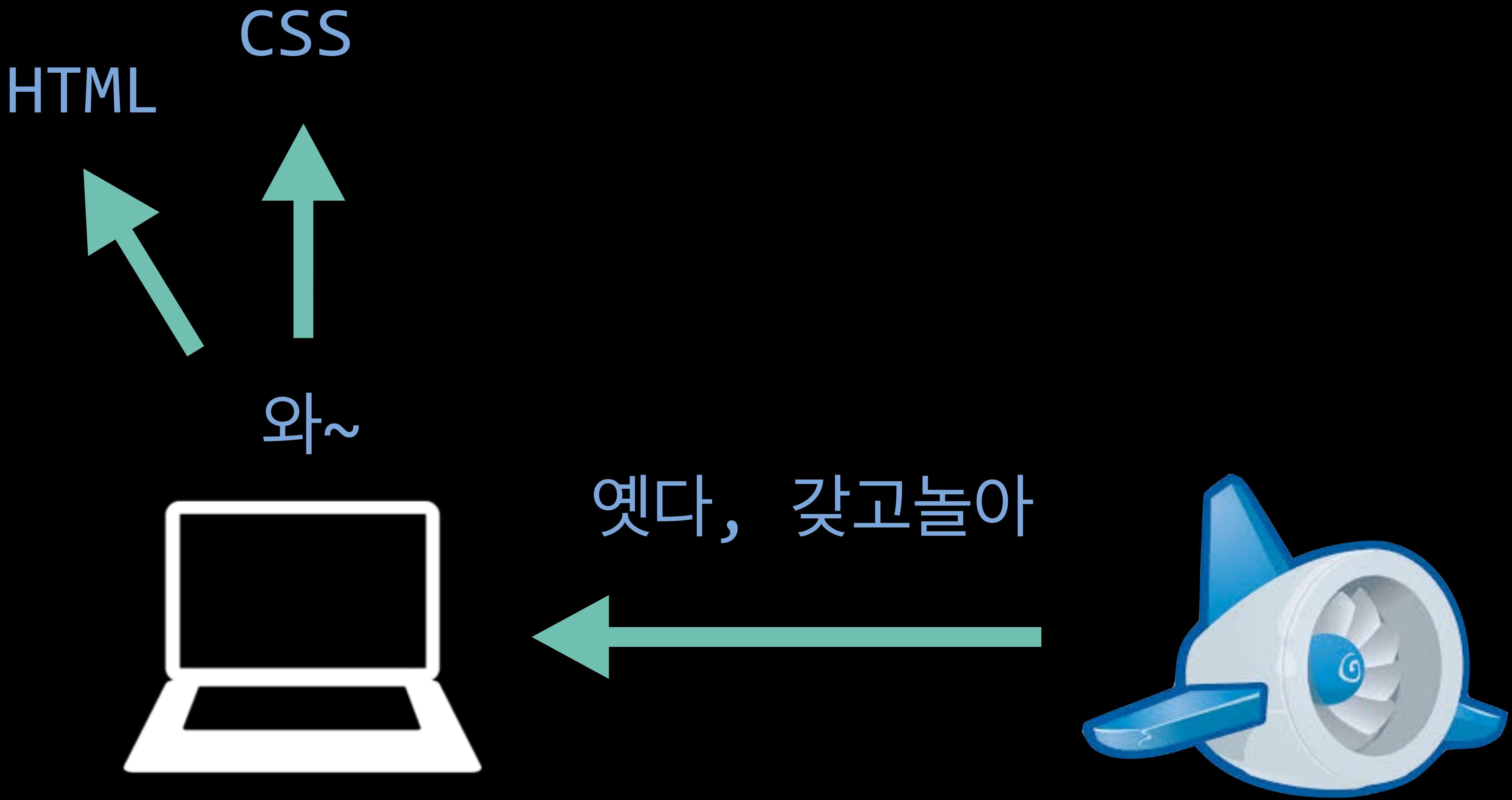
옜다, 갖고놀아



HTML

와~

옛다, 갖고놀아

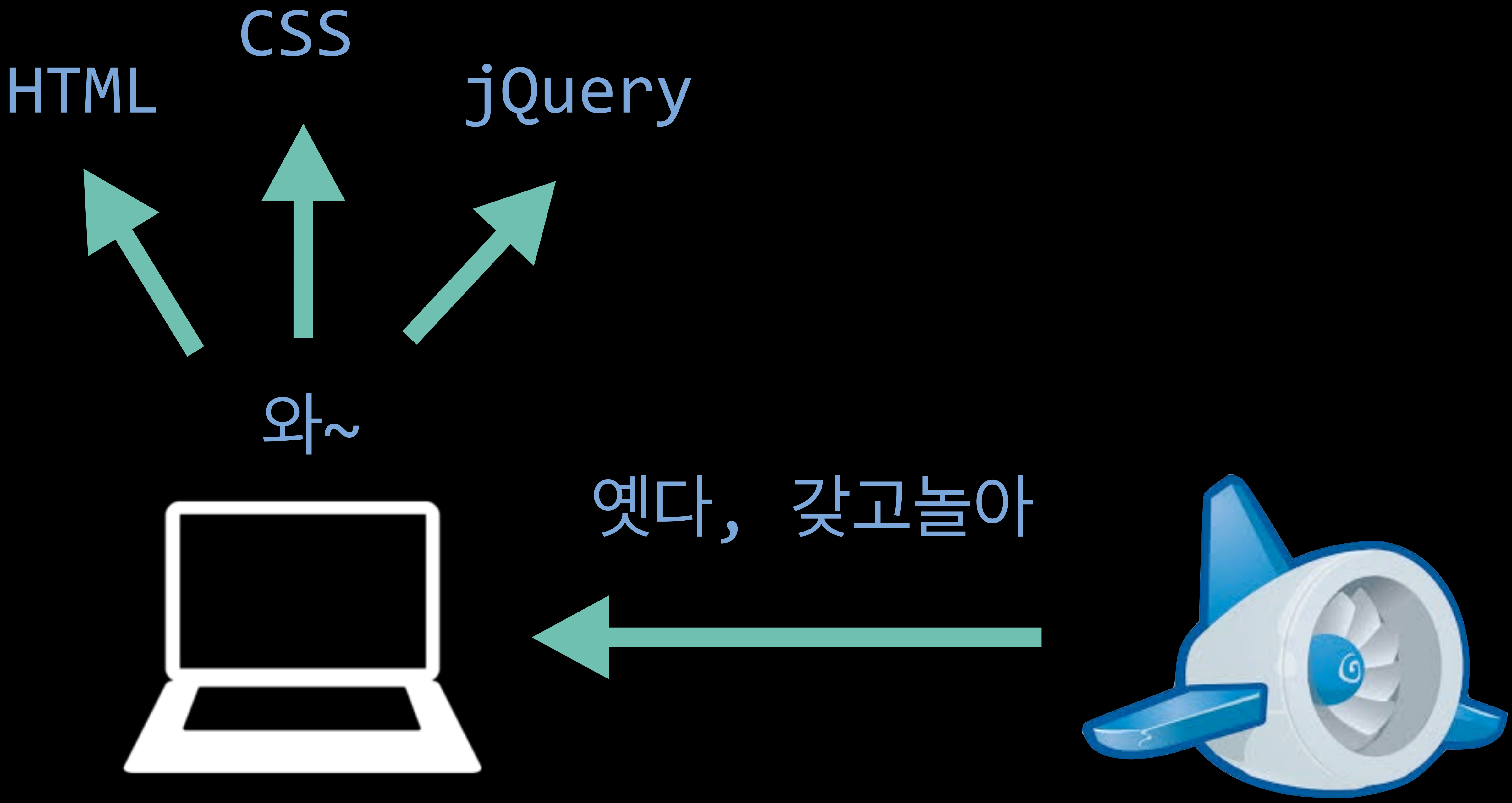


CSS

HTML

와~

옛다, 갖고놀아



HTML

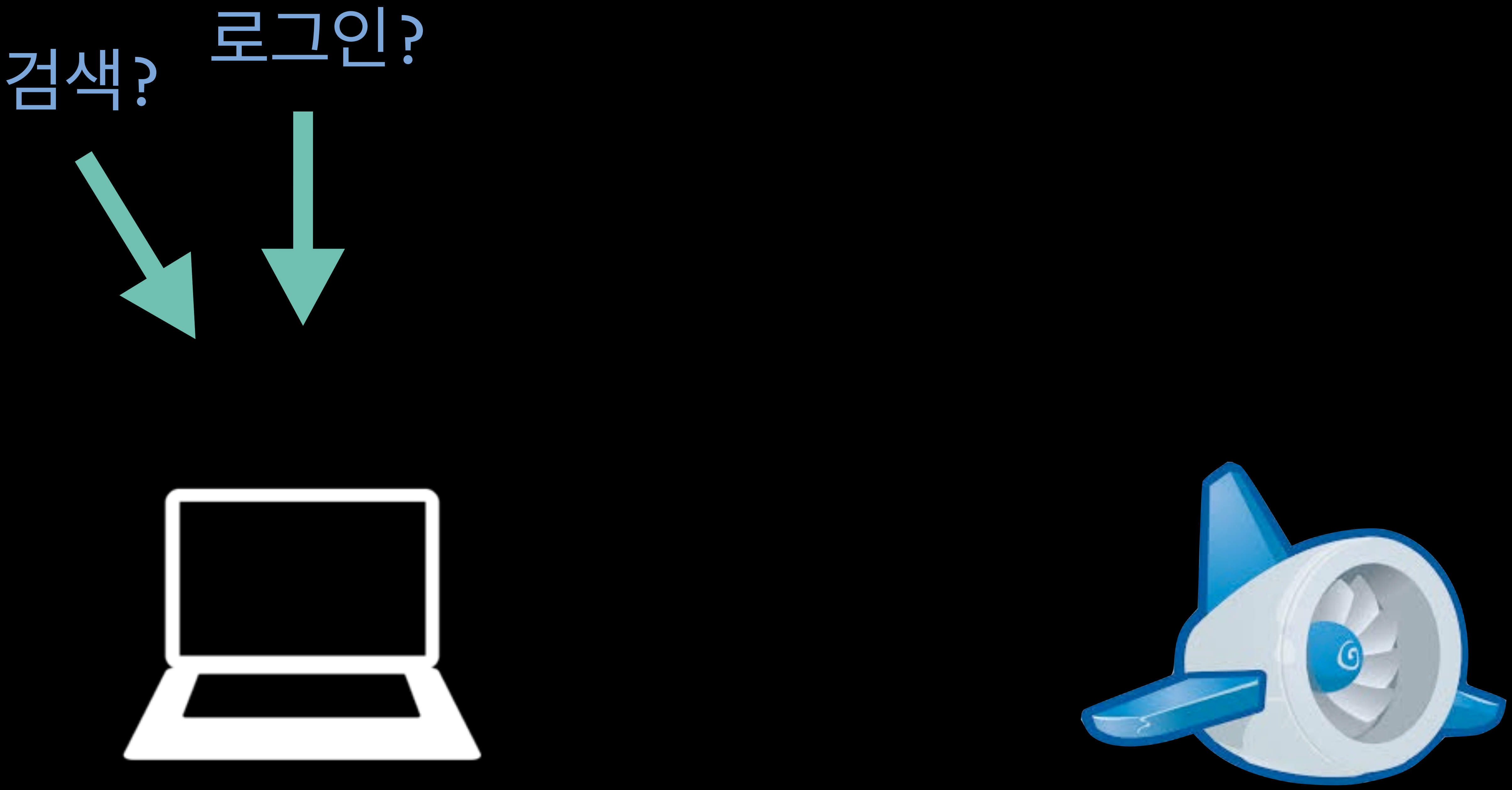
CSS

jQuery

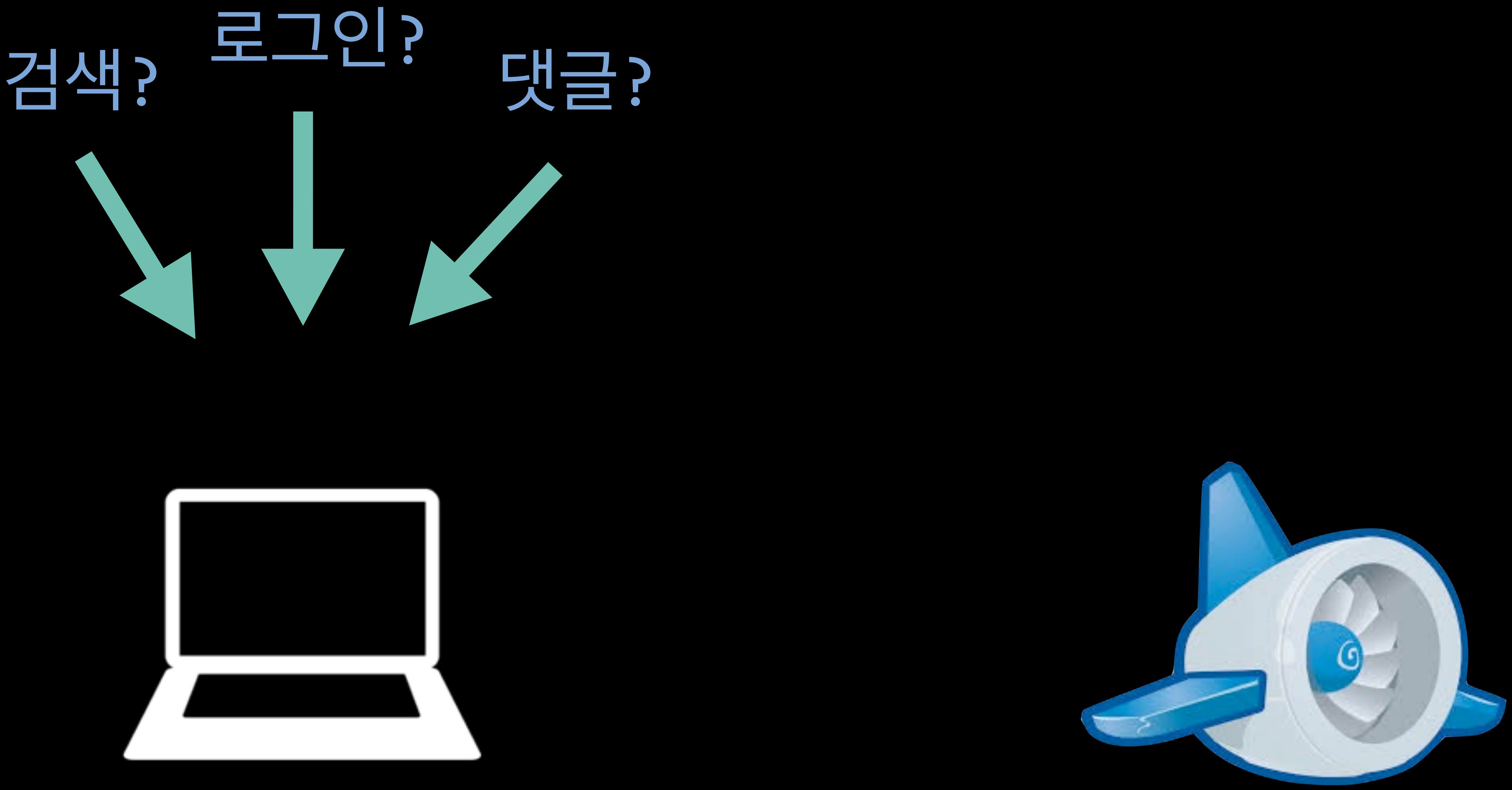
와~

옛다, 갖고놀아

검색?



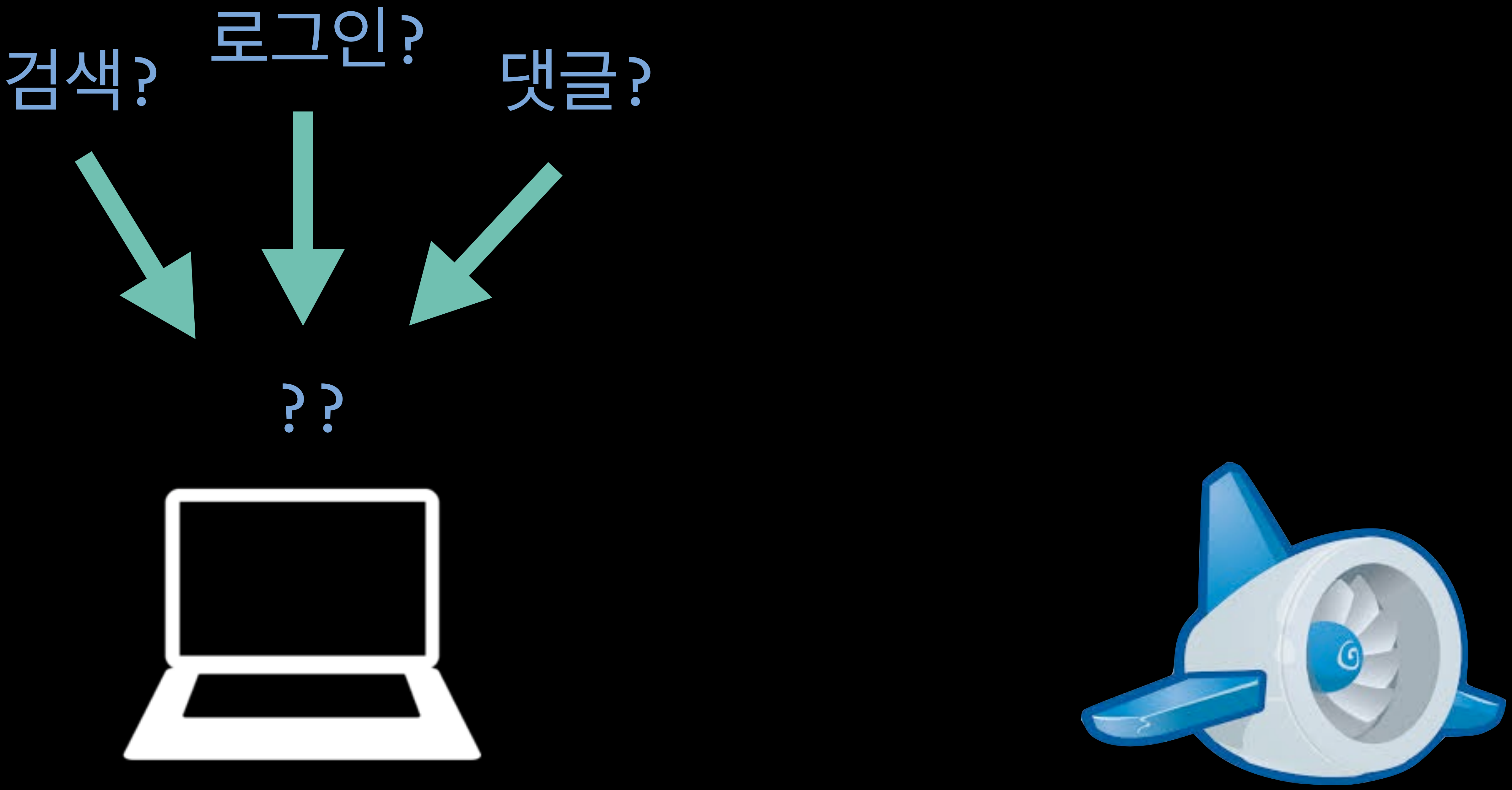
검색?
로그인?



검색?

로그인?

댓글?

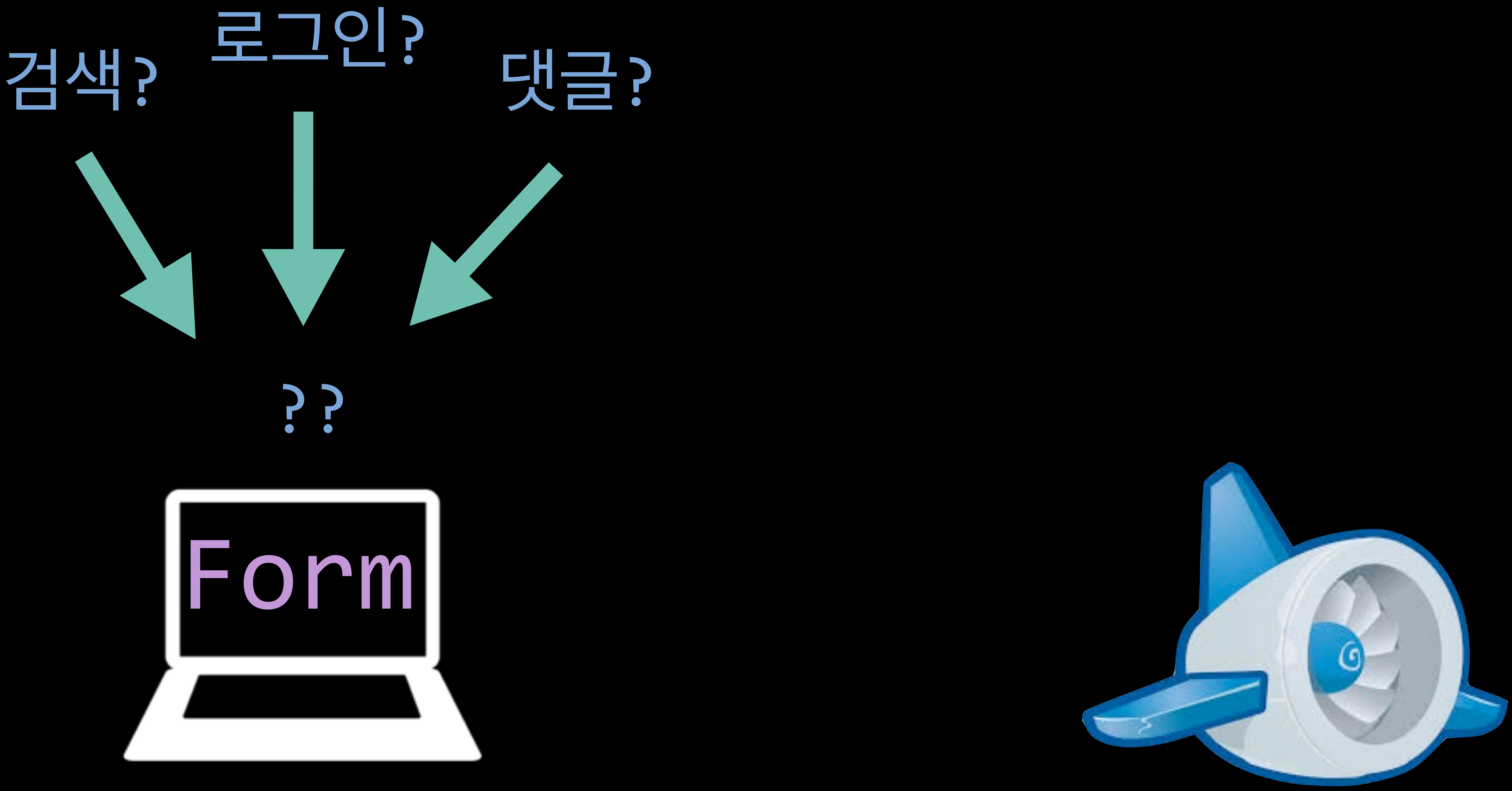


검색?

로그인?

댓글?

??



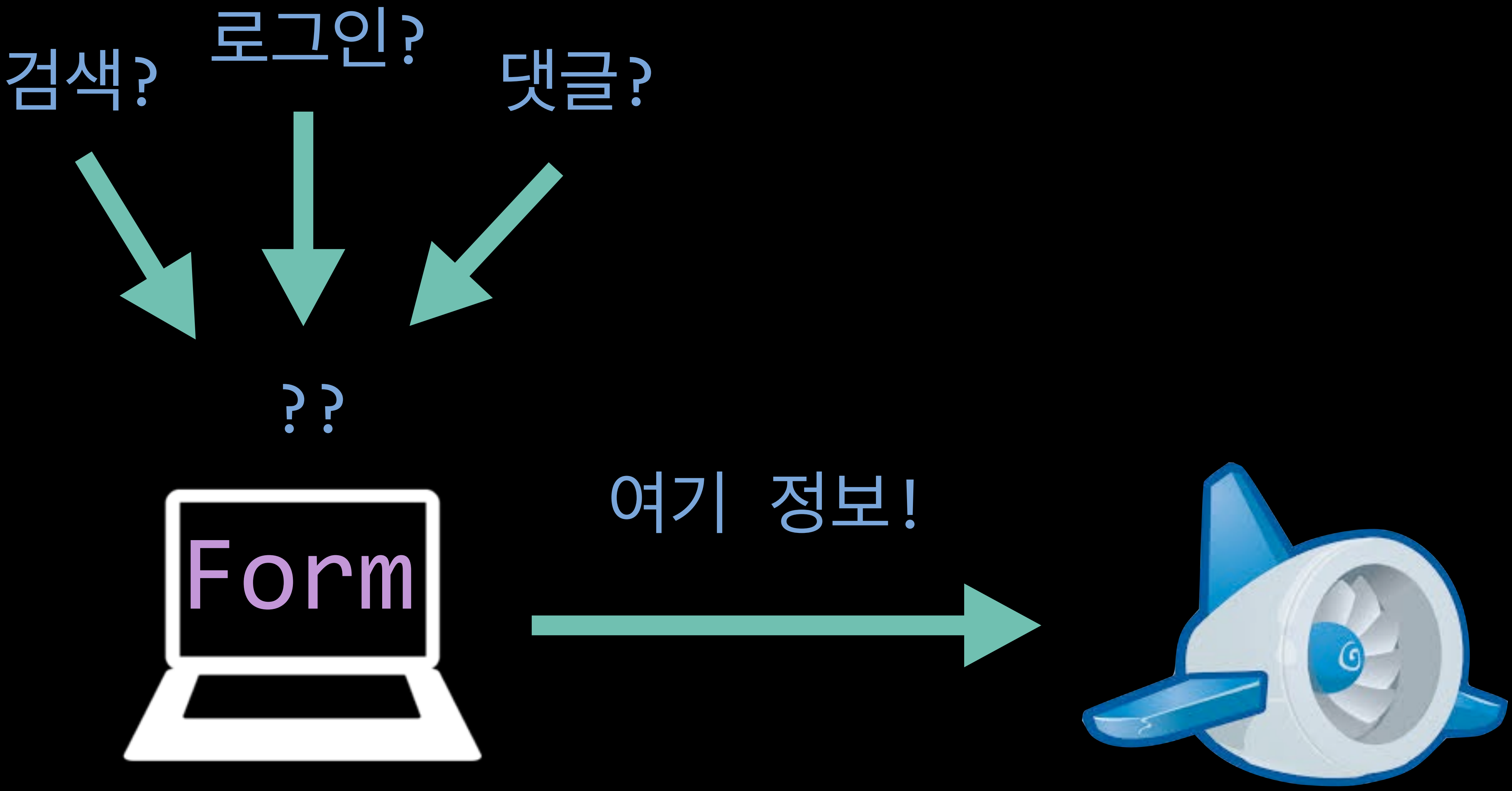
검색?

로그인?

댓글?

??

Form



검색?

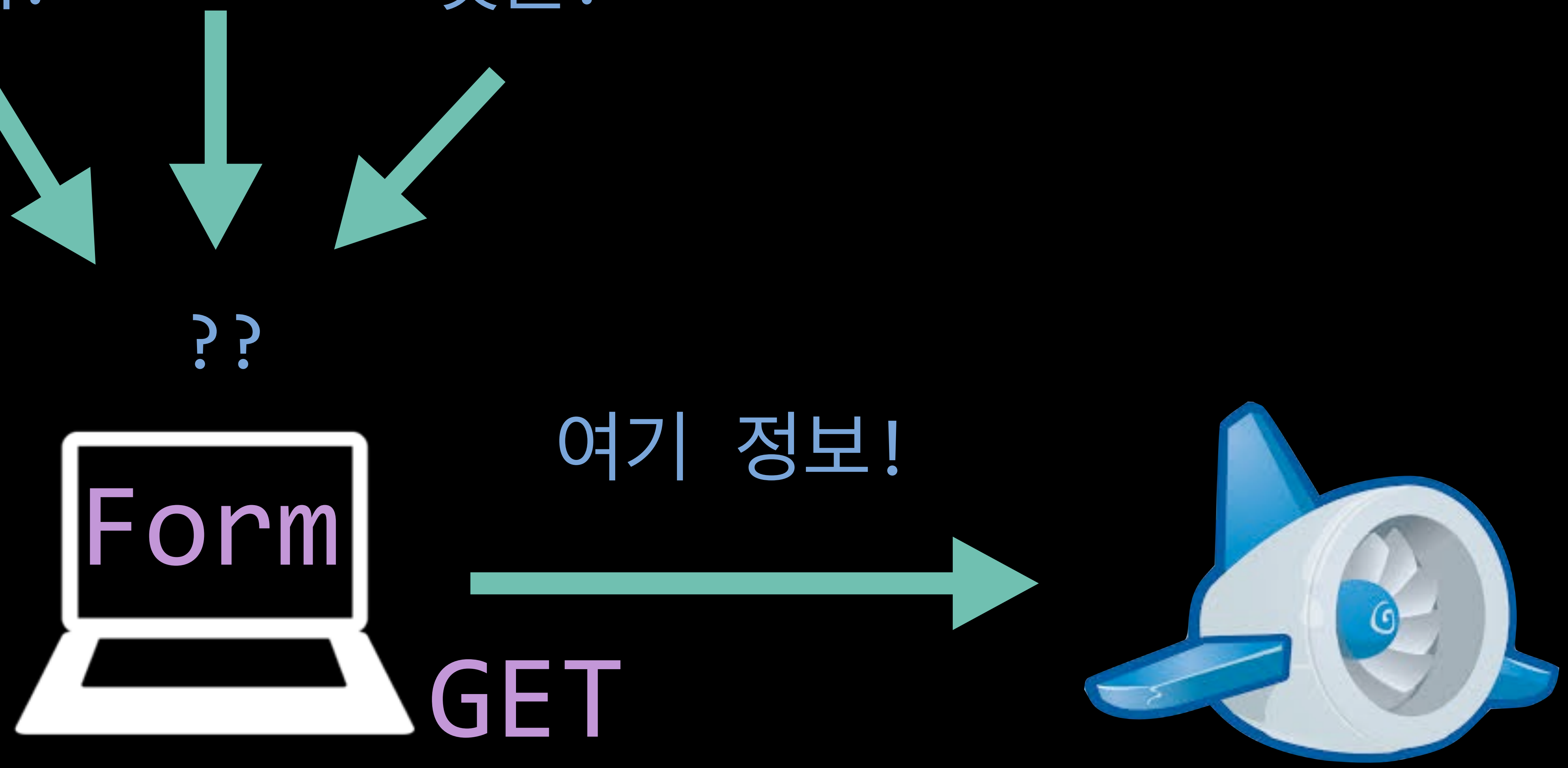
로그인?

댓글?

??

Form

여기 정보!



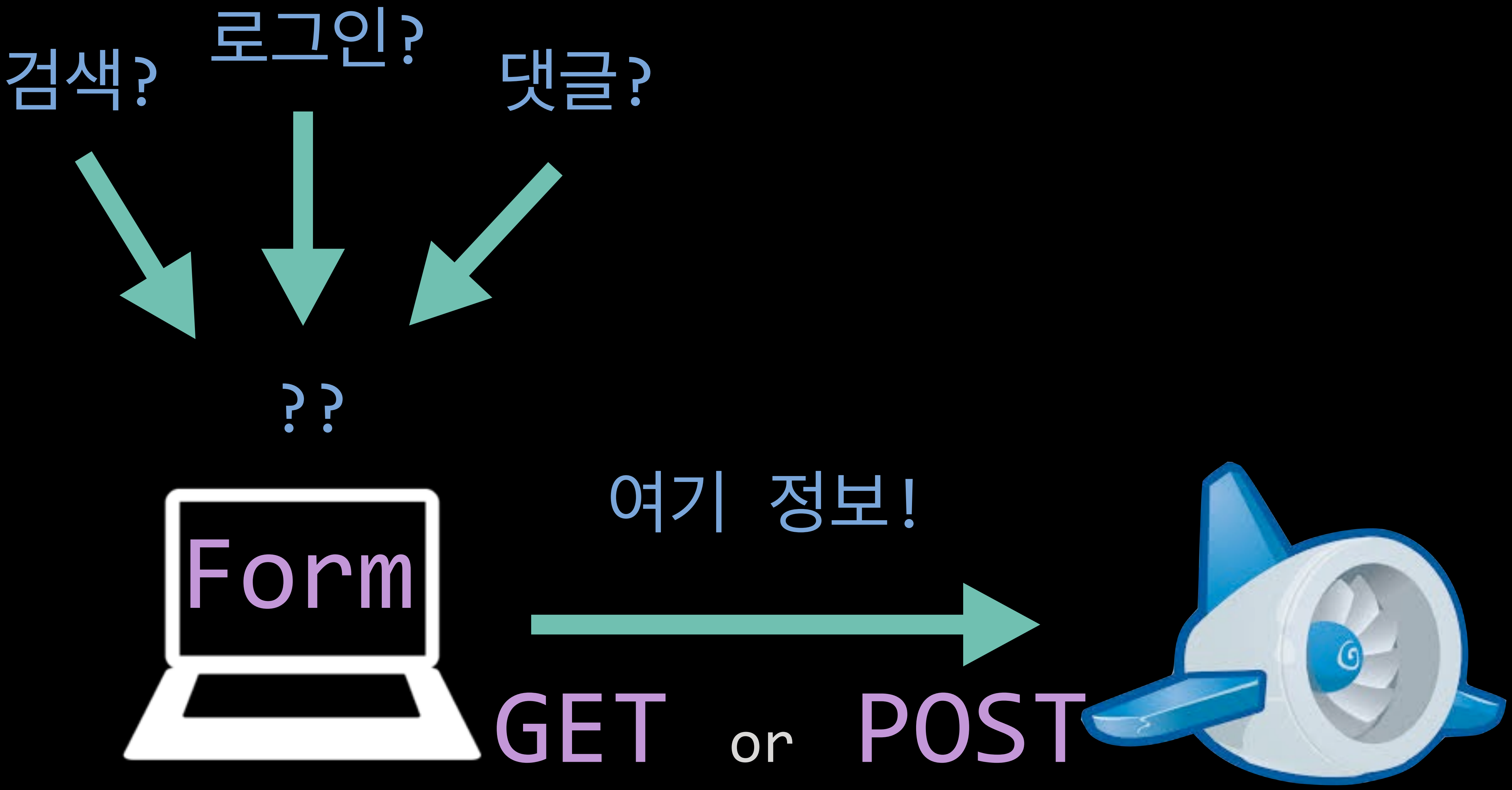
검색? 로그인? 댓글?

??

Form

여기 정보!

GET



검색? 로그인? 댓글?

??

Form

여기 정보!

GET or POST

```
<form action="{{ url_for('login') }}" method=post>
  <dl>
    <dt>Username:</dt>
      <dd><input type=text name=username></dd>
    <dt>Password:</dt>
      <dd><input type=password name=password></dd>
      <dd><input type=submit value=Login></dd>
  </dl>
</form>
```

login.html

---

flaskr.py

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != app.config['USERNAME']:
            error = 'Invalid username'
        elif request.form['password'] != app.config['PASSWORD']:
            error = 'Invalid password'
        else:
            session['logged_in'] = True
            flash('You were logged in')
            return redirect(url_for('show_entries'))
    return render_template('login.html', error=error)
```

```
<form action="{{ url_for('login') }}" method=post>
  <dl>
    <dt>Username:</dt>
      <dd><input type=text name=username></dd>
    <dt>Password:</dt>
      <dd><input type=password name=password></dd>
      <dd><input type=submit value=Login></dd>
  </dl>
</form>
```

login.html

flaskr.py

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != app.config['USERNAME']:
            error = 'Invalid username'
        elif request.form['password'] != app.config['PASSWORD']:
            error = 'Invalid password'
        else:
            session['logged_in'] = True
            flash('You were logged in')
            return redirect(url_for('show_entries'))
    return render_template('login.html', error=error)
```

```html
<form action="{{ url_for('login') }}" method=post>
  <dl>
    <dt>Username:</dt>
      <dd><input type=text name=username></dd>
    <dt>Password:</dt>
      <dd><input type=password name=password></dd>
      <dd><input type=submit value=Login></dd>
  </dl>
</form>
```

login.html

flaskr.py

```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != app.config['USERNAME']:
            error = 'Invalid username'
        elif request.form['password'] != app.config['PASSWORD']:
            error = 'Invalid password'
        else:
            session['logged_in'] = True
            flash('You were logged in')
            return redirect(url_for('show_entries'))
    return render_template('login.html', error=error)
```

```
<form action="{{ url_for('login') }}" method=post>
  <dl>
    <dt>Username:</dt>
      <dd><input type=text name=username></dd>
    <dt>Password:</dt>
      <dd><input type=password name=password></dd>
      <dd><input type=submit value=Login></dd>
  </dl>
</form>
```

login.html

---

flaskr.py

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != app.config['USERNAME']:
            error = 'Invalid username'
        elif request.form['password'] != app.config['PASSWORD']:
            error = 'Invalid password'
        else:
            session['logged_in'] = True
            flash('You were logged in')
            return redirect(url_for('show_entries'))
    return render_template('login.html', error=error)
```

GET vs POST

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached		
browser history		
bookmark		
length restrictions		
sensitive data		

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached	O	X
browser history		
bookmark		
length restrictions		
sensitive data		

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached	O	X
browser history	O	X
bookmark		
length restrictions		
sensitive data		

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached	O	X
browser history	O	X
bookmark	O	X
length restrictions		
sensitive data		

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached	O	X
browser history	O	X
bookmark	O	X
length restrictions	O	X
sensitive data		

GET vs POST

- GET: /test/demo_form.asp?name1=value1&name2=value2

	GET	POST
cached	O	X
browser history	O	X
bookmark	O	X
length restrictions	O	X
sensitive data	X	O