



SESSION med PHP

En let begynderguide som kan føre dig ind i en verden af muligheder

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

Artiklen er delt op i følgende afsnit:

- a) **Hvad er session kort fortalt**
- b) **Hvilke funktioner skal jeg bruge**
- c) **Eksempler**
- d) **Hvad kan du ellers bruge sessions til**
- e) **Lidt fra skribenten**

a) **Hvad er session kort fortalt**

Session er et utroligt stærkt værktøj som du som kommende super web-programmør kan gøre stor nytte af.

En session er, rent billedligt fortalt, en lille kasse(array) som der følger det browservindue hvori du åbner session og følger med hele tiden uanset hvilken side du befinder dig på.

En session gemmes i et array som du selv bestemmer hvad hedder, der kan oprettes utallige arrays i din session. Der kan laves et array som indeholder 1 eller flere værdier.

Du kan når som helst hente værdier fra din session og skrive nye værdier til den og rette i andre af dem. Det kunne lyde som et eller andet overvildt besværligt noget, men det er det langt fra. Lidt tålmodighed og gåpåmod, så du oppe og i gang med sessions i løbet af 30 min.

b) **Hvilke funktioner skal jeg bruge**

For at vi kan arbejde med session er det vigtigt, at du stifter bekendtskab med 3 funktionskald som er basis i brug af session. Jeg stiller dem op og fortæller kort hvilken funktion de har.

```
session_start();
```

Denne funktion starter vores session eller den lille kasse(array) der blev omtalt tidligere. Nu er vores session eller kasse tom og klar til at vi bruger den.

```
unset();
```

Denne funktion bruges til at 'nulstille' værdier(ne) i vores session, den kan 'nulstille' en eller flere værdier. Det vender vi tilbage til senere i nogle af eksemplerne.

```
session_destroy();
```

Denne funktion sletter ubetinget alle sessions, altså når denne er kaldt - Så er alt tilbage hvor vi startede

c) **Eksempler**

Eks. 1 Start af session

```
<?php // filnavn: eks1.php
```

```
session_start(); // Nu er vores session åben og klar.
```

```
echo "Session er startet";
```

```
?>
```

session_start(); kaldes som det første efter <?php så er man sikker på at den er kaldt og kører.

Eks. 2 Vi sætter en værdi i vores session

```
<?php // filnavn: eks2.php
```

```
session_start(); // Nu er vores session åben og klar.
```

```
$_SESSION['mitnavn'] = "McCookie";
```

```
echo "Mit navn er: "; echo $_SESSION['mitnavn'];
```

```
?>
```

Nu har vi lavet en session med en værdi under 'mitnavn' og den kan vi bruge når som helst det passer os. Som tidligere nævnt kan sessions bruges på alle sider så længe vi holder os til det browservindue vi startede i. \$_SESSION er en servervariabel, så derfor skal der altid skrives \$_ foran SESSION

Eks. 3 Vi henter en værdi fra vores session

```
<?php // filnavn: eks3.php
```

```
// Nu er vi på en ny side men kan sagtens bruge vores værdi fra sidste eks.
```

```
session_start(); // Nu er vores session åben og klar.
```

```
echo "Mit navn er: "; echo $_SESSION['mitnavn'];
```

```
?>
```

Forhåbentligt er det til at se hvor let det er at sætte en session

Eks. 4 Vi sletter en værdi i vores session

```
<?php // filnavn: eks4.php
```

```
session_start(); // Nu er vores session åben og klar.
```

```
unset($_SESSION['mitnavn']);
```

```
echo "Mit navn er: "; echo $_SESSION['mitnavn'];
```

```
?>
```

Nu er vores session tom og outputtet vil begrænse sig til: Mit navn er:

Eks. 5 Vi sætter værdier i vores session på en avanceret måde

```
<?php // filnavn: eks5.php
```

```
session_start(); // Nu er vores session åben og klar.
```

```
$_SESSION['userdata']['fornavn'] = "Henrik";
$_SESSION['userdata']['efternavn'] = "Jensen";
$_SESSION['userdata']['email'] = "henrik@jensen.dk";
// Nu har vi 3 værdier i den kasse der hedder userdata.
```

```
$_SESSION['login']['tid'] = "12:12:00";
$_SESSION['login']['sted'] = "www.eksperten.dk";
$_SESSION['login']['hvorfor'] = "Fordi det er verdens bedste site";
// Nu har vi 3 værdier i den kasse der hedder login.
```

```
echo $_SESSION['userdata']['fornavn'];
echo $_SESSION['userdata']['efternavn'];
echo $_SESSION['userdata']['email'];
echo $_SESSION['login']['tid'];
echo $_SESSION['login']['sted'];
echo $_SESSION['login']['hvorfor'];
```

```
// Der skal lige skrives en ekstra parameter for se hvad der er i kassen
// Nu er det smarte at hvis vi ønsker at slette alle userdata så gøre det så'n:
```

```
unset($_SESSION['userdata']);
```

```
// Vi har stadig værdierne liggende i vores kasse(array), selvom vi slettede de andre.
?>
```

Dette eksempel skulle vise dig hvor smart det kan være at samle værdier i undergrupper, så det ikke bliver for uoverskueligt

Eks. 6 Vi sætter værdier i vores session på en avanceret måde og sletter dem alle

```
<?php // filnavn: eks6.php
```

```
session_start(); // Nu er vores session åben og klar.
```

```
$_SESSION['userdata']['fornavn'] = "Henrik";
$_SESSION['userdata']['efternavn'] = "Jensen";
$_SESSION['userdata']['email'] = "henrik@jensen.dk";
// Nu har vi 3 værdier i den kasse der hedder userdata.
```

```
$_SESSION['login']['tid'] = "12:12:00";
$_SESSION['login']['sted'] = "www.eksperten.dk";
$_SESSION['login']['hvorfor'] = "Fordi det er verdens bedste site";
// Nu har vi 3 værdier i den kasse der hedder login.
```

```
echo $_SESSION['userdata']['fornavn'];
echo $_SESSION['userdata']['efternavn'];
echo $_SESSION['userdata']['email'];
echo $_SESSION['login']['tid'];
echo $_SESSION['login']['sted'];
echo $_SESSION['login']['hvorfor'];
// Vi udskriver alle vores data
```

```
// Nu er vi blevet trætte af alle vores data og ønsker at slette det hele.
```

```
session_destroy();
```

```
// Så er alt slettet og pist borte og kan ikke findes mere  
?>
```

Dette eksempel skulle vise dig hvor let det er alt slette alle informationer i din session

d) **Hvad kan du ellers bruge sessions til**

Du kan anvende session til utallige formål og gemme alle mulige tænkelige værdier.

Du kunne anvende session til at gemmes brugerens navn og tale direkte til brugeren fra din hjemmeside.

Du kunne bruge session til et loginsystem og styre hvornår brugeren er logget ind og hvad denne har adgang til

Du kunne gemme værdier fra diverse formularer mv.

You name it and you uses it...

e) **Lidt fra skribenten**

Jeg håber at disse ovenstående <cut and paste eksempler> har givet dig en god start til at arbejde med sessions. Jeg har gjort et forsøg på at begrænse den tekniske forklaring og vise virkemåden gennem eksempler med kommentarer.

Artiklen er som tidligere nævnt primært henvendt til dem, der ikke har arbejdet med sessions før og kunne tænke sig noget basislære omkring emnet.

Dette er den første artikel jeg ever har skrevet i forbindelse med et IT relateret emne. Det har været spændende og lærerigt for mig at skrive den. Så jeg håber at den målgruppe jeg sigtede efter og var den som jeg ramte :)

Jeg har samlet min inspiration fra mit eget arbejde omkring et loginsystem, hvori session er en fundamental del.

Der er samlet viden her fra eksperten.dk og diverse hjemmesider på nettet både danske og udenlandske.

Skulle der falde lidt god karma af til mig, så ville jeg da forsøge at lave endnu et par artikler :o)

- Opdateret: unset istedet for SESSION_unset

God arbejdslyst
Henning Jeppesen

Kommentar af cf560 d. 13. Jan 2008 | 1

Kommentar af gider_ikke_mere d. 02. Sep 2006 | 2

En artikel der kaldes "SESSION med PHP", bør afdække alt der har med sessions at gøre. Hvad f.eks med levetiden for en SESSION. Der bør følges op med en ny artikel, eller en udvidelse af denne! Der står heller ikke noget om at session_start() skal kaldes før der outputttes noget på skærmen!

Kommentar af per-d d. 25. Jul 2006 | 3

En middel artikel der bestemt er læseværdig, men vil kun give den en middel karakter da du glemmer at

nævne faldgruber. Den mest hyppige fejl med session er hvis man sender noget til browseren før man starter en session. Den burde du have med da stort set alle nybegyndere falder over den fejl.

Kommentar af denniskragh d. 04. Aug 2006 | 4

Meget overskuelig

Kommentar af epimp d. 27. Jul 2006 | 5

Du lærer at omgå sessions. Og det er det. Generelt for overfladisk og ukritisk. Hvilke sikkerhedsrisici eksisterer fx?

Kommentar af sbmoller d. 08. Dec 2007 | 6

Kommentar af geggeri d. 20. Aug 2006 | 7

Rigtig god artikel til nybegyndere! :D

Kommentar af hombre d. 04. Nov 2006 | 8

Kommentar af thegimp d. 31. Aug 2007 | 9