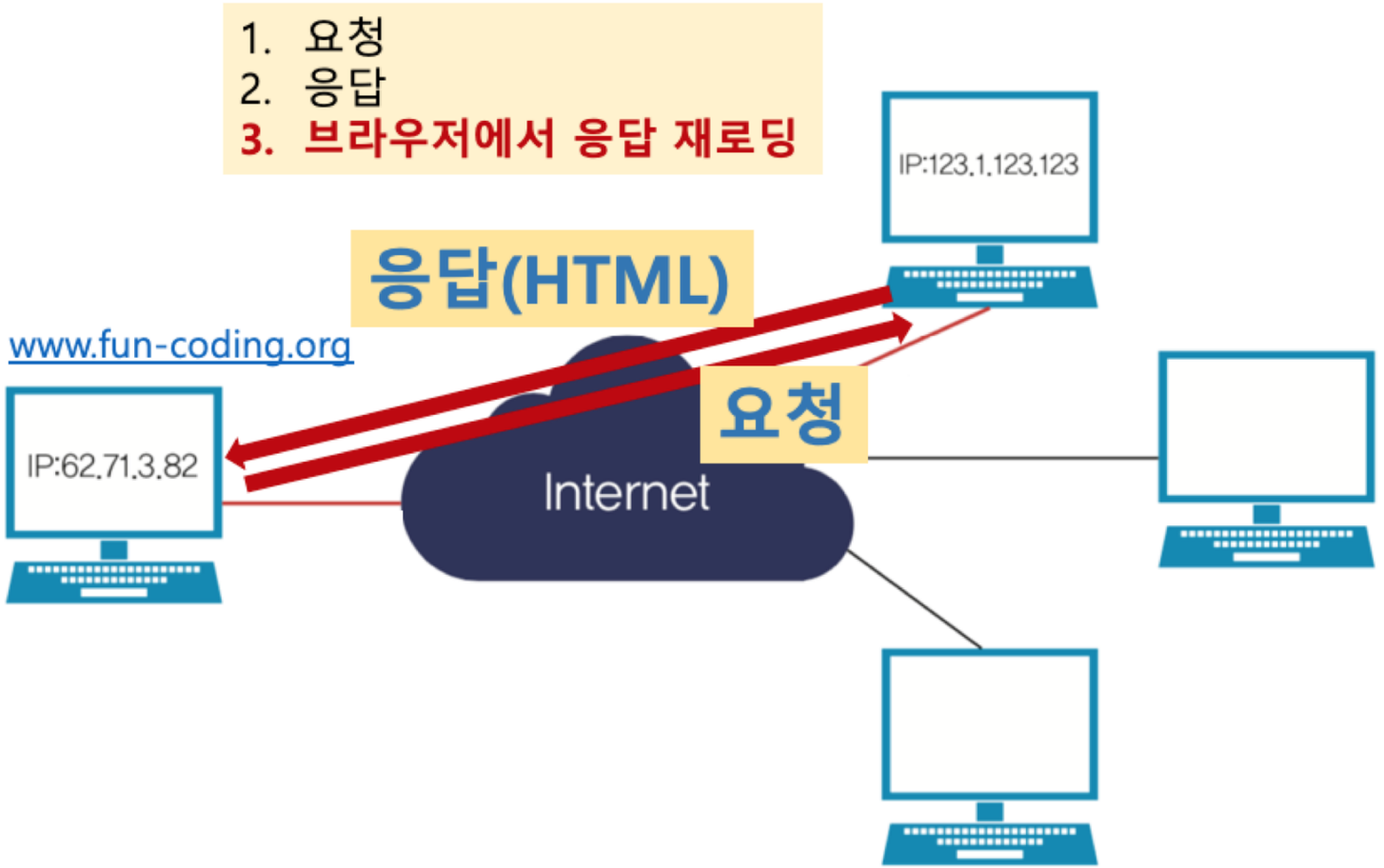
Selenium 과 PhantomJS 활용

> 특정 웹페이지는 HTML로는 표시가 안되는 내용이 웹페이지에 표기됨

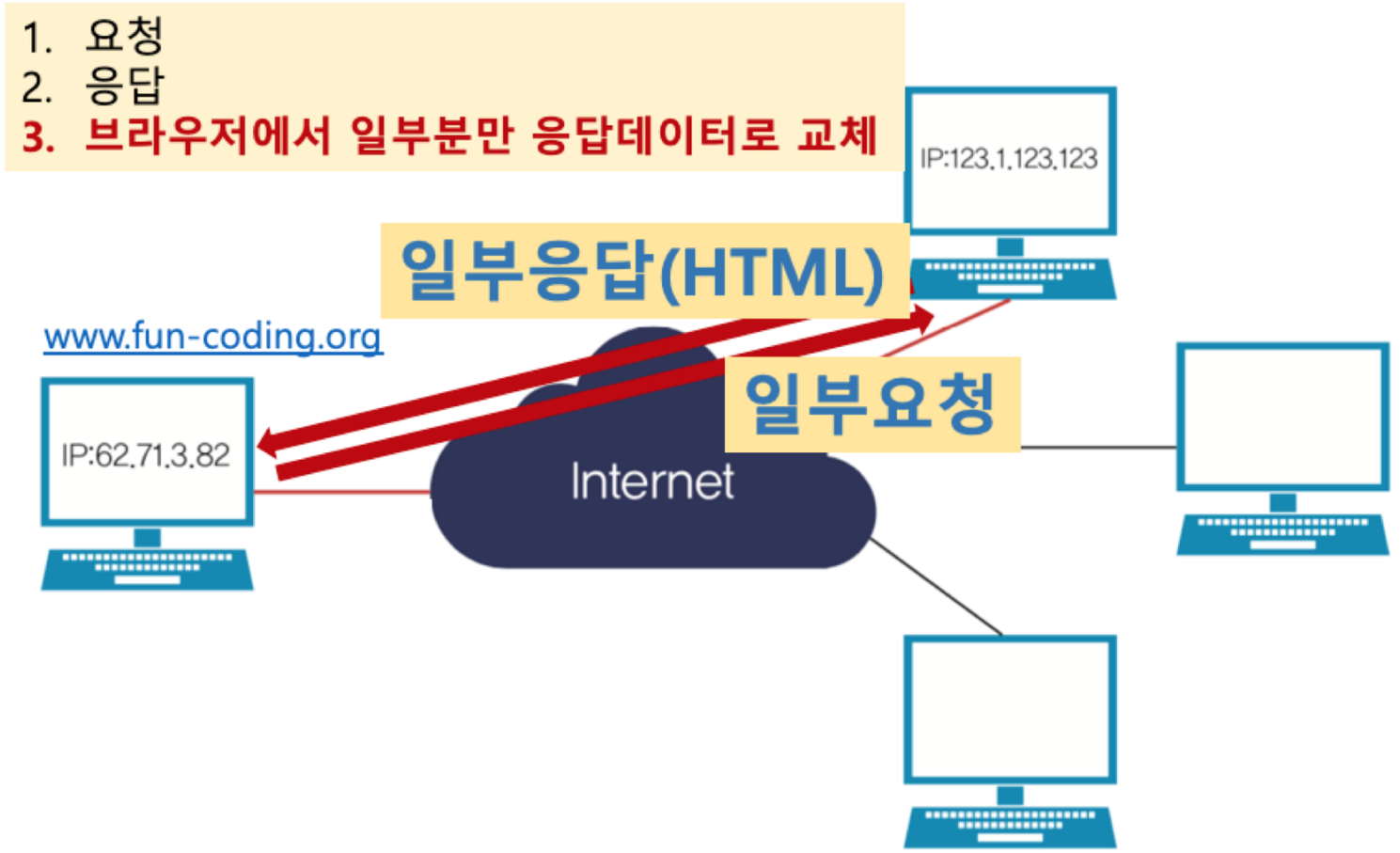
Ajax과 같은 동적 웹페이지 데이터 로딩 기술

> 웹페이지 새로고침 을 눌러보세요!



Ajax과 같은 동적 웹페이지 데이터 로딩 기술

> 웹페이지를 새로고침 하지 않고, 일부분만 바꾸면, 시간단축!



특정 태그 일정 시간 기다리기 기능

```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

element = WebDriverWait(driver, 3).until(
        EC.presence_of_element_located((By.ID, "alex-area"))
    )
```

특정 태그 존재 여부 확인 기능

```
from selenium.webdriver.common.by import By
```

- 해당 태그가 존재하는지 확인
  - 예: (By.ID, "alex-area")
- 태그 선택 방법
  - By.CLASS_NAME: class name
  - By.CSS_SELECTOR: css selector
  - By.ID: id
  - By.NAME: name
  - By.TAG_NAME: tag name

특정 태그 일정 시간 기다리기 기능

```
from selenium.common.exceptions import TimeoutException
```

```
        try:
            element = WebDriverWait(driver, 3).until(
                EC.presence_of_element_located((By.CSS_SELECTOR, "a"))
            )
            more_button = driver.find_element_by_css_selector("a")
            more_button.click()
            count += 1
        except TimeoutException:
            loop = False
```

키보드/마우스 동작 자동화하기

```
from selenium import webdriver

hidden_submenu = driver.find_element_by_css_selector(".nav #submenu1")

actions = webdriver.ActionChains(driver)
actions.click(hidden_submenu)
actions.perform()

또는
webdriver.ActionChains(driver).click(hidden_submenu).perform()
```

> ActionChains() 참고 페이지

실전 예제1: 다음 뉴스 댓글 가져오기

- 다음 뉴스 댓글 태그 검색해보기
    - 직접 페이지 HTML과 댓글 검색해보기

> 문제는 기존 기법으로는 크롤링이 안됨!
> Selenium, PhantomJS 와 같이 브라우저를 직접 제어하지 않으면 크롤링할 수 없음

Selenium 과 PhantomJS 활용 다음 뉴스 댓글 가져오기

- WebDriverWait() 메서드
  - 명시적인 페이지 로드 대기에 사용됨
  - 주로 다음 코드와 같이 사용됨

```
try:
    element = WebDriverWait(driver, 몇초).until(
        # By.ID 는 ID로 검색, By.CSS_SELECTOR 는 CSS Selector 로 검색
        EC.presence_of_element_located((By.ID, "cMain"))
    )
except TimeoutException:
    print("타임아웃")
finally:
    driver.quit()
```

여기서 잠깐! try, except, finally 란?

- 예외 처리
  - 예외 경우를 처리할 수 있는 문법
  - try 에서 실행한 명령이 정상 실행되지 않을 경우, except 에서 예외 상황 처리
  - finally 에서는 정상 동작하든, 예외 동작이든 간에, 반드시 실행되어야 하는 코드를 넣음 (옵션)

Selenium 과 PhantomJS 활용 다음 뉴스 댓글 가져오기

- presence_of_element_located() 메서드
  - 특정 태그가 있을 때까지 기다리는 코드
- 보통 from selenium.webdriver.common.by import By 와 같이 임포트 한 후에, 다음과 같이 사용
  - 파이썬 튜플 형태로 인자 기입

```
EC.presence_of_element_located((By.ID, "alex-area"))
```

- 주요 HTML 코드 지정 방법
  - By.ID - 태그에 있는 ID 로 검색
  - By.CSS_SELECTOR - CSS Selector 로 검색
  - By.NAME - 태그에 있는 name 으로 검색
  - By.TAG_NAME - 태그 이름으로 검색

Selenium 과 PhantomJS 활용 다음 뉴스 댓글 가져오기

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

driver = webdriver.PhantomJS('/usr/local/Cellar/phantomjs/2.1.1/bin/phantomjs')
driver.get('http://v.media.daum.net/v/20170202180355822')
try:
    element = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "alex-area"))
    )
    print(element.text)
except TimeoutException:
    print("해당 페이지에 alex-area 을 ID 로 가진 태그가 존재하지 않거나, 해당 페이지가 10초 안에
finally:
    driver.quit()
```

다음 뉴스 댓글 가져오기

- http://v.media.daum.net/v/20170922175202762 페이지의 타이틀과 댓글 3개(추천순) 을 다음과 같이 출력하기
- 출력 포멧:

```
댓글1
댓글2
댓글3
```

- 고려 사항:
  - 추천순 댓글이 3개 이하일 경우에는 댓글 갯수만큼 출력할 것

다음 뉴스 댓글 가져오기

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException

driver = webdriver.Chrome('/usr/local/Cellar/chromedriver/chromedriver')
driver.get("http://v.media.daum.net/v/20170922175202762")

try:
    element = WebDriverWait(driver, 3).until(
        EC.presence_of_element_located((By.ID, "alex-area"))
    )
except:
    print("댓글 관련 태그가 없습니다.")
else:
    loop = True
    comment_box = driver.find_element_by_css_selector("#alex-area > div > div > (
    comment_list = comment_box.find_elements_by_tag_name("li")
    for num, comment_item in enumerate(comment_list):
        print(comment_item.find_element_by_css_selector("div p").text)
driver.quit()
```

특정 웹사이트에 이름 입력하고, 제출하기

> 브라우저에서 마우스와 키보드 동작을 프로그래밍으로 제어하기

- 주요 마우스와 키보드 동작 프로그래밍
  - element 클릭: element.click()
  - element 더블 클릭: element.double_click()
  - element 키보드 입력 전송: element.send_keys()
  - element 로 마우스 이동: element.move_to_element()

특정 웹사이트에 이름 입력하고, 제출하기

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver.common.keys import Keys
from selenium.webdriver import ActionChains

driver = webdriver.PhantomJS('C:/dev_python/phantomjs-2.1.1-windows/bin/phantomj
# driver = webdriver.PhantomJS('/usr/local/Cellar/phantomjs/2.1.1/bin/phantomjs'
driver.get("http://pythonscraping.com/pages/files/form.html")

firstnameField = driver.find_element_by_name("firstname")
lastnameField = driver.find_element_by_name("lastname")
submitButton = driver.find_element_by_id("submit")

firstnameField.send_keys("Doky")
lastnameField.send_keys("Kim")
submitButton.click()

print(driver.find_element_by_tag_name("body").text)

driver.close()
```

특정 웹사이트에 이름 입력하고, 제출하기

- 여러 마우스와 키보드 동작 한번에 묶어서 실행하기
  - ActionChains(): 행동 여러 개를 체인 으로 묶어서 저장하고 원하는 만큼 실행
  - perform() 메서드 실행시 전체 행동을 실행함

특정 웹사이트에 이름 입력하고, 제출하기

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver.common.keys import Keys
from selenium.webdriver import ActionChains

driver = webdriver.PhantomJS('C:/dev_python/phantomjs-2.1.1-windows/bin/phantomj
# driver = webdriver.PhantomJS('/usr/local/Cellar/phantomjs/2.1.1/bin/phantomjs'
driver.get("http://pythonscraping.com/pages/files/form.html")

firstnameField = driver.find_element_by_name("firstname")
lastnameField = driver.find_element_by_name("lastname")
submitButton = driver.find_element_by_id("submit")

actions = ActionChains(driver).click(firstnameField).send_keys("Doky").click(las
actions.perform()

print(driver.find_element_by_tag_name("body").text)

driver.close()
```

다음 뉴스 댓글 가져오기

- http://v.media.daum.net/v/20170922175202762 페이지의 타이틀과 댓글 10개(추천순) 을 다음과 같이 출력하기
- 출력 포멧:

```
댓글1
댓글2
댓글3
```

- 고려 사항:
  - 추천순 댓글이 10개 이하일 경우에는 댓글 갯수만큼 출력할 것

다음 뉴스 댓글 가져오기

- 뉴스 하단부의 더보기 버튼을 자동으로 누른 후, 데이터를 가져오는 것이 핵심

```
    loop = True
    count = 0
    while loop and count < 1:
        try:
            element = WebDriverWait(driver, 3).until(
                EC.presence_of_element_located((By.CSS_SELECTOR, "#alex-area > d
            )
            more_button = driver.find_element_by_css_selector("#alex-area > div >
            webdriver.ActionChains(driver).click(more_button).perform()
            count += 1
        except TimeoutException:
            loop = False
```

다음 뉴스 댓글 가져오기1

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException

driver = webdriver.Chrome('/usr/local/Cellar/chromedriver/chromedriver')
driver.get("http://v.media.daum.net/v/20170922175202762")

print("[" + driver.find_element_by_tag_name('title').get_attribute('text') + "]"
try:
    element = WebDriverWait(driver, 3).until(
        EC.presence_of_element_located((By.ID, "alex-area"))
    )
```

다음 뉴스 댓글 가져오기2

```
except:
    print("댓글 관련 태그가 없습니다.")
else:
    loop, count = True, 0
    while loop and count < 1:
        try:
            element = WebDriverWait(driver, 3).until(
                EC.presence_of_element_located((By.CSS_SELECTOR, "#alex-area > d
            )
            more_button = driver.find_element_by_css_selector("#alex-area > div >
            webdriver.ActionChains(driver).click(more_button).perform()
            count += 1
        except TimeoutException:
            loop = False
    comment_box = driver.find_element_by_css_selector("#alex-area > div > div > (
    comment_list = comment_box.find_elements_by_tag_name("li")
    for num, comment_item in enumerate(comment_list):
        if num < 10:
            print(comment_item.find_element_by_css_selector("div p").text)
        else:
            break
driver.quit()
```