

PHP DOMDocument Sınıfı Kullanımı

(PHP 4 >= 4.1.0) ile gelen **DOMDocument** sınıfı ile bir html sayfasını **parsing** etmek için veya bir **HTML** string'ini parsing etmek için şu kodları kullanmak gereklidir:

```
1 //UTF-8 sayesinde karakter problemi ortadan kalkar.
2 $dom = new DOMDocument('1.0','UTF-8');
3 $encodingHint = '<meta http-equiv="Content-Type" content="text/html; charset=utf-8">';
4
5 $dom->substituteEntities = TRUE;
6 //loadHTML metodu parametre olarak bir string veya html dosya adi alır. Buradaki $content bir HTML string'i
7 $dom->loadHTML($encodingHint.$content);
8 $domx = new DOMXPath($dom);
9
10 //Bu kodta ise benimCssSınıfim isimli css sınıfına göre parsing yapılacağı belirtilmektedir.
11 $entries = $domx->query("//*[contains(concat(' ', normalize-space(@class), ' '), ' benimCssSınıfim ')]");
12 $satirlar=array();
13 foreach ($entries as $entry) {
14     $satirlar[] = '<' . $entry->tagName . '>' . $entry->nodeValue . '<!--' . $entry--->tagName . '>';
15 }
```

Eğer bir css sınıfına göre parsing değil de, bir elemente göre parsing yapmak istiyorsak **DOMXPath** sınıfının **evaluate()** metodu içerisinde **HTML** elementini yazmamız gerekecektir.

```
1 $parsingEdilecekHTMLKodu="<p>Phasellus blandit enim eget odio euismod eu dictum quam scelerisque.</p>".
2 "<p>Sed ut diam nisi.</p><p>Ut vestibulum volutpat luctus.</p>";
3 $dom = new DOMDocument();
4 $dom->loadHTML($parsingEdilecekHTMLKodu);
5 $domx = new DOMXPath($dom);
6 $entries = $domx->evaluate("//p");
7 $arr = array();
8 foreach ($entries as $entry) {
9     $arr[] = '<' . $entry->tagName . '>' . $entry->nodeValue . '<!--' . $entry--->tagName . '>';
10 }
11 print_r($arr);
```

Bu kod çalıştığı zaman ekrana şu şekilde bir array yazılır:

```
1 Array(
2     [0] = '<p>Phasellus blandit enim eget odio euismod eu dictum quam scelerisque.</p>';
3     [1] = '<p>Sed ut diam nisi. Ut vestibulum volutpat luctus.</p>';
4     [2] = '<p>Ut vestibulum volutpat luctus.</p>';
5 )
```