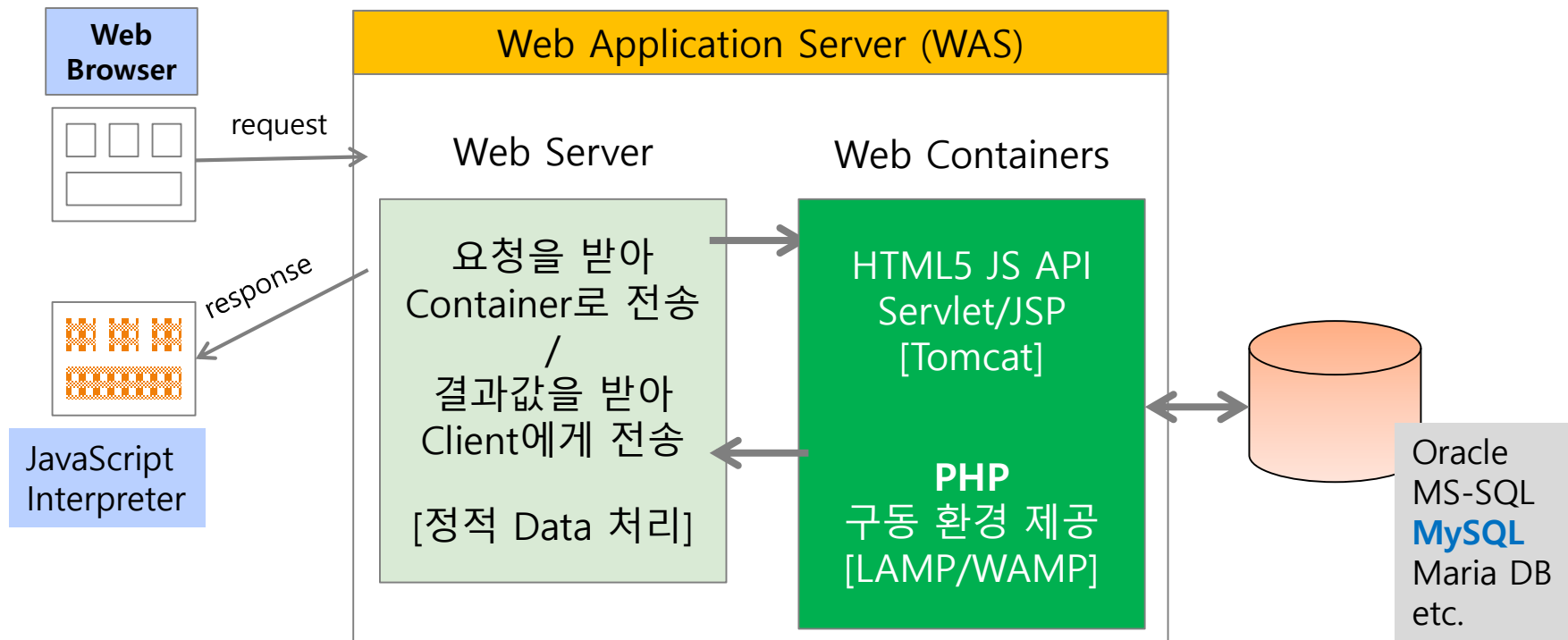


jQuery and Ajax: Please pass the data

Samkeun Kim <skim@hknu.ac.kr>

<http://cyber.hknu.ac.kr/>

Web Application Development Environment



[Apache/IIS/Tomcat] +
PHP 가능

대부분 서버 호스팅 업체
⇒ Apache web server +
PHP 지원

Dothome
⇒ PHP+MySQL 지원

- Tomcat (+ Catalina => Servlet/JSP)
- **LAMP/WAMP**

L => Linux, W => Windows
A => Apache web server
M => MySQL database
P => PHP Hypertext Preprocessor

⇒ Bitnami Stack [사용법: 첨부파일 참조]

관리 도구:

- MySQL Workbench
- **phpMyAdmin**

⇒ 웹 기반
MySQL 관리
도구

From: **Webville MegaCorps Marketing**
Subject: **42nd Annual Bit to Byte Race results page**

Hey Web Design Team,

As you're all aware, every year we sponsor Webville's Annual Bit to Byte 10K run by providing the race results page. But our page is way behind the times, as we only update it after all the results are in. People want instant gratification, and with Twitter and Facebook, folks attending the race are beating us at providing real-time results.

So we've got a challenge for you with a sweet payoff. If you can update our Webville Results page by next week to provide real-time results, you'll get to hang out in the VIP section at the end of the race. (Oh, and did we mention the race is in Maui this year?)

Here's what we need:

- 1) The page should provide the option to show either male or female runners, or all participants at once.
- 2) It should provide automatic updates as runners cross the finish line.
- 3) People shouldn't have to refresh the page as the results update.
- 4) Lastly, we want to indicate on the page when it was last updated and the frequency of the updates, and to enable people the ability to start and stop the updates if they want.

It doesn't look super different from last year's page, so that would be a good place to start. This is a great event, so we can't wait to see what you come up with!

--
Dionah C. Housney
Head of Marketing
Webville MegaCorp

It looks like the web team is already set to go to Hawaii... guess it's time to figure this out!



사용자 요구사항

매년 열리는 웹빌의 Bit to Byte 10K 마라톤 홈페이지에 레이스 결과 페이지를 제공해 왔다. 그러나 늘 모든 결과가 나온 다음에야 페이지를 업데이트했기 때문에 실시간과는 거리가 멀었다.

경기 참가자들은 트위터와 페이스북을 통해 실시간으로 결과를 제공해 달라고 우리를 압박해 오고 있다.

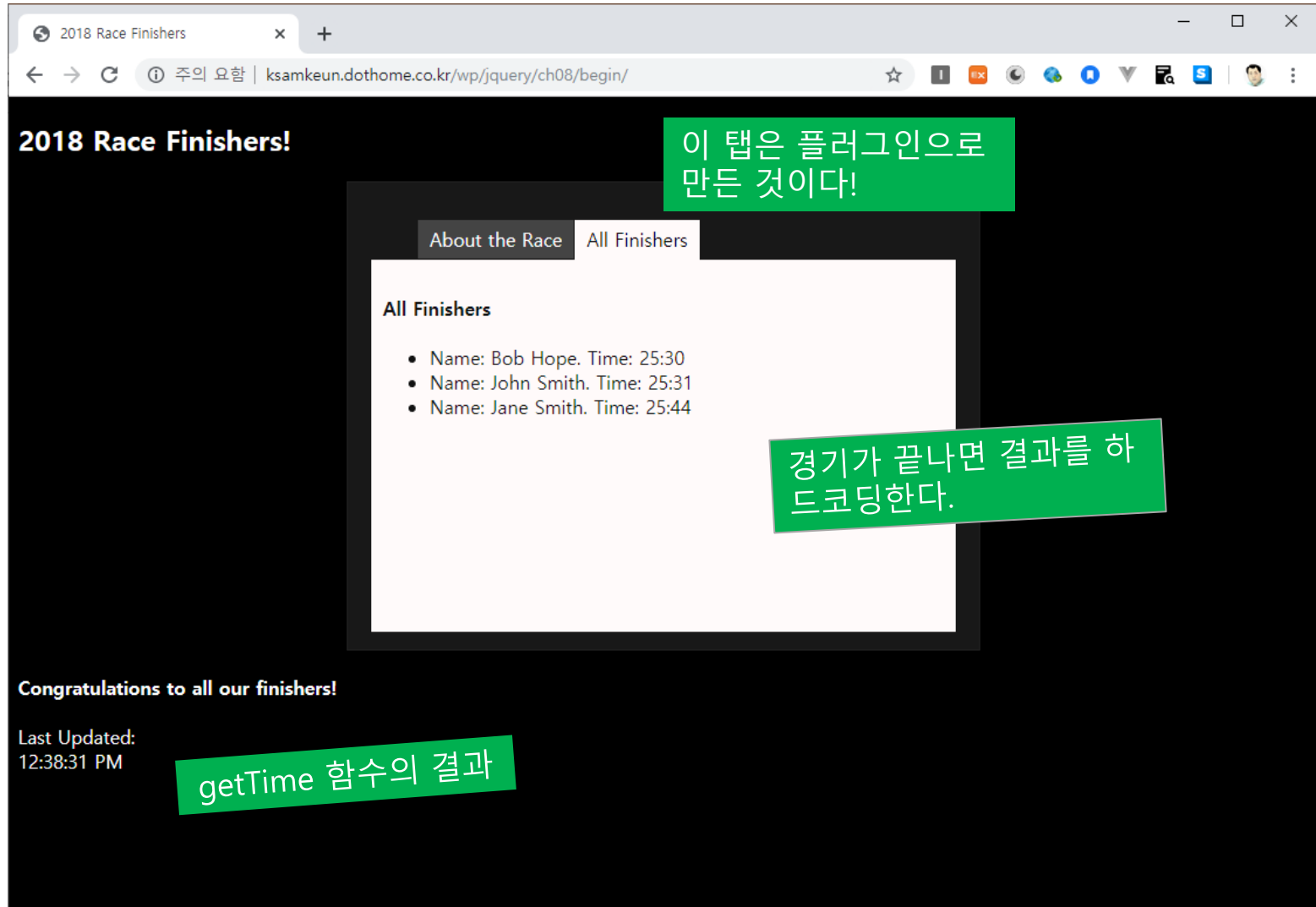
결과 페이지를 **실시간으로** 만들어 주면 좋겠다.

세부 요구사항:

- 페이지에는 남성 참가자만 표시하는 옵션, 여성 참가자만 표시하는 옵션, 그리고 모든 참가자를 한꺼번에 표시하는 옵션이 있어야 한다.
- 참가자가 결승점을 통과할 때마다 자동으로 업데이트되어야 한다.
- 사용자가 직접 페이지를 업데이트할 필요는 없어야 한다.
- 마지막으로 최종 업데이트를 언제 했는지 그리고 업데이트 주기가 어떻게 되는지 페이지에 표시하고 사용자가 업데이트를 ON/OFF 할 수 있어야 한다.

작년 페이지와 크게 다르지는 않으니까 작년 페이지를 수정해서 사용해도 된다.

작년 페이지



Configuring a plug-in

플러그인

- ✓ Base jQuery 라이브러리에 대한 확장
- ✓ 기능을 개선시키거나 특정 작업을 더 쉽게 쓸 수 있도록 해준다

idTabs 플러그인 사용

[jquery.idTabs.min.js](#)

<http://ksamkeun.dotheme.co.kr/wp/jquery/ch08/ch08-ajax-xml-js.zip>
(jquery.idTabs.min.js, finishers.xml)

my_style.css, index.html, my_scripts.js

```
body{
    background-color: #000;
    color: white;
}

/* Style for tabs */
#main {
    color:#111;
    width:500px;
    margin:8px auto;
}
#main > li, #main > ul > li
{ list-style:none; float:left; }
#main ul a {
    display:block;
    padding:6px 10px;
    text-decoration:none!important;
    margin:1px 1px 1px 0;
    color:#FFF;
    background:#444;
}
```

A CSS comment

From here, the rest of the CSS is dedicated to building tabs on the page.

```
#main ul a:hover {
    color:#FFF;
    background:#111;
}
#main ul a.selected {
    margin-bottom:0;
    color:#000;
    background:snow;
    border-bottom:1px solid snow;
    cursor:default;
}
#main div {
    padding:10px 10px 8px 10px;
    *padding-top:3px;
    *margin-top:-15px;
    clear:left;
    background:snow;
    height: 300px ;
}
#main div a {
    color:#000; font-weight:bold;
}
```



탭으로 변환될 링크를 생성한다.

탭 내용을 저장할 div 엘리먼트

```
<div id="main">
  <ul class="idTabs">
    <li><a href="#about">About the Race</a></li>
    <li><a href="#finishers">All Finishers</a></li>
  </ul>
  <div id="about">
    <h4>About the race</h4>This race Bit to Byte Campaign!
  </div>
  <div id="finishers">
    <h4>All Finishers</h4>
    <ul id="finishers_all">
      <li>Name: Bob Hope. Time: 25:30</li>
      <li>Name: John Smith. Time: 25:31</li>
      <li>Name: Jane Smith. Time: 25:44</li>
      ...
    </ul>
  </div>
  ...
  <script src="scripts/jquery-1.6.2.min.js"></script>
  <script src="scripts/my_scripts.js"></script>
  <script src="scripts/jquery.idTabs.min.js"></script>
  ...
</div>
```

A portion of last year's runners, hardcoded into the page. This must have been a pain to update...

Include the JavaScript files, as per usual. We use the same method for including plug-ins.



index.html

```
$(document).ready(function() {
  getTime(); // Call our custom getTime function.
  function getTime() {
    var a_p = "";
    var d = new Date();
    var curr_hour = d.getHours();
    (curr_hour < 12) ? a_p = "AM" : a_p = "PM";
    (curr_hour == 0) ? curr_hour = 12 : curr_hour = curr_hour;
    (curr_hour > 12) ? curr_hour = curr_hour - 12 : curr_hour = curr_hour;
    var curr_min = d.getMinutes().toString();
    var curr_sec = d.getSeconds().toString();
    if (curr_min.length == 1) { curr_min = "0" + curr_min; }
    if (curr_sec.length == 1) { curr_sec = "0" + curr_sec; }
    $('#updatedTime').html(curr_hour + ":" + curr_min + ":" + curr_sec + " " + a_p);
  }
});
```

JavaScript Date 객체의 인스턴스

Methods of the Date object

JavaScript 3항 연산자



my_scripts.js

Getting dynamic

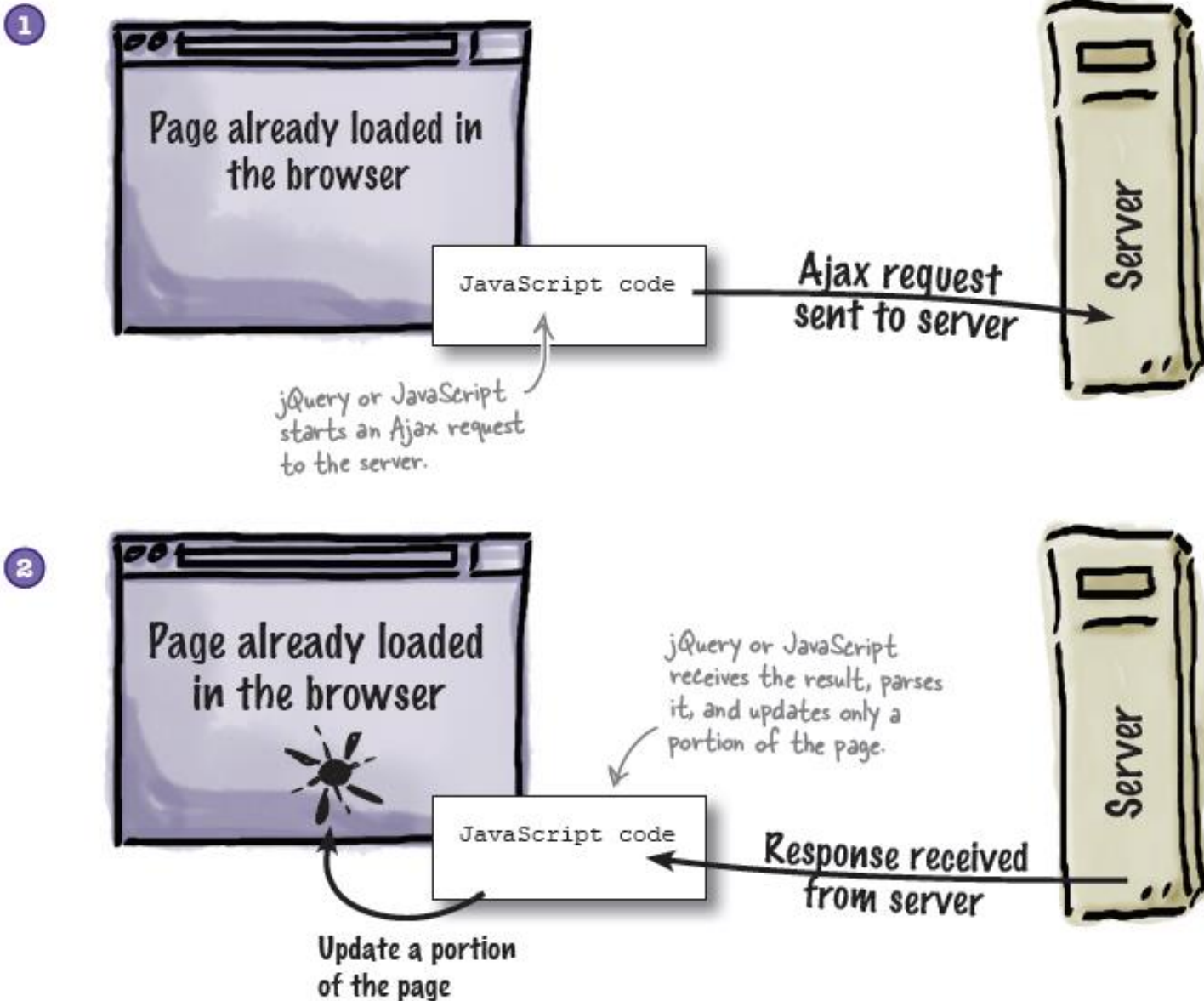
사용자 요구사항 => 페이지를 실시간으로 Update 되도록

예상 사용 기술 => jQuery, JavaScript, Ajax, XML

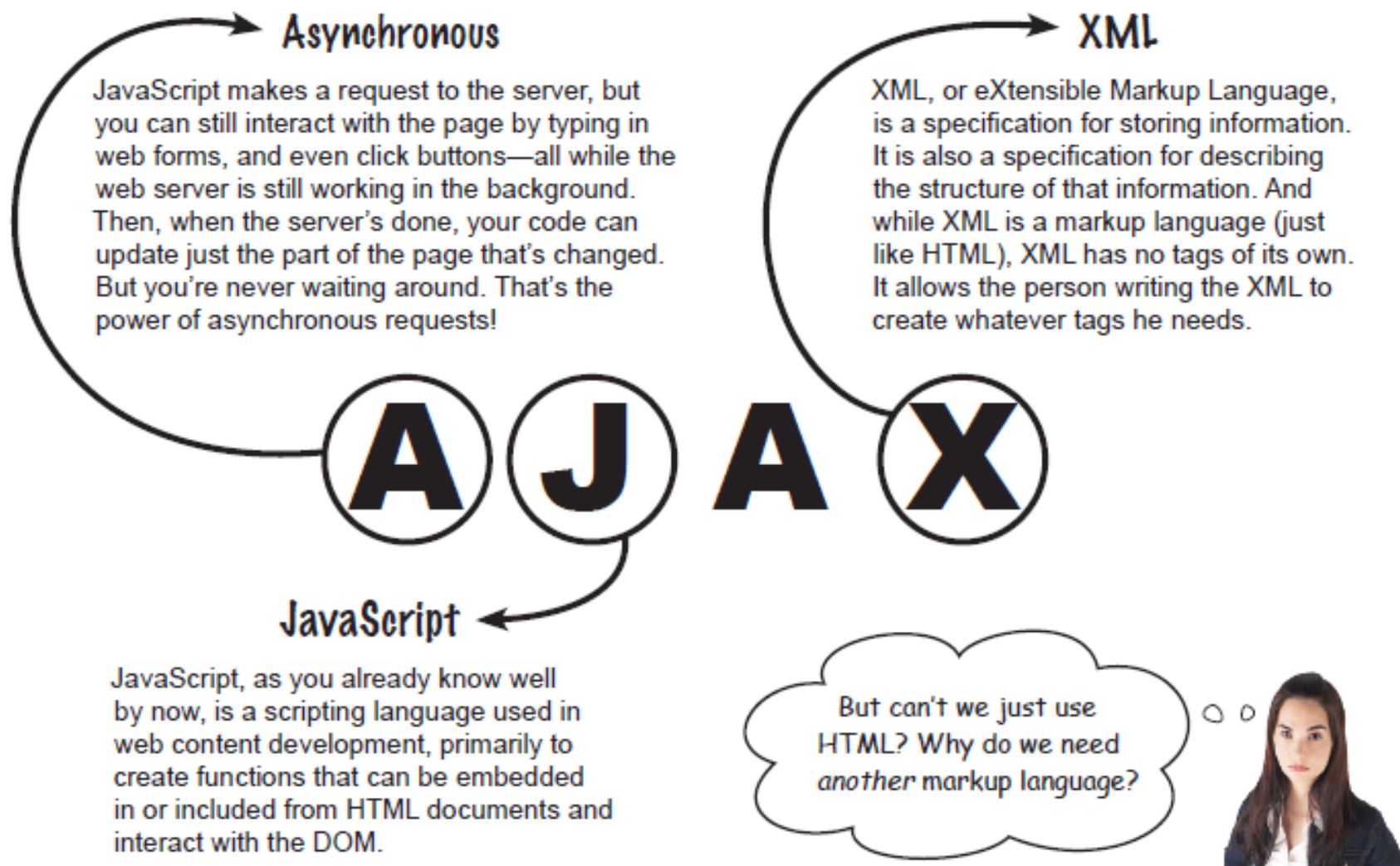
Ajax => Asynchronous JavaScript and XML

- ✓ 사용자를 방해하지 않고 브라우저와 웹 서버가 정해진 형식대로 데이터를 주고 받을 때 사용한다
- ✓ Ajax를 사용하면 웹 페이지나 애플리케이션에서 정말로 필요한 데이터, 서버에서 받아와야만 하는 데이터만 요청할 수 있다
- ✓ 가장 좋은 점 => Ajax는 이미 알고 있는 개념이다

Enter Ajax



Understanding Ajax



XML doesn't DO anything

The XML declaration. This should always be included. It defines the XML version of the document.

```
<?xml version="1.0" encoding="utf-8"?>
```

<books> ← The root node or tag

```
<book>
```

```
<title>The Hitchhikers Guide to the Galaxy</title>
```

```
<author>Douglas Adams</author>
```

```
<year>1980</year>
```

```
</book>
```

```
<book>
```

```
<title>The Color of Magic</title>
```

```
<author>Terry Pratchett</author>
```

```
<year>1983</year>
```

```
</book>
```

```
<book>
```

```
<title>Mort</title>
```

```
<author>Terry Pratchett</author>
```

```
<year>1987</year>
```

```
</book>
```

```
<book>
```

```
<title>And Another thing...</title>
```

```
<author>Eoin Colfer</author>
```

```
<year>2009</year>
```

```
</book>
```

```
</books> ← Close the root node or tag.
```

Other tags used to hold the data

Children of the root node (opening and closing tags). In this case, we are describing books.

XML is used to format data for transportation, while HTML is used to format and display data.



books.xml

HTML Code Magnets

```
<body>
  <header>
    <h2>_____</h2>
  </header>
  <div id="main">
    <ul class="idTabs">
      <li><a href="_____">Male Finishers</a></li>
      <li><a href="#female">_____</a></li>
      <li><a href="#all">All Finishers</a></li>
    </ul>
    <div id="male">
      <h4>Male Finishers</h4>
      <ul id="_____"></ul>
    </div>
    <div _____>
      <h4>Female Finishers</h4>
      <ul id="finishers_f"></ul>
    </div>
    <div _____>
      <h4>All Finishers</h4>
      <ul id="_____"></ul>
    </div>
  </div>
  <footer>
    <h4>Congratulations to all our finishers!</h4>
    <br>Last Updated: <div id="_____"></div>
  </footer>
  <script src="scripts/jquery-1.6.2.min.js"></script>
  <script src="_____"></script>
  <script src="scripts/jquery.idTabs.min.js"></script>
</body>
```



index.html

"scripts/my_scripts.js"

#male

"finishers_all"

Race Finishers!

"updateTime"

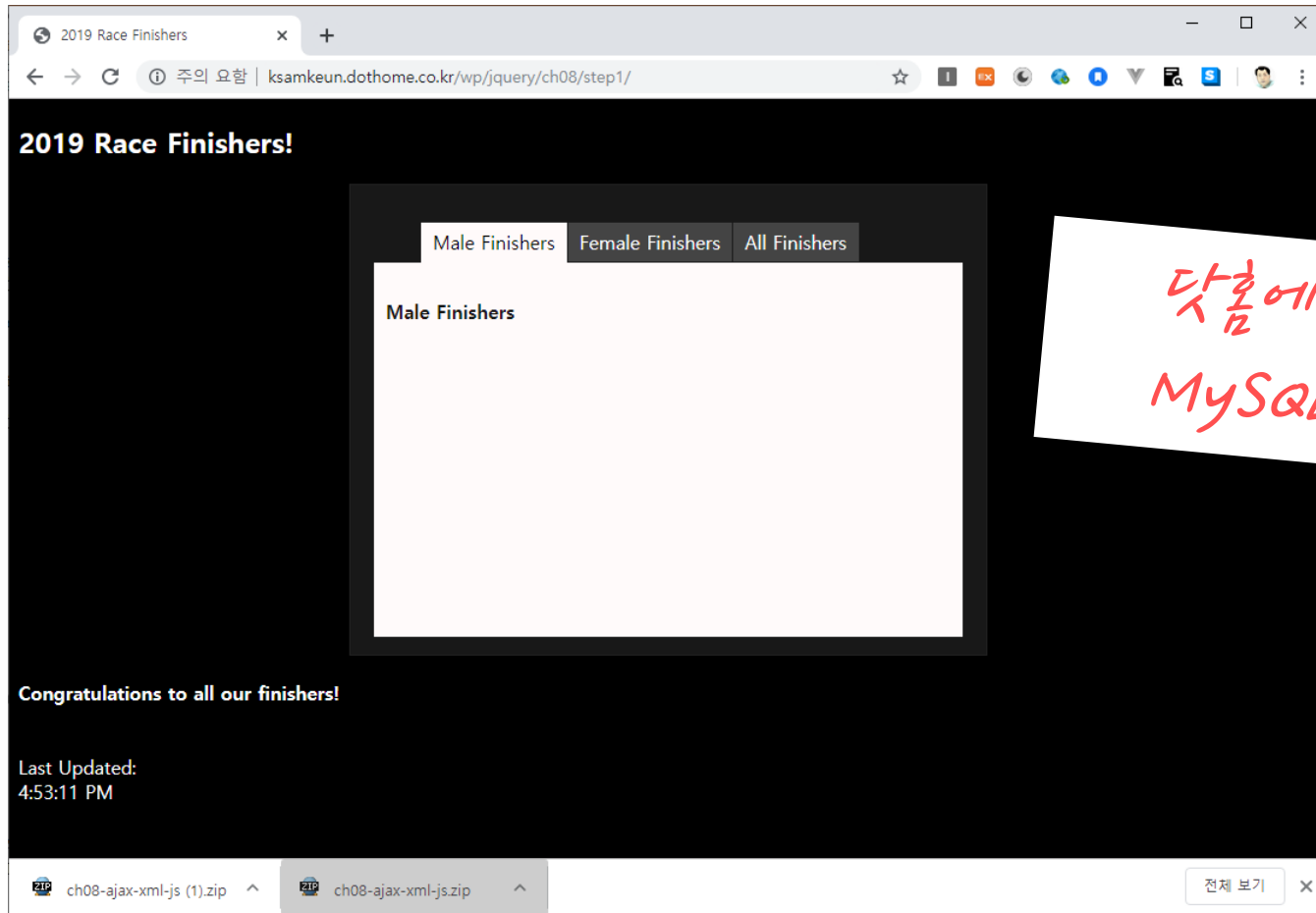
Female Finishers

finishers_m

id="female"

id="all"

실습과제 19-1 Test Drive



<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step1/>

GETting data with the ajax method

.ajax() 메소드는 여러 가지 파라미터를 받아 들어서 서버로부터 POST 데이터 또는 GET 데이터를 받을 수 있다.

The diagram illustrates the jQuery `.ajax()` method with several annotations:

- The jQuery shortcut**: Points to the `$.ajax({` part of the code, which is enclosed in a green box.
- The jQuery ajax method**: Points to the `.ajax()` part of the code.
- The URL of what you want to GET via Ajax**: Points to the `url: "my_page.html"` property.
- Run this function if the Ajax method is successful. We'll put more code in here in a bit.**: Points to the `success: function(data) {` block.
- The data returned from the Ajax call**: Points to the `data` parameter inside the success function.

```
$.ajax({  
  url: "my_page.html"  
  success: function(data) {  
  }  
});
```

GETting data with the ajax method

```
$(document).ready(function() {  
    $.ajax({  
        url: "finishers.xml",  
        cache: false,  
        dataType: "xml",  
        success: function(xml) {  
        }  
    });  
});
```

Load the finishers.xml
file via Ajax.

This parameter caches the
results locally. That can cut
down on calls to the server.

The data type we're
expecting to get back
from the server

```
getTime();
```

```
function getTime() {  
    var a_p = "";  
    var d = new Date();
```

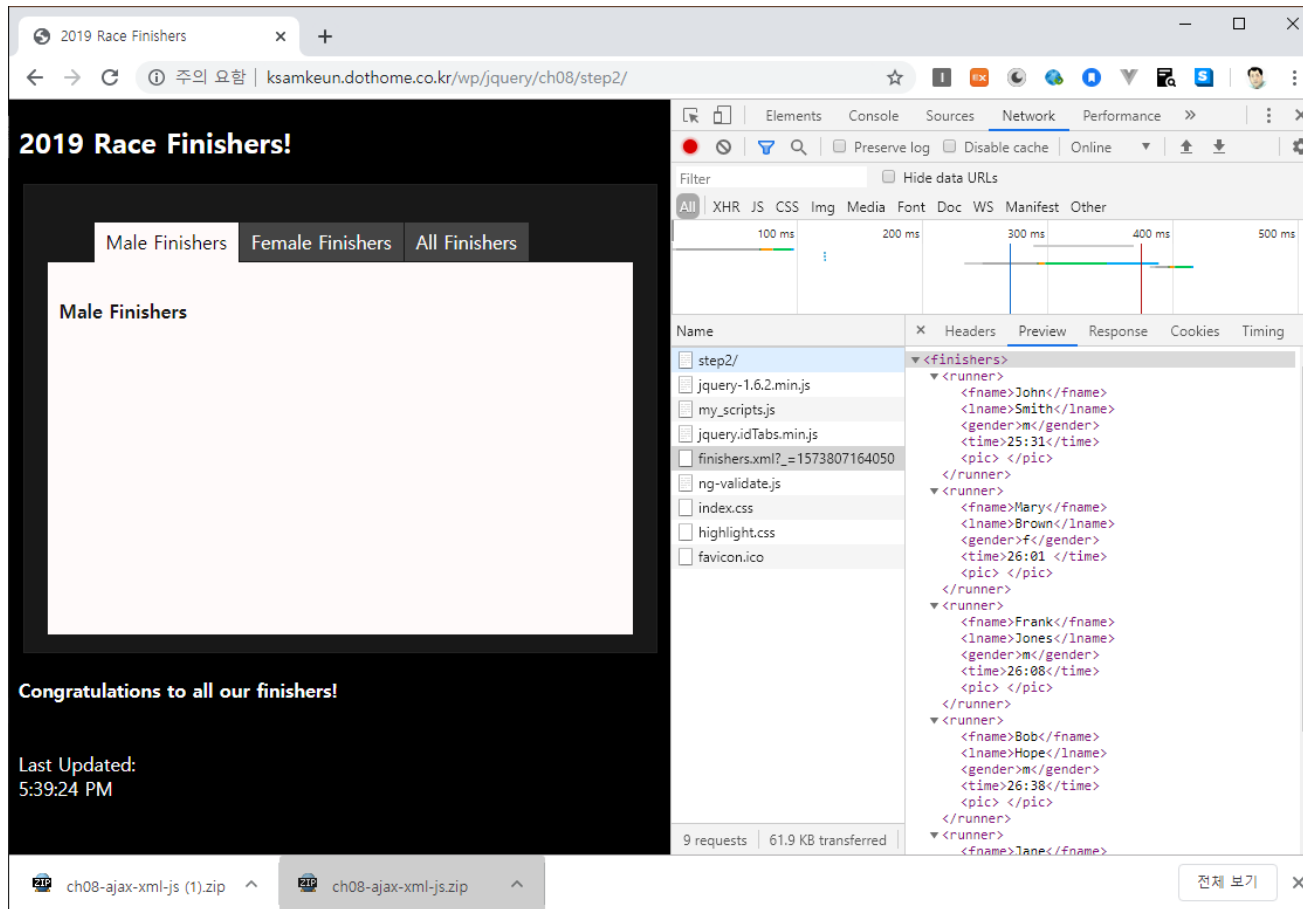


my_scripts.js

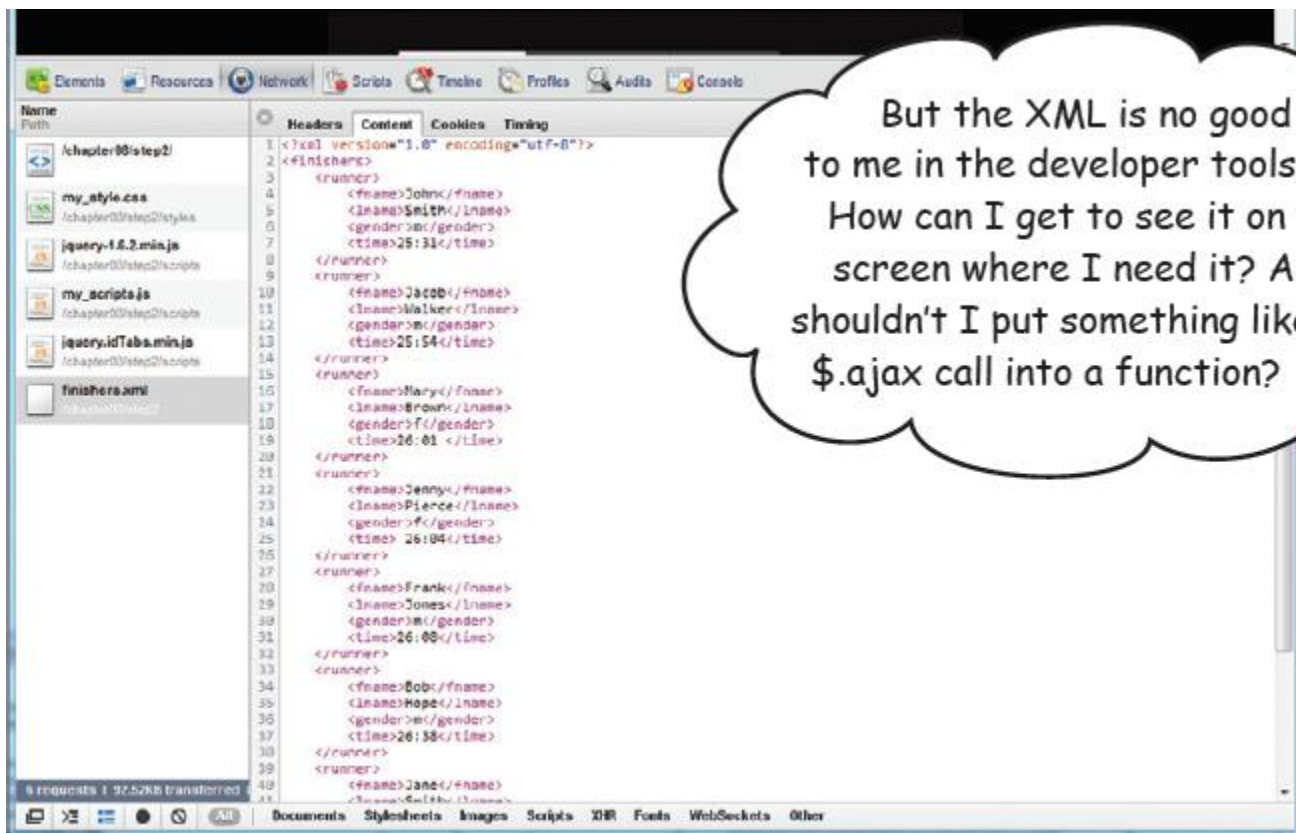
실습과제 19-2 Test Drive

앞 페이지의 스크립트 코드 업데이트

finishers.xml 파일 다운로드 -> F12 클릭 -> Network 패널 -> xml 파일 확인

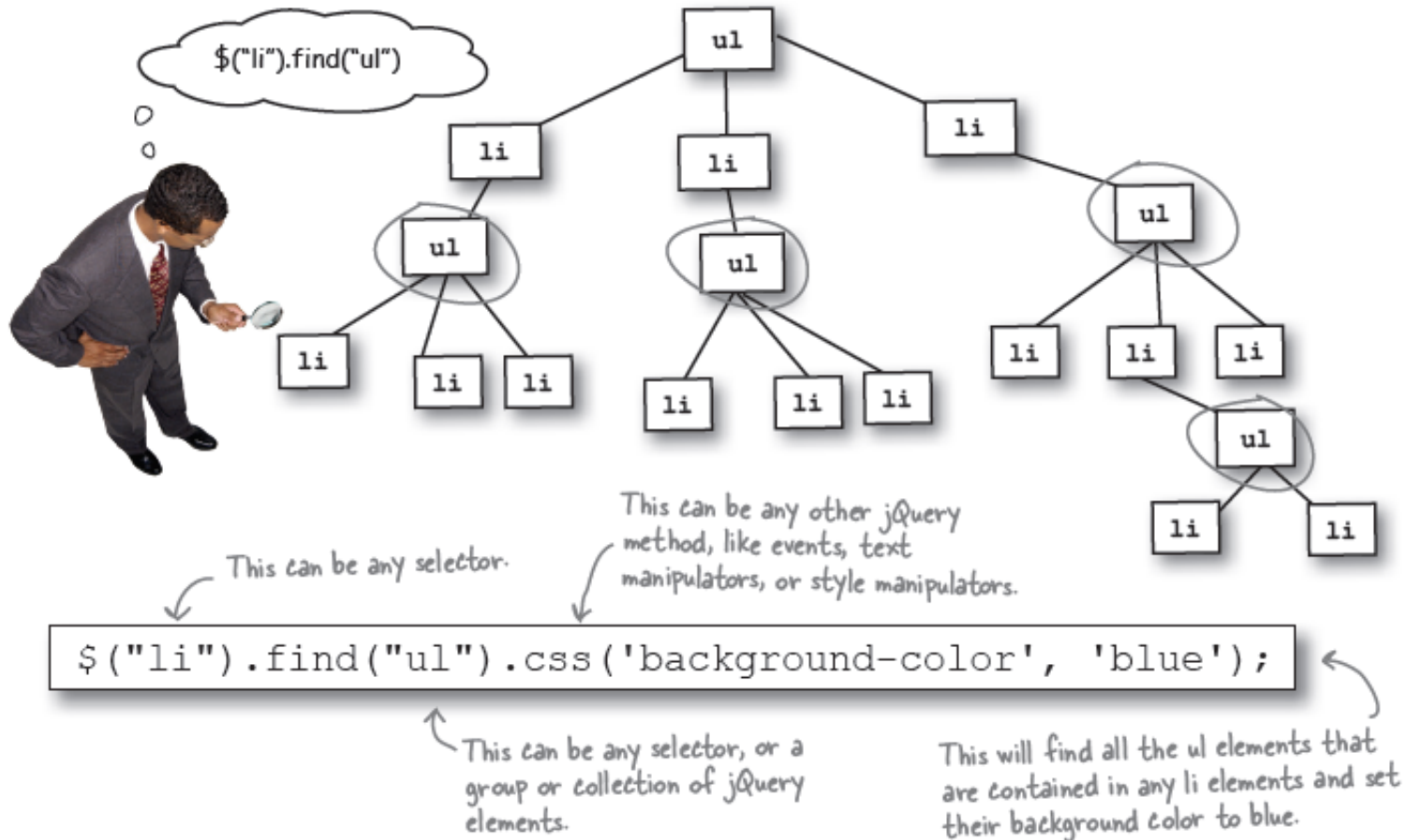


<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step2/>



Parsing XML data

DOM 트리: **.find()** => 파라미터로 받은 선택자에 일치하는 엘리먼트를 찾는 메소드



jQuery Code Magnets

```
function _____
$.ajax({
  url: _____,
  cache: false,
  dataType: "xml",
  _____ function(xml) {
    $(_____).empty();
    $('#finishers_f')_____
    $('#finishers_all').empty();
    $(xml).find_____ (function() {
      var info = '<li>Name: ' + $(this).find_____ + ' ' + $(this).
find("lname").text() + '. Time: ' + _____.text() + '</li>';
      if( $(this).find("gender").text() == "m" ){
        $('#finishers_m').append_____
      }else if ( $(this).find("gender").text() == "f" ){
        _____.append(info);
      }else{ }
        _____.append(info);
    });
    _____
  }
});
}
```

Code magnets for the above function:

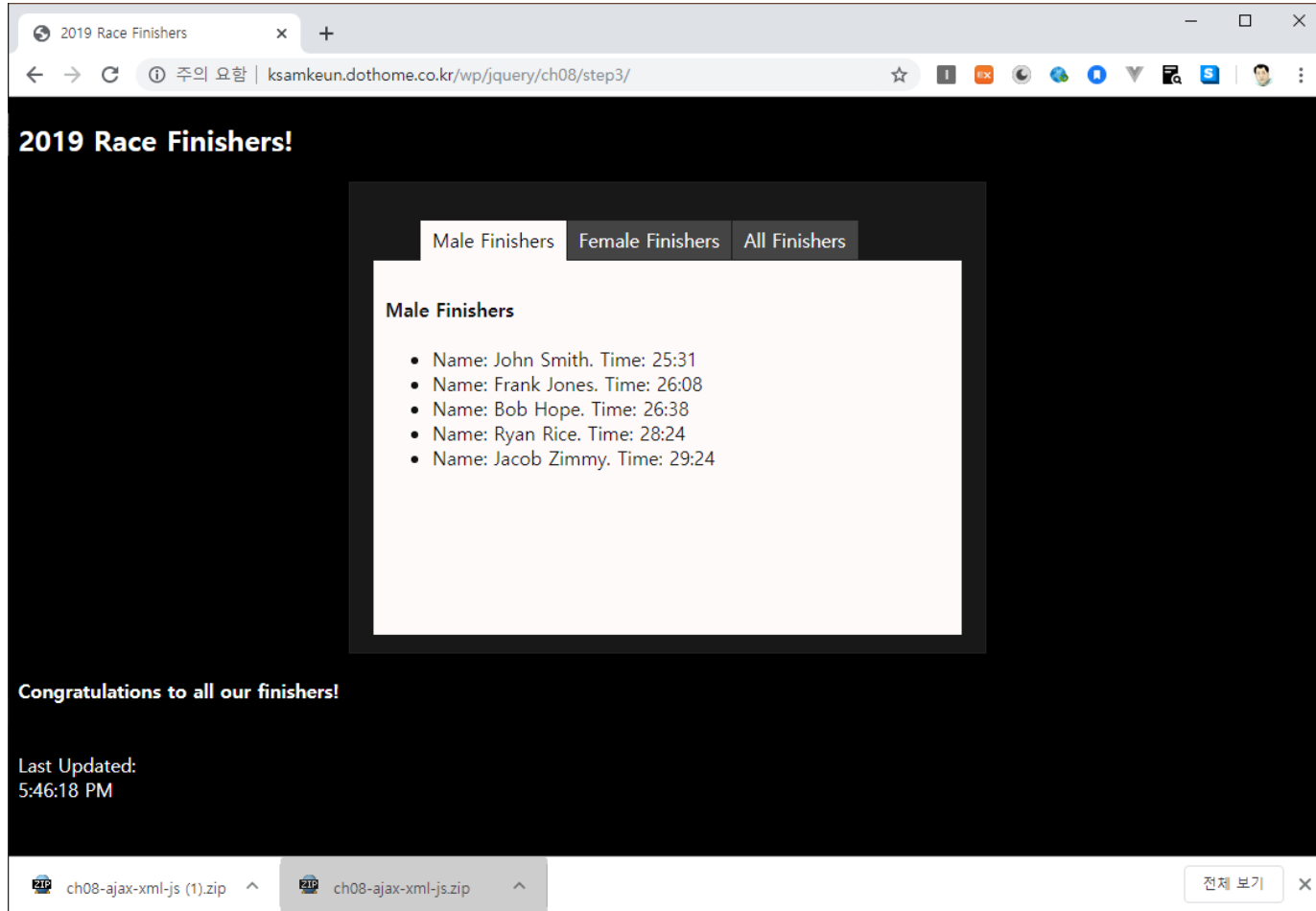
- "finishers.xml"
- getTime();
- success:
- (info)
- \$('#finishers_all')
- '#finishers_m'
- \$("#runner").each
- \$('#finishers_f')
- .empty()
- \$(this).find("time")
- ("fname").text()

Rearrange the code magnets to create a function called `getXMLRacers` that will call the `ajax` method and load the *finishers.xml* file. Once the load is successful, empty all the lists that will hold the information, and then find each runner in the XML file and determine whether the runner is male or female. Append the runners to the appropriate list for their gender, and always add them to the `finishers_all` list. Then, call the `getTime` function to update the time on the page.



실습과제 19-3 Test Drive

my_scripts.js 파일에 **getXMLRacers()** 함수 추가



<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step3/>

Scheduling events on a page

JavaScript의 window 객체에 있는 타이머 메소드


setTimeout, clearTimeout, setInterval, clearInterval

jQuery의 **.delay()**

- ✓ 주로 애니메이션의 타이밍 조절에 쓰이고 함수 예약 기능 지원 안함

JavaScript timer methods

setTimeout




Use me when you want to set a period of time to wait until telling a function to run.

```
setTimeout(myFunction, 4000);
```

The function to call when the timeout duration has passed.

타이머 시간
(in milliseconds)

setInterval



I tell a function to run repetitively with an interval of time between each repetition.


```
setInterval(repeatMe, 1000);
```

The function to repeat after each interval is up.

함수 사이의 대기 시간
(in milliseconds)

~~jQuery's delay method~~

~~delay~~



~~I add a pause between effects that are queued up in a chain of effects.~~

```
slideDown().delay(5000).slideUp();
```

~~When this chain runs, it's known in jQuery as an effects queue.~~

~~In this example, the~~
5초간 일시 정지
~~a five-second pause between the slideUp and slideDown effects.~~



It's obvious, isn't it? We use `setInterval`, just like last time. Right?

Not so fast!

항상 그렇다고 할 수는 없다!

페이지에서 함수를 반복 실행할 때는 보통 `setInterval`을 사용하는 하지만, 상황이 외부 자원(예를 들어, XML 파일)에 의존하고 있을 때는 문제가 생길 수도 있다.



Watch it!

`setInterval` will run even if the function it is calling isn't finished yet.

If you're waiting on information from another server, or waiting on user interaction, `setInterval` could call your function again before you're ready. Your functions may not always return in the order that you called them.

Self-referencing functions

자기 참조 함수

- 정상적인 실행과정에서 자기 자신을 호출하는 함수
- 함수를 다시 실행하기 전에 현재 실행중인 오퍼레이션이 끝나기를 기다려야만 할 때 특히 유용

자기참조 함수 + setTimeout

- 함수에 대한 이전 콜이 성공적이었을 때만 계속 실행되도록 스케줄 설정 가능
- 콜이 실패하면 더 이상 호출되지 않음

Sharpen your pencil

페이지 로딩이 완료된 후 10초마다 **getXMLRacers** 함수를 호출하는 **startAJAXcalls** 함수를 작성해 보자.

Set our **FREQ** variable to 10000, since the **setTimeout** function requires its parameter in milliseconds.

Since we wait until the last call to our function is finished, we use the **setTimeout** function.

페이지를 처음 불러왔을 때도 표시할 내용이 있어야 하므로 **getXMLRacers** 함수를 호출한다.

```
$(document).ready(function() {  
    var FREQ = 10000;  
  
    function startAJAXcalls() {  
        setTimeout(function() {  
            getXMLRacers();  
            startAJAXcalls();  
        }, FREQ);  
    }  
  
    getXMLRacers();  
    startAJAXcalls();  
  
    function getXMLRacers() {  
        $.ajax({  
            url: "finishers.xml",  
            cache: false,  


```

Call our **getXMLRacers** function repeatedly inside the **setTimeout**.

10초 후에 다시 실행되도록 호출한다.

Pass our variable in as a parameter.

새로 만든 함수를 호출한다.



my_scripts.js

실습과제 19-4 Test Drive

The screenshot shows a web browser window with the address bar displaying `ksamkeun.dothome.co.kr/wp/jquery/ch08/step4/`. The page title is "2019 Race Finishers!". Below the title, there are three tabs: "Male Finishers", "Female Finishers", and "All Finishers". The "Male Finishers" tab is selected, showing a list of five finishers:

- Name: John Smith. Time: 25:31
- Name: Frank Jones. Time: 26:08
- Name: Bob Hope. Time: 26:38
- Name: Ryan Rice. Time: 28:24
- Name: Jacob Zimmy. Time: 29:24

Below the list, it says "Congratulations to all our finishers!" and "Last Updated: 6:19:00 PM".

The Network console on the right shows a list of requests. The first request is `finishers.xml?_=1573809420168`. The response is an XML document:

```
<finishers>
  <runner>
    <fname>John</fname>
    <lname>Smith</lname>
    <gender>m</gender>
    <time>25:31</time>
  </runner>
  <runner>
    <fname>Frank</fname>
    <lname>Jones</lname>
    <gender>m</gender>
    <time>26:08</time>
  </runner>
  <runner>
    <fname>Bob</fname>
    <lname>Hope</lname>
    <gender>m</gender>
    <time>26:38</time>
  </runner>
  <runner>
    <fname>Ryan</fname>
    <lname>Rice</lname>
    <gender>m</gender>
    <time>28:24</time>
  </runner>
  <runner>
    <fname>Jacob</fname>
    <lname>Zimmy</lname>
    <gender>m</gender>
    <time>29:24</time>
  </runner>
</finishers>
```

A handwritten note with an arrow points to the XML structure:

Add this runner to your XML file and save it. Watch him get loaded onto your page automatically.

```
<runner>
  <fname>Justin</fname>
  <lname>Jones</lname>
  <gender>m</gender>
  <time>29:14</time>
</runner>
```

<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step4/>

Getting more from your server

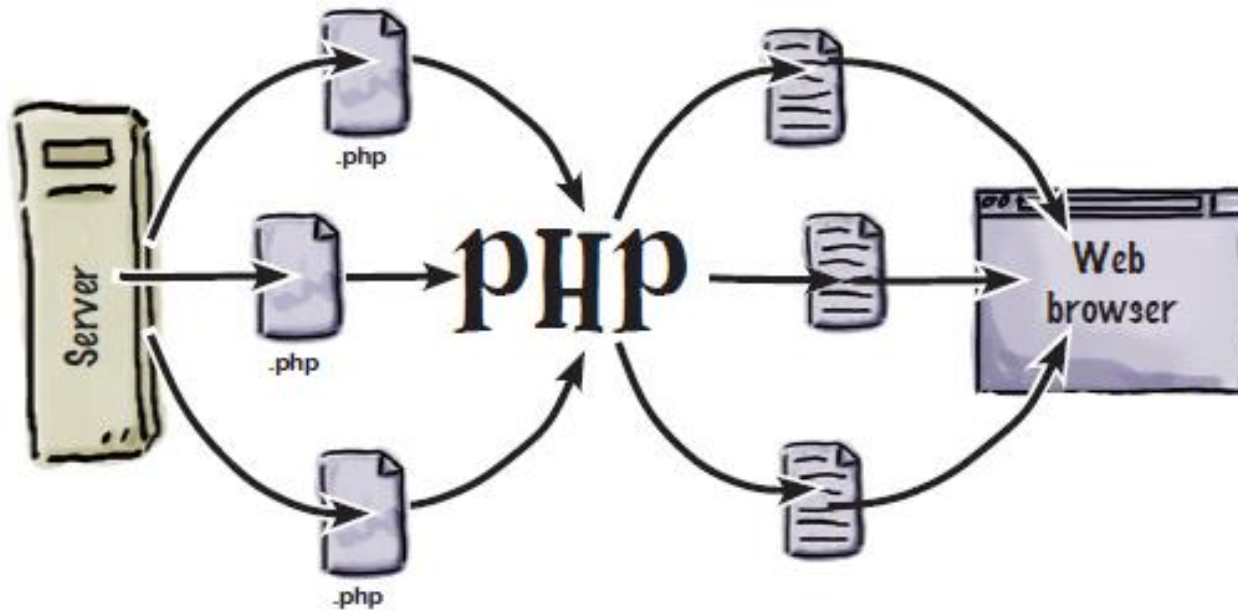
As we've seen so far, HTML is great for displaying information on a page and XML is great for formatting data for transportation to a page, but what if you need your page to actually **do** something, like tell the time or get data from a **database**?

Sure, we could probably do some more fun things with jQuery and JavaScript, but why not use something designed for the job?

Server-side languages to the rescue!

Server-side Scripting Languages: ASP, JSP, ColdFusion, **PHP**

PHP => PHP is Hypertext Preprocessor



PHP is used to dynamically produce HTML, which is then displayed in the browser.

What time is it?

Call the PHP date function.

Anything inside the square brackets, [and], is an optional parameter.

```
date (string $format [, int $timestamp = time() ])
```

PHP 또한 \$를 사용한다.
그러나, 단지 변수에 대해서만 사용!!

Pass in a parameter for the format of the date you want returned. This is a string.

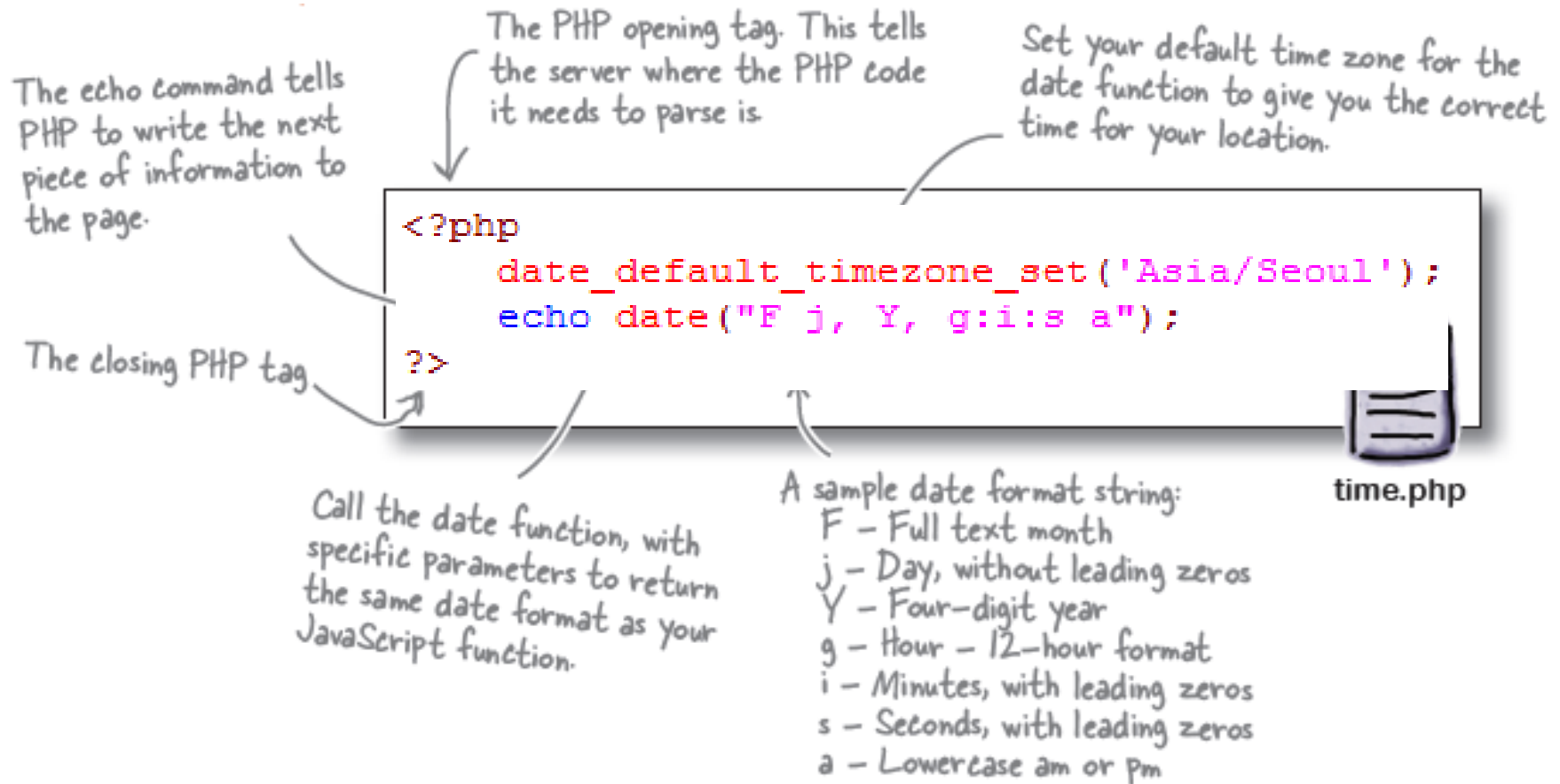
Always end each line in PHP with a semicolon.

Date 함수에 사용할 수 있는 파라미터를 모두 보려면

<http://php.net/manual/en/function.date.php>

What time is it?

Create a new file in the same folder as your *index.html* file, and call it *time.php*. Add the following code to your new *time.php* file.



The diagram shows a code editor window titled `time.php` containing the following PHP code:

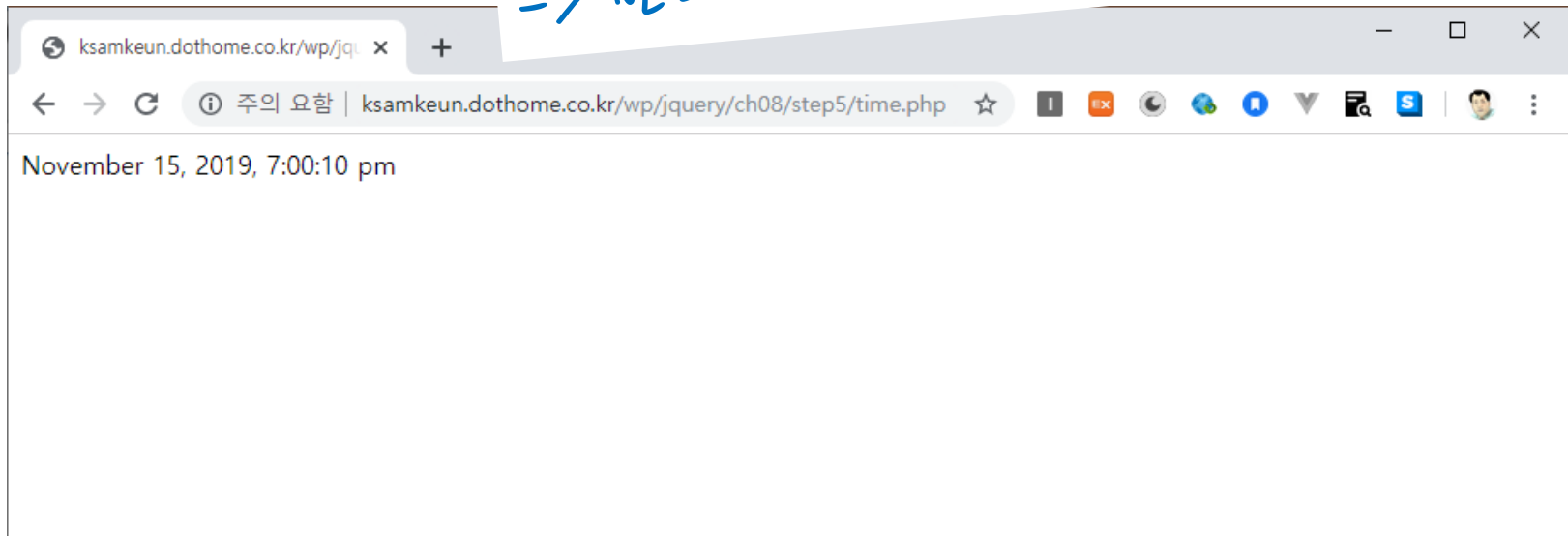
```
<?php
    date_default_timezone_set('Asia/Seoul');
    echo date("F j, Y, g:i:s a");
?>
```

Annotations with arrows pointing to the code:


- The PHP opening tag. This tells the server where the PHP code it needs to parse is.** (points to `<?php`)
- Set your default time zone for the date function to give you the correct time for your location.** (points to `date_default_timezone_set('Asia/Seoul');`)
- The echo command tells PHP to write the next piece of information to the page.** (points to `echo date("F j, Y, g:i:s a");`)
- The closing PHP tag.** (points to `?>`)
- Call the date function, with specific parameters to return the same date format as your JavaScript function.** (points to the format string `"F j, Y, g:i:s a"`)
- A sample date format string:**
 - F - Full text month
 - j - Day, without leading zeros
 - Y - Four-digit year
 - g - Hour - 12-hour format
 - i - Minutes, with leading zeros
 - s - Seconds, with leading zeros
 - a - Lowercase am or pm

실습과제 19-5 Test Drive

Dothome: PHP, MySQL 지원
=> 반드시 dothome에 업로드하여 실행할 것!!



<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step5/time.php>



I am SO ready to get rid of this pasty-white skin. PHP made the updated time bit a snap, but don't we have a few requirements we still haven't gotten to?

Yeah, we're not getting on that plane quite yet. Let's see what's left to do:

- 1 We want to indicate on the page how recently the page was updated.
- 2 We want to indicate the frequency of the updates.
- 3 We want to give people the ability to stop and start the updates, should they so choose.

Sharpen your pencil – 1번 & 2번 해결

```
<footer>
  <h4>Congratulations to all our finishers!</h4>
  <span id="freq"></span>
  <br><br>
  Last Updated: <div id="updatedTime"></div>
</footer>
<script src="scripts/jquery-1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
```

Add the span element to display the frequency.



index.html

```
function showFrequencyOf{
  $("#freq").html( "Page refreshes every " + FREQ/1000 + " second(s).");
}
```

Create two new functions, one to show the frequency, and the other to get the time from the server via Ajax.

Divide by 1,000 to convert milliseconds to seconds.

load: Ajax convenience method

```
function getTimeAjaxOf{
  $("#updatedTime").load( "time.php" );
}
```

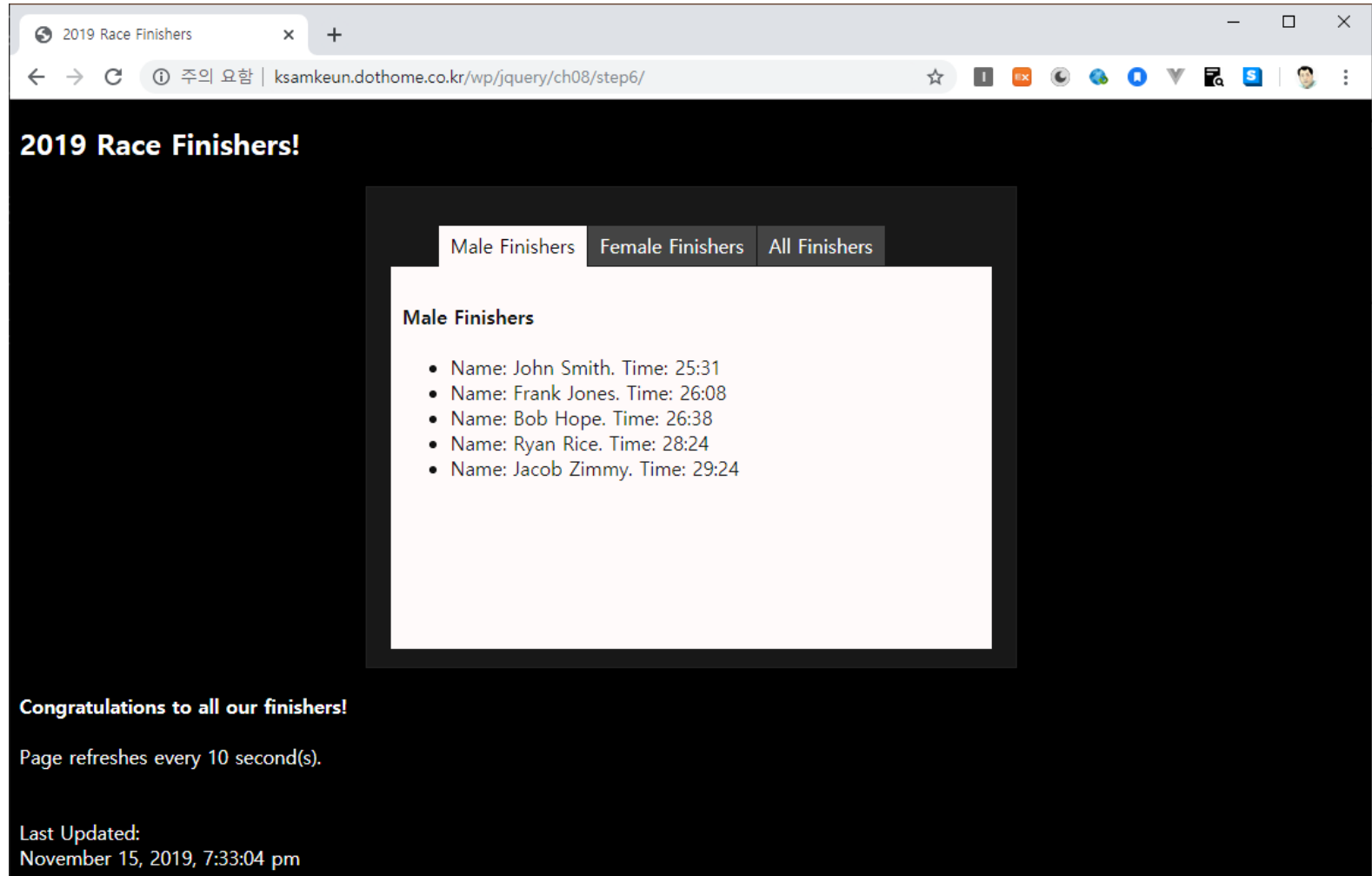
Output the result to the screen in the element updatedTime.

Load the time.php file using Ajax.



my_scripts.js

실습과제 19-6 Test Drive



2019 Race Finishers

← → ↻ ⓘ 주의 요함 | ksamkeun.dothome.co.kr/wp/jquery/ch08/step6/ ☆ ⓘ

2019 Race Finishers!

Male Finishers

Female Finishers

All Finishers

Male Finishers

- Name: John Smith. Time: 25:31
- Name: Frank Jones. Time: 26:08
- Name: Bob Hope. Time: 26:38
- Name: Ryan Rice. Time: 28:24
- Name: Jacob Jimmy. Time: 29:24

Congratulations to all our finishers!

Page refreshes every 10 second(s).

Last Updated:
November 15, 2019, 7:33:04 pm

<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/step6/>

- ~~1 We want to indicate on the page how recently the page was updated.~~
- ~~2 We want to indicate the frequency of the updates.~~
- 3 We want to give people the ability to stop and start the updates, should they so choose.

Turning off scheduled events on your page

setTimeout => 끊임없이 함수 호출

그렇다고 함수에 대한 이전 호출이 종료될 때까지 기다려야 하기 때문에 **setInterval**로 바꿀 수도 없다.

window.onblur 나 **window.onfocus** 같은 브라우저 이벤트도 사용할 수 없다

조건 로직에 기반한 코드 작성 필요!!

Repeat라는 전역변수를 만들고 기본값으로 true를 저장하자.

btnStop이란 ID를 갖는 버튼을 만들고, 이 버튼을 클릭했을 때 repeat 변수의 값을 바꾸게 하자.

버튼을 클릭하면 ID가 freq인 span 엘리먼트의 내용을 'updates paused'로 바꾸자.

ID가 btnStart인 버튼을 만들고, 이 버튼을 클릭했을 때 repeat 변수를 true로 지정하고 startAJAXcalls 함수와 setTimeout 함수를 모두 실행하자.

Set the variable back to true when the btnStart button is clicked. Also call the startAJAXcalls function to start getting the file again.

새로 만든 두 버튼을 페이지 footer에 추가하자.

```
$(document).ready(function() {  
    var repeat = true;  
    var FREQ = 10000;  
    function startAJAXcalls() {  
        if(repeat) {  
            setTimeout( function() {  
                getXMLRacers();  
                startAJAXcalls();  
            },  
                FREQ  
            );  
        }  
    }  
    $( "#btnStop" ).click(function() {  
        repeat = false;  
        $( "#freq" ).html( "Updates paused." );  
    });  
    $( "#btnStart" ).click(function() {  
        repeat = true;  
        startAJAXcalls();  
        showFrequency();  
    });  
});
```



my_scripts.js

```
<footer>  
    <h4>Congratulations to all our finishers!</h4>  
    <button id="btnStart">Start Page Updates</button>  
    <button id="btnStop">Stop Page Updates</button>  
    <br>  
    <span id="freq"></span> <br><br>
```



index.html

실습과제 19-7 Test Drive

2019 Race Finishers!

Male Finishers Female Finishers All Finishers

Male Finishers

- Name: John Smith. Time: 25:31
- Name: Frank Jones. Time: 26:08
- Name: Bob Hope. Time: 26:38
- Name: Ryan Rice. Time: 28:24
- Name: Jacob Zimmy. Time: 29:24

Congratulations to all our finishers!

Page refreshes every 10 second(s).

Last Updated:
November 15, 2019, 7:35:15 pm

<http://ksamkeun.dothome.co.kr/wp/jquery/ch08/end/>

Q & A

