



At klippe en streng over på det mest hensigtsmæssige sted

Formålet med denne artikel er at kaste lidt lys over, hvordan man klipper en streng over på det mest hensigtsmæssige sted.

Skrevet den **03. Feb 2009** af **jensgram** | kategorien **Programmering / PHP** | ★★★★★

[Hvorfor denne artikel?]

Et spørgsmål, der går igen i PHP-kategorien er "Hvordan finder jeg det første af en tekst?" osv. Det er derfor meningen, at denne artikel - med den dertilhørende funktion - skal forklare princippet bag.

[Inden vi kommer for godt i gang!]

Det havde været fedt, hvis Eksperten ville "highlighte" PHP, men det må vi tænke os til ;) Desuden er koden pænere med indrykninger, men de forsvinder også.

[Hurtigt i gang]

En hurtig måde at klippe en streng er ved brug af substr(). Man kan dog risikere at klippe ord over, så lad os lave en funktion, der først tjekker, om den ikke kan klippe ved et punktum:

```
<?php

function klip($tekst, $laengde = 50, $tolerance = 0)
{
    $punktum    = strpos(substr($tekst, $laengde - $tolerance, 2 * $tolerance),
    '.'); // position af punktum
    $mere       = '...'; // punktummer hvis mere tekst

    if (strlen($tekst) <= $laengde + $tolerance) {
        $slut_pos = strlen($tekst); // slutposition = slut på tekst, da denne
er kortere end $laengde og $tolerance
        $mere     = ''; // ikke andet
    } elseif ($punktum !== false) {
        $slut_pos = $laengde - $tolerance + $punktum; // slutposition =
punktum
    } else {
        $slut_pos = $laengde; // slutposition $laengde (ikke noget punktum
indenfor $tolerance)
    }

    return substr($tekst, 0, $slut_pos) . $mere; // udskriv del + evt.
punktummer
}

?>
```

Nu har vi en funktion der tager 1, 2 eller 3 argumenter:

\$tekst: Den tekst der skal klippes.

\$laengde: Længden af den tekst du vil have tilbage (50 hvis udeladt).

\$tolerance: Dette angiver, hvor tæt på \$laengde din tekst skal være. Som standard er \$tolerance sat til 0, hvilket vil sige, at din streng bliver klippet på \$laengde - uanset om der er et punktum eller ej. Sætter du derimod \$tolerance til eksempelvis 10 vil funktionen lede efter et punktum mellem \$laengde - 10 og \$laengde + 10!

Variablen \$punktum indeholder et indeks for det første punktum mellem \$laengde - \$tolerance og \$laengde + \$tolerance!

\$mere indeholder det vi vil tilføje, hvis der bliver skåret noget af strengen. Tre punktummer er standard, men du kan og benytte HTML mv. Eg. \$mere = '<i >[læs mere...]</i >'

If() bruges til flere ting:

- 1) Først tjekkes, om \$tekst er kortere end \$laengde + \$tolerance. Er dette tilfældet så sættes \$slut_pos til \$tekst's længde. Desuden fjernes de tre punktummer, der skal indikere, at der er mere tekst (det er der jo ikke). Opfyldes denne betingelse ikke:
- 2) Vi ser nu, om der overhovedet var et punktum. I bekræftende fald sættes \$slut_pos til \$laengde - \$tolerance (vores udgangspunkt) PLUS positionen at det første punktum (\$punktum). Ellers:
- 3) Hvis intet af det ovenstående kunne lade sig gøre klippes der bare på \$laengde uden at skelne til \$tolerance.

[Vi tester]

Lad os teste lidt på ovenstående:

```
<?php
print klip('Jeg spiser tit ILD!'); // => 'Jeg spiser tit ILD!'
print klip('Jeg spiser tit ILD!', 10); // => 'Jeg spiser...'
print klip('Jeg er træt af ILD. Det smager ikke som i gamle dage.', 25, 10);
// => 'Jeg er træt af ILD...'
?>
```

Forklaring:

- 1) 'Jeg spiser tit ILD!' er kortere end 50 tegn, hvorfor hele strengen returneres.
- 2) Vi vælger at klippe ved 10 tegn (uden tolerance), så de første 10 tegn returneres ('Jeg spiser...' - strengen er jo længere, så der kommer ... på).
- 3) Vi har en streng med flere punktummer i. Vi vil have en streng på ca. 25 tegn (15 - 35). Dette er: 'ILD. Det smager ikke'. Der er et punktum, så \$slut_pos sættes og strengen returneres med tre punktummer til sidst.

[Del også ved mellemrum!]

Ønsker man, at funktionen skal dele ved mellemrum, hvis den ikke lige finder et egnet punktum, så kan man benytte:

```
<?php

function klip($tekst, $laengde = 50, $tolerance = 0)
{
    $punktum    = strpos(substr($tekst, $laengde - $tolerance, 2 * $tolerance),
    '.'); // position af punktum
    $mellemrum  = strpos(substr($tekst, $laengde - $tolerance, 2 * $tolerance),
    ' '); // position af mellemrum
    $mere       = '...'; // punktummer hvis mere tekst

    if (strlen($tekst) <= $laengde + $tolerance) {
        $slut_pos = strlen($tekst); // slutposition = slut på tekst, da denne
er kortere end $laengde og $tolerance
        $mere     = ''; // ikke andet
    } elseif ($punktum !== false) {
        $slut_pos = $laengde - $tolerance + $punktum; // slutposition =
punktum
    } elseif ($mellemrum !== false) {
        $slut_pos = $laengde - $tolerance + $mellemrum; // slutposition =
mellemrum
    } else {
        $slut_pos = $laengde; // slutposition $laengde (ikke noget at gøre -
skær ord over)
    }

    return substr($tekst, 0, $slut_pos) . $mere; // udskriv del + evt.
punktummer
}

?>
```

Vi har blot indsat endnu en betingelse, der testes hvis der ikke findes et punktum. Eller er funktionen som før:

```
<?php

print klip('Jeg er træt af ILD. Det smager ikke som i gamle dage.', 25, 10);
// => 'Jeg er træt at ILD...'

print klip('Jeg er træt af ILD.', 10, 2); // => 'Jeg er træt...'

?>
```

Eksempel nummer to viser hvordan der ikke blev fundet et punktum (mellem tegn 8 og 12). I stedet fandtes der et mellemrum, som så blev brugt til at dele.

[Udvidelse]

I princippet kan du selv tilføje flere tjeks som den med mellemrum. Vil du dele ved flere tegn vil jeg dog foreslå:

```
<?php

function klip($tekst, $laengde = 50, $tolerance = 0)
{
    $tegn = '., :;-_'; // De tegn du gerne vil klippe ved
    $mere = '...'; // Punktummer hvis mere tekst

    if (strlen($tekst) <= $laengde + $tolerance) {
        $slut_pos = strlen($tekst); // Slutposition = slut på tekst, da denne
er kortere end $laengde og $tolerance
        $mere = ''; // Ikke andet
    } else {
        $slut_pos = $laengde; // Slutposition = længde, hvis der ikke findes
et af vores tegn!
        $plads = array(); // Array, der indeholder pladser for de
forskellige tegn
        $del = substr($tekst, $laengde - $tolerance, 2 * $tolerance); //
streng indenfor $tolerance
        for ($i = 0; $i < strlen($tegn); $i++) {
            $plads[$i] = strpos($del, $tegn[$i]); // Position af det aktuelle
tegn
            if ($plads[$i] !== false) {
                $slut_pos = $laengde - $tolerance + $plads[$i]; //
Slutposition = position af tegn
                break; // Afslut løkke
            }
        }
    }

    return substr($tekst, 0, $slut_pos) . $mere; // udskriv del + evt.
punktummer
}

?>
```

Her lader du \$tegn indeholde de tegn, du vil dele på.
Desuden kræver Else-delen vist lidt forklaring:

1) Vi starter med at sætte slut-positionen til \$laengde, da vi jo endnu ikke ved, om der findes nogle af de

tegn vi vil klippe ved.

- 2) \$plads er et array, der skal indeholde placeringen er første tilfælde af hvert tegn i \$tegn.
- 3) \$del laver vi blot, da vi så tit skal bruge indholdet indenfor tolerancen.
- 4) Vi laver en løkke der kører gennem hvert tegn i \$tegn.
- 5) Positionen af det aktuelle tegn i \$tegn i strengen \$tekst findes
- 6) Hvis tegnet findes...
- 7) Vi sætter slut-positionen til tegnets plads.
- 8) Løkken afbrydes da der eksempelvis ikke er nogen grund til at lede efter mellemrum hvis vi har fundet et punktum at dele ved.

[Afslutning]

Dette skulle gerne have kastet lidt lys over, hvordan du laver en "appetitvækker" til dine lange tekster. Ovenstående funktion har jeg længe brugt og har ikke haft problemer - blot glæde af den.

Venlig hilsen - og god fornøjelse
- Jens Gram

PS: Jeg ved godt, at det er lidt sent, men vil bare lige gøre opmærksom på, at en formateret version af denne artikel kan findes på <http://www.jensgram.dk/web/e-artikler/?aid=45>

Kommentar af coldray d. 28. Jan 2004 | 1

Kommentar af razmuz d. 21. Jul 2004 | 2

Artiklen er jo ikke til at læse!!!!

Kommentar af mysitesolution d. 27. Jun 2005 | 3

hmm griner lidt ad denne linie
`print klip('Jeg spiser tit ILD!', 10); // => 'Jeg spiser...'`

man har valgt at der max må være 10 tegn, men alligevel spytter den 13 tegn ud :D

hvad nu hvis teksten havde været "det spiser i", vi kan alle se den er mere end 10 tegn, men hov, så kommer der til at stå "det spiser..." hvilket er 13 tegn, hvor hele linien ville have fyldt 12 :D

Kommentar af fjensen111 d. 20. Jan 2004 | 4

Kommentar af sandbox d. 15. Jan 2004 | 5

Synd at kodeformateringen går i ged.