PHP Simple HTML DOM Parser Manual

Index

- Quick Start
- How to create HTML DOM object?
- How to find HTML elements?
- How to access the HTML element's attributes?
- How to traverse the DOM tree?
- How to dump contents of DOM object?
- How to customize the parsing behavior?
- API Reference
- FAQ

Quick Start

Top

```
// Create DOM from URL or file
$html = file_get_html('http://www.google.com/');

// Find all images
foreach($html->find('img') as $element)
      echo $element->src . '<br>';

// Find all links
foreach($html->find('a') as $element)
      echo $element->href . '<br>';
```

```
// Create DOM from string
$html = str_get_html('<div id="hello">Hello</div><div id="world">World</div>');

$html->find('div', 1)->class = 'bar';

$html->find('div[id=hello]', 0)->innertext = 'foo';

echo $html; // Output: <div id="hello">foo</div><div id="world" class="bar">World</div>
```

```
// Dump contents (without tags) from HTML
echo file_get_html('http://www.google.com/')->plaintext;
```

```
// Create DOM from URL
$html = file_get_html('http://slashdot.org/');

// Find all article blocks
foreach($html->find('div.article') as $article) {
    $item['title']    = $article->find('div.title', 0)->plaintext;
    $item['intro']    = $article->find('div.intro', 0)->plaintext;
    $item['details'] = $article->find('div.details', 0)->plaintext;
    $articles[] = $item;
}

print_r($articles);
```

How to create HTML DOM object?

Top

```
// Create a DOM object from a string
$html = str_get_html('<html><body>Hello!</body></html>');

// Create a DOM object from a URL
$html = file_get_html('http://www.google.com/');

// Create a DOM object from a HTML file
$html = file_get_html('test.htm');
```

```
// Create a DOM object
$html = new simple_html_dom();

// Load HTML from a string
$html->load('<html><body>Hello!</body></html>');

// Load HTML from a URL
$html->load_file('http://www.google.com/');

// Load HTML from a HTML file
$html->load_file('test.htm');
```

How to find HTML elements?

Top

```
// Find all anchors, returns a array of element objects
$ret = $html->find('a');
```

```
// Find (N)th anchor, returns element object or null if not found (zero based)
$ret = $html->find('a', 0);

// Find lastest anchor, returns element object or null if not found (zero based)
$ret = $html->find('a', -1);

// Find all <div> with the id attribute
$ret = $html->find('div[id]');

// Find all <div> which attribute id=foo
$ret = $html->find('div[id=foo]');
```

```
// Find all element which id=foo
$ret = $html->find('#foo');

// Find all element which class=foo
$ret = $html->find('.foo');

// Find all element has attribute id
$ret = $html->find('*[id]');

// Find all anchors and images
$ret = $html->find('a, img');

// Find all anchors and images with the "title" attribute
$ret = $html->find('a[title], img[title]');
```

Supports these operators in attribute selectors:

Filter	Description
[attribute]	Matches elements that **have** the specified attribute.
[!attribute]	Matches elements that **don't have** the specified attribute.
[attribute=value]	Matches elements that have the specified attribute with a **certain value**.
[attribute!=value]	Matches elements that **don't have** the specified attribute with a certain value.
[attribute^=value]	Matches elements that have the specified attribute and it **starts** with a certain value.
[attribute$=value]	Matches elements that have the specified attribute and it **ends** with a certain value.
[attribute*=value]	Matches elements that have the specified attribute and it **contains** a certain value.

```
// Find all <li> in <ul>
$es = $html->find('ul li');

// Find Nested <div> tags
$es = $html->find('div div div');

// Find all <td> in <table> which class=hello
$es = $html->find('table.hello td');

// Find all td tags with attribite align=center in table tags
$es = $html->find(''table td[align=center]');
```

```
// Find all text blocks
$es = $html->find('text');

// Find all comment (<!--...-->) blocks
$es = $html->find('comment');
```

```
// Find all <li> in <ul>
foreach($html->find('ul') as $ul)
{
    foreach($ul->find('li') as $li)
    {
        // do something...
    }
}

// Find first <li> in first <ul>
$e = $html->find('ul', 0)->find('li', 0);
```

How to access the HTML element's attributes?

Top

```
// Get a attribute ( If the attribute is non-value attribute (eg. checked, selected...), it will returns true or false)
$value = $e->href;

// Set a attribute(If the attribute is non-value attribute (eg. checked, selected...), set it's value as true or false)
$e->href = 'my link';

// Remove a attribute, set it's value as null!
$e->href = null;

// Determine whether a attribute exist?
if(isset($e->href))
```

```
    echo 'href exist!';
```

```
// Example
$html = str_get_html("<div>foo <b>bar</b></div>");
$e = $html->find("div", 0);

echo $e->tag; // Returns: " div"
echo $e->outertext; // Returns: " <div>foo <b>bar</b></div>"
echo $e->innertext; // Returns: " foo <b>bar</b>"
echo $e->plaintext; // Returns: " foo bar"
```

Attribute Name	Usage
$e->**tag**	Read or write the **tag name** of element.
$e->**outertext**	Read or write the **outer HTML text** of element.
$e->**innertext**	Read or write the **inner HTML text** of element.
$e->**plaintext**	Read or write the **plain text** of element.

```
// Extract contents from HTML
echo $html->plaintext;

// Wrap a element
$e->outertext = '<div class="wrap">' . $e->outertext . '<div>';

// Remove a element, set it's outertext as an empty string
$e->outertext = '';

// Append a element
$e->outertext = $e->outertext . '<div>foo<div>';

// Insert a element
$e->outertext = '<div>foo<div>' . $e->outertext;
```

How to traverse the DOM tree?

Top

```
// If you are not so familiar with HTML DOM, check this link to learn more...

// Example
echo $html->find("#div1", 0)->children(1)->children(1)->children(2)->id;
// or
echo $html->getElementById("div1")->childNodes(1)->childNodes(1)->childNodes(2)->getAttribute('id');
```

You can also call methods with **Camel naming convertions**.

Method	Description
mixed $e->**children** ([int $index])	Returns the Nth **child object** if **index** is set, otherwise return an **array of children**.
element $e->**parent** ()	Returns the **parent** of element.
element $e->**first_child** ()	Returns the **first child** of element, or **null** if not found.
element $e->**last_child** ()	Returns the **last child** of element, or **null** if not found.
element $e->**next_sibling** ()	Returns the **next sibling** of element, or **null** if not found.
element $e->**prev_sibling** ()	Returns the **previous sibling** of element, or **null** if not found.

How to dump contents of DOM object?

Top

```
// Dumps the internal DOM tree back into string
$str = $html->save();

// Dumps the internal DOM tree back into a file
$html->save('result.htm');
```

```
// Dumps the internal DOM tree back into string
$str = $html;

// Print it!
echo $html;
```

How to customize the parsing behavior?

Top

```
// Write a function with parameter "$element"
function my_callback($element) {
    // Hide all <b> tags
    if ($element->tag=='b')
        $element->outertext = '';
}

// Register the callback function with it's function name
$html->set_callback('my_callback');

// Callback function will be invoked while dumping
echo $html;
```

Author: S.C. Chen (me578022@gmail.com)
Original idea is from Jose Solorzano's HTML Parser for PHP 4.
Contributions by: Contributions by: Yousuke Kumakura, Vadim Voituk, Antcs