

options parser

Parsing command line arguments

- [Home](#)
- [Java](#)
- [C/C++](#)
- [Databases/SQL](#)
- [PHP](#)
- [Javascript](#)
- [Data Structures](#)
- [Design Pattern Questions](#)
- [Operating Systems](#)
- [Recursion](#)
- [Networking](#)
- [Excel Interview Questions](#)
- [HTML5](#)
- [General/Miscellaneous](#)
- [Apache Interview Questions](#)
- [Non-Technical Questions](#)
- [Interviewing in India](#)
- [Working As a Software Engineer](#)
- [Job Advice For Programmers](#)
- [Financial Analyst Questions](#)
- [Puzzles](#)
- [Assortment of Knowledge](#)
- [American Vocabulary](#)
- [Technical Vocabulary](#)
- [Science Questions](#)
- [popup](#)
- [About](#)
- [searchresults](#)
- [newsletter-signup](#)

PHP

- [PHP Interview Questions](#)
- [Difference between == and === in PHP?](#)
- [How would you parse HTML in PHP?](#)
- [PHP: What are magic methods?](#)
- [PHP: Example of the __autoload function](#)
- [PHP: self vs \\$this](#)
- [PHP: self vs. static](#)
- [Find if string contains another string – php](#)
- [How to delete an element from an array in php?](#)
- [PHP to String](#)
- [Return JSON from PHP](#)
- [How to remove warnings in PHP?](#)
- [Advanced PHP Interview Questions And Answers](#)
- [Advanced PHP Interview Questions And Answers Part 2](#)
- [Advanced PHP Interview Questions And Answers Part 3](#)
- [How to return an array from a PHP function](#)
- [How to delete cookies in PHP](#)
- [PHP cookie versus session](#)
- [Can sessions work without cookies?](#)

Advanced PHP Practice Interview Questions And Answers

Here we present some more challenging practice PHP interview questions and answers that were asked in a real in questions are really good to not just test your PHP skills, but also your general web development knowledge. We t good practice by going through these questions. The questions are for intermediate to somewhat advanced PHP so beginner or fresher you should be able to understand the answers and explanations we give – but you may not be a Here is the first part of the question – read it carefully to really understand it, and we give a simple, easy to unders

Write a PHP script to report the total download size of any URL. You may not use any 3rd-party code that perfi

No HTML interface is necessary for this exercise; you can write this as a command-line script that accepts the

For a single-file resource such as an image or SWF, the script would simply report on the total size of the docu

For a complex resource such as an HTML document, the script would need to parse it to find references to eml files, iframes, etc.

The goal of this exercise is to output the following information for a given URL:

- total number of HTTP requests
- total download size for all requests

So, there are 2 primary goals that this question asks us to solve: For any URL, find the total number of HTTP requ total download size for all requests. You may not understand what is meant by an HTTP request, but don't worry

We'll have to break down this question into more manageable pieces since it is a lot to comprehend. So, we'll go v with the easier parts of the question first.

Accepting arguments in PHP scripts

The question says that “No HTML interface is necessary; you ca the URL as an argument”.

So, let's just say that we want to just write this as a command lin arguments inside a PHP command-line script?

Well, if we plan on having the script called from the command l the URL is passed as an argument, then inside the PHP script we variable “\$argv[1];”. Inside our PHP script the code to retrieve t like:

```
/* If this script is invoked as ourscript.php www.theurl.
then $argv[1] will hold the value www.theurl.com, and
that value will be stored in the $URL variable as we
*/
$URL = $argv[1];
```

That's very simple code – now, let's move on to other parts of tl

How to connect to a URL in PHP?

It should also be clear that we will need to somehow be able to connect to a URL and view the contents of the pag do this? Well PHP provides a library called [cURL](#), that may already be included in your installation of PHP by def you to connect to a URL and retrieve information from that page – like the HTML content of the page, the HTTP l see the use of cURL in our code below – don't worry if you've never used cURL before, it's fairly easy to underst

Understanding resources

If you are confused by what exactly is meant by the term “resource” in the question above, then you should just th So, a CSS file, a Javascript file, an HTML file, a SWF (a file used for Adobe Flash) file, an image file (jpg, png, e and as you know there are many more types of resources on the web.

The difference between single file resources and other resources

Email

Country





The question specifically calls HTML files **complex** resources because of the simple fact that HTML documents a single file resources like image files, and SWF files. A single file resource does not contain references to other resources, and that is why they are both considered single file resources. An HTML file, on the other hand, because it contains references to other resources, it is not considered to be a single file resource.

In order to retrieve a resource from the web server where that resource is stored, a web browser has to make an HTTP request.

► [PHP Programming Tutorials](#)

► [Job Interview Answers](#)

► [SQL Query Examples](#)

► [SQL Online Training](#)

► [Basic SQL Commands](#)

► [PHP For Beginners](#)

AdChoices

What exactly is an HTTP request?

The question asks for two major things from a URL – the total number of HTTP requests and the total download size enough to understand, but you may be confused by what exactly is meant by an HTTP request. HTTP is the protocol to visit a webpage, your browser will make an HTTP request to the server that hosts that webpage, and the server will return an HTTP response.

But, what is important to understand here, is that your browser will probably have to make **multiple** HTTP requests given a URL, because that webpage will probably have some CSS files to go along with it, some Javascript files, and those resources are a separate HTTP request – 2 image files, 2 Javascript files, and 2 CSS files means 6 separate HTTP requests at a time – so we can not have 1 request for 6 different resources, instead we must have 6 requests for them.

So, for the purpose of this interview question, we have to find out the number of HTTP requests that will be made now clear to you. We'll go more in depth on this later – and show some actual code – as we cover some other things.

How to find the download size of a file?

The question also asks us to find the total download size of a URL. But what if that URL passed into the script just a GIF file? Well, for a single file resource we just need to find the size of that particular file and then return it as the document we will need to find the total size of all resources that are embedded and included on the page and then remember that we want the **total download** size of a URL.

So, let's write a PHP function that will return the download size of a **single** file resource. How should we approach finding the download size of a single file resource on the web?

Well, there is an HTTP header called "Content-Length" which will actually tell us the size of a particular resource requested. So, all we have to do is use PHP's built-in "get_headers" function, which will retrieve all the HTTP headers in the request.

The get_headers function accepts a URL as an argument. So, the PHP code to retrieve the "Content-Length" header is:

```
function get_remote_file_size($url) {
    $headers = get_headers($url, 1);

    if (isset($headers['Content-Length']))
        return $headers['Content-Length'];

    //checks for lower case "L" in Content-length:
    if (isset($headers['Content-length']))
        return $headers['Content-length'];
}
```

But, there is actually a problem with this code: you will not *always* receive the Content-Length header in an HTTP request. The Content-Length header is not guaranteed to be sent back by the web server hosting that particular URL, because it depends on the server configuration. That you need an alternative that always works in case the approach above fails.

An alternative to using the content-length header

Well, we can actually download the file ourselves and then just get the download size for that URL. How can we do this? As we discussed above. Once we download the resource, we can retrieve the download size using the CURLINFO_SIZE_DOWNLOAD_TOTAL approach as a backup to our first approach, we can come up with this code (the code in red below is the new code)

```
function get_remote_file_size($url) {
    $headers = get_headers($url, 1);
```

```

if (isset($headers['Content-Length']))
    return $headers['Content-Length'];

//checks for lower case "L" in Content-length:
if (isset($headers['Content-length']))
    return $headers['Content-length'];

//the code below runs if no "Content-Length" header is found:

$c = curl_init();
curl_setopt_array($c, array(
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => array('User-Agent: Mozilla/5.0
    (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1.3)
    Gecko/20090824 Firefox/3.5.3'),
));
curl_exec($c);

$size = curl_getinfo($c, CURLINFO_SIZE_DOWNLOAD);

return $size;

curl_close($c);
}

```

How should we parse HTML in PHP?

What exactly is meant by the sentence “For a complex resource such as an HTML document, the script would need to include resources: javascript files, CSS files, iframes, etc.”?

Well, as you probably know, an HTML page often uses other files to render the HTML page – like CSS file(s) for functionality to the HTML page, and so on. But the question is how do we take an HTML page and find all of those resources? It's not easy to do if we are reading the HTML page with the human eye. But, we want to find these resources using a program, which is actually more complicated than it seems – and the process by which a program (like PHP) reads an HTML file and finds all the resources is known as *parsing* the HTML. Any text can be parsed, but we are exclusively focused on HTML for this question.

Parsing HTML in PHP is definitely something that you do not want to do on your own, because it is so complex – [in PHP](#). The best way to parse HTML in PHP is to use a library that already exists – because writing an entire library is a considered way too much work for an answer to an interview question.

Note that the question states that “You may not use any 3rd-party code that performs the entire task described below, but using a PHP library to help you with *part* of this question is perfectly OK. OK, so the interviewer if you are in doubt, but we know for sure that for this particular question there's no way that the interviewee can do this task without using a library to help you parse the HTML.

With that in mind, here is the library we plan on using: [PHP HTML parser](#).

Note that the instructions say: “For a single-file resource such as an image or SWF, the script would simply report the size of the file.”

This means that if the URL is single file resources like an image file, we can just return the size of the file and we can distinguish between a single-file resource and a non-single file resource? Well, we could just say that all non-HTML pages are single file resources, but you can read about in part 3, but we will pretend it is for the sake of keeping things simple.

But wait, you might be thinking – what about PHP, JSP, ASP and all of those pages? Well, of course there is some non-HTML pages, but once those pages are rendered in a browser they are all HTML pages, regardless of what their file extension is.

So, all we have to do in order to determine if a URL points to a single file resource is to see if it is an HTML page or not. If it is, the file is a single resource file.

Using the HTTP Content-Type Header

But how do we check to see if a webpage is an HTML page? Clearly we can't just look at the URL by itself, because the file extension does not tell us that. Well, once again we can use the HTTP headers to our advantage. The HTTP Content Type header.

And, if the Content-Type header is equal to “text/html”, then we know that we are dealing with an HTML page. If it is equal to “text/html”, then we know that we are dealing with a single file resource, and we can just return the size.

Let's write some code in PHP that will tell us if a given URL is actually an HTML page by checking the HTTP header. Here is the code:

```

function check_if_html($url){
    $ch = curl_init($url);

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_HEADER, TRUE);
    curl_setopt($ch, CURLOPT_NOBODY, TRUE);

    $data = curl_exec($ch);
    $contentType = curl_getinfo($ch, CURLINFO_CONTENT_TYPE );

    curl_close($ch);

    if (strpos($contentType,'text/html') !== false)

```

```

        return TRUE;    // this is HTML, yes!
    else
        return FALSE;
}

```

In the code above, we just use a simple cURL connection to the URL to retrieve the headers, and then check the content type "text/html". If it does, then we return true, otherwise we return false.

Then, we can add some code that will actually call the function to determine if a URL points to just a single resource.

```

/*
check to see if the URL points to an HTML page,
if it doesn't then we are dealing with a single
file resource:
*/
if (!check_if_html($URL))
{
$totalSize = get_remote_file_size($URL);

echo "Final Total Download Size: $totalSize Bytes ";

$totalNumResources += 1; //single resource is an HTTP request

echo " Final total HTTP requests: $totalNumResources" ;

return;
}

```

How to find the total number of HTTP requests

We mentioned that we would need to find the total number of HTTP requests generated by a given URL – let's figure it out for us. It's clear that we must have some variable that maintains a total count of all HTTP requests, and this variable will be used for more HTTP requests.

We know that images will be wrapped in an "img" tag – so if we just do a search for all img tags we can take a look at the given image. For each image we find, we can increment the variable that holds the total count of the HTTP requests. We will also be referenced inside "link" tags, and also for JavaScript files, which will be referenced inside "script" tags.

We will need to use the simple HTML DOM parser that we discussed earlier in order to find all of the references in the code. The code looks like – note that we are using the simple HTML DOM library functionality to parse through the HTML. We will use \$totalNumResources to hold the total number of resources, and another variable called \$totalSize to hold the total

```

include('simple_html_dom.php');

$URL = $argv[1];

// Create DOM from URL or file
$html = file_get_html($URL);

// find all images!!
foreach($html->find('img') as $element){

    $size = get_remote_file_size($element->src);

    $totalSize = $totalSize + $size;

    $totalNumResources += 1;

    /*
    echo "Total Size So Far: $totalSize.\n";

    echo "total resources: $totalNumResources .\n";

    echo "IMAGE SIZE: $size.\n";

    echo "$element->src.\n";
    */
}

// find all CSS files
foreach($html->find('link') as $element)
{
    if (strpos($element->href, '.css') !== false) {

        $size = retrieve_remote_file_size($element->href);

        echo "SIZE: $size.\n";

        $totalSize = $totalSize + $size;

        $totalNumResources += 1;

    }
}

// find all script tags
foreach($html->find('script') as $element)
{
    //make sure this is javascript
    if (strpos($element->src, '.js') !== false) {
        $size = get_remote_file_size($element->src);
    }
}

```

```

        echo " Javascript SIZE: $size.\n";

        $totalSize = $totalSize + $size;

        $totalNumResources += 1;
    }
}

```

The answer to Advanced PHP Interview Question Part 1

Finally, we present our complete answer to the advanced PHP interview question part 1 below – with all the source question. You can also continue on to [Part 2 of the PHP Interview Questions and Answers](#), or just click the next button.

```

include('simple_html_dom.php');

$URL = $argv[1];

$totalSize = 0;

$totalNumResources = 0;

/*
check to see if the URL points to an HTML page,
if it doesn't then we are dealing with a single
file resource:
*/

if (!check_if_html($URL))
{
    $totalSize = get_remote_file_size($URL);

    echo "Final Total Download Size: $totalSize Bytes ";

    $totalNumResources += 1; //a single resource is still an HTTP request

    echo " Final total HTTP requests: $totalNumResources" ;

    return;
}

/* at this point we know we are dealing with an HTML document
   which also counts as a resource, so increment the $totalNumResources
   variable by 1
*/

$totalNumResources += 1;

$html = file_get_html($URL);

// find all images:
foreach($html->find('img') as $element){

    $size = get_remote_file_size($element->src);

    $totalSize = $totalSize + $size;

    $totalNumResources += 1;

    /*
    echo "Total Size So Far: $totalSize.\n";

    echo "total resources: $totalNumResources .\n";

    echo "IMAGE SIZE: $size.\n";

    echo "$element->src.\n";
    */
}

// Find all CSS:
foreach($html->find('link') as $element)
{

    if (strpos($element->href, '.css') !== false) {

        $size = get_remote_file_size($element->href);

        $totalSize = $totalSize + $size;

        $totalNumResources += 1;

        /*
        echo "total resources: $totalNumResources .\n";

        echo "Total Size So Far: $totalSize.\n";

        echo "$element->href.\n";
        */
    }
}

//only output the ones with 'css' inside...
}

```

```

//find all javascript:
foreach($html->find('script') as $element)
{

//check to see if it is javascript file:
if (strpos($element->src, '.js') !== false) {

    $size = get_remote_file_size($element->src);

    //echo " JS SIZE: $size.\n";

    $totalSize = $totalSize + $size;

    $totalNumResources += 1;

    /*

    echo "Total Size So Far: $totalSize.\n";

    echo "total resources: $totalNumResources .\n";

    echo "$element->src.\n";
    */
}

}

echo "Final total download size: $totalSize Bytes" ;

echo "Final total HTTP requests: $totalNumResources";

function get_remote_file_size($url) {
    $headers = get_headers($url, 1);

    if (isset($headers['Content-Length'])) return $headers['Content-Length'];

    //this one checks for lower case "L" IN CONTENT-length:
    if (isset($headers['Content-length'])) return $headers['Content-length'];

    $c = curl_init();

    curl_setopt_array($c, array(
        CURLOPT_URL => $url,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_HTTPHEADER => array('User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US
    ));

    curl_exec($c);

    $size = curl_getinfo($c, CURLINFO_SIZE_DOWNLOAD);

    return $size;

    curl_close($c);

}

/*checks content type header to see if it is
an HTML page...
*/

function check_if_html($url){
    $ch = curl_init($url);

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_HEADER, TRUE);
    curl_setopt($ch, CURLOPT_NOBODY, TRUE);

    $data = curl_exec($ch);
    $contentType = curl_getinfo($ch, CURLINFO_CONTENT_TYPE );

    curl_close($ch);

    if (strpos($contentType, 'text/html') !== false)
        return TRUE;    // this is HTML, yes!
    else
        return FALSE;
}

```

If you see some improvements we can make to the code above, please let us know in the comments. Press next to : interview questions.

[▶ PHP Programming Tutorials](#)
[▶ Job Interview Answers](#)
[▶ SQL Query Examples](#)
[▶ SQL Online Training](#)
[▶ Basic SQL Commands](#)
[▶ PHP For Beginners](#)
AdChoices 
[Hiring? Job Hunting? Post a JOB or your RESUME on our JOB BOARD](#)
[Subscribe to our newsletter for more free interview questions.](#)
[Follow @programmerintvw](#)


+3 Recommend this on Google

FOLLOW Varoon Sahgal, Author of ProgrammerInterview on


[« Previous](#)
[Next »](#)

4 Comments Programmer Interview

Sort by Best ▾



Join the discussion...

**Nico** • 9 months ago

array_change_key_case() is what you'll want. Also, curl_close() after the return won't do a hell of a lo

1 ^ | ▾ • Reply • Share ›

**Abirami R** • 6 months ago

thank you...for the great article...

www.besanttechnologies.com

^ | ▾ • Reply • Share ›

**Ayesha Ali** • 6 months ago

Thank you

^ | ▾ • Reply • Share ›

**anil yadav** • 10 months ago

good tutorial

^ | ▾ • Reply • Share ›

ALSO ON PROGRAMMER INTERVIEW

Finally block

1 comment • a year ago



Swoop Diggity — OK, that's pretty huge... I didn't know finally block would be called from an explicit return contained in the try or catch block

Directory directive in .ht

1 comment • a year ago



dhruva — good to know plz?

Does Java have pointers?

2 comments • 5 months ago



Swoop Diggity — "You can not perform arithmetic operations on references. So, adding 1 to a pointer is not possible, but is possible in C++." I think you mean "... So, adding 1 to a reference is not possible..." : ...

Find trailing zeros in fac

3 comments • 5 months ago



sireesh — in java code, i

Subscribe

Add Disqus to your site

Privacy

Would you like to thank [ProgrammerInterview.com](http://www.programmerinterview.com) for being a helpful free resource? [Then why not tell a friend about us](#), or **simply add a link to this page from your webpage** using the HTML below.

Link

to this `<a href="http://www.programmerinterview.com/index.php/php-questions/php-interview-questions-and-answers/"`
page:

connect [programmerinterview.com](http://www.programmerinterview.com)

Please bookmark with social media, your votes are noticed and appreciated:



Copyright© 2013 | [Programmer Job Board](#) | [India Job Board for Programmers](#) | [About](#) |