

Back End-utveckling i PHP

50 YHP

iDag

Att spara data från browsern och serverside

iMorgon

Olika generella dataformat som JSON och XML

Introduktion till PHP ?

DATAFORMAT & LAGRING

Torsdag 25 Januari 2018

dagens förberedelse

- skapa en ny mapp med en kopia av gårdagens innehåll, eller fortsatt från samma map.
- Öppna mappen i **VS CODE** och **en terminal**
- Kom ihåg att i terminalen kan man använda **cd** följt av en **sökväg** för att navigera till rätt mapp.

"minne"

- Kan variera mellan bråkdelar av sekunder och flera år.
- *Lagrings*-minne. Hårddiskar, SSD, Minneskort, CD-Skivor, ROM.
- *Arbetes*-minne. RAM (även Swap minne)

Arbetsminne

- Skrivs till och från *väldigt väldigt* snabbt
- Allt som syns på skärmen finns i arbetsminnet
- Variabler sparas i arbetsminnet.
- Direkt när ingen variabel (*egentligen "pekare"*) finns kvar för ett värde så försvinner det ur arbetsminnet.

Lagringsminne

- Kan lagra information även utan en strömkälla
- Behöver flyttas till arbetsminnet för att användas

Minnesåtkomst från webbläsaren

- Skyddad och isolerat system.
- Ingen åtkomst till filsystemet
- **SessionStorage** och **LocalStorage**

Minnesåtkomst från webbläsaren

- Local- och SessionStorage är båda `{ key: value }` objekt
- SessionStorage är till för det nuvarande besöket
- LocalStorage är till för *inställningar* för den här datorn
- **Value** måste vara en sträng `""`
- Objekt och värden kan konverteras till strängar med `JSON.stringify('')`

övning

- På index.html sidan, lägg till en `<a>` länk och en text som ser ut något så här:
"du har besökt sidan totalt X gånger, varav Y var under denna sessionstid"
- När man klickar på länken ska man navigera till samma sida igen.
- När sidan laddas ska siffrorna X och Y öka varje gång med hjälp av Session- och LocalStorage
- **Använd inte onClick!**
- Tänk på att sätta X och Y till 0 om inget värde finns i storage.



minne från servern

Minnesåtkomst från en server

- *Vad är en server nu igen?*
- I regel inte ett isolerat system - *varför är webbläsaren ett isolerat system?*
- Full åtkomst till filsystemet
- **I nodeJS: `FileSystem ('fs')`**

Filer

- På en dator körs (oftast) ett filsystem
- Filsystem ansvar för att hålla koll på filerna på lagringsmediumet
- Kända Filsystem: FAT, NTFS, HFS+...
- Operativsystemet har ett gemensamt gränsnit för att jobba med olika filsystem, via en drivrutin
- **fs** i NodeJS jobbar mot operativsystemets drivrutin

Filer



operativsystemet



FAT drivrutin



FAT



övning

- Gör en ny fil kallad **editor.js**
- Installera paketet "**cli-prompt**"
- När editor.js körs, fråga användaren efter en text rad och spara den direkt till en fil som heter '**latestInput.txt**' och sparas till samma mapp.
- **Funktionen** som frågar användaren ska vara i en klass som sen kan användas från ett annat skript.
- **Tips:** importera 'fs' och kolla vad det finns för metoder för att skriva filer där
- 👉 När man skriver *sökvägar* så refererar . (en punkt) till den nuvarande mappen, och '..' till mappen ett steg upp. exempelvis *./index.html* betyder "filen index.html" i samma mapp som terminalen står i"

övning

- Ändra **parser.js** och **editor.js** så att klasserna exporteras, med hjälp av `'module.exports ='`
- Gör en ny fil kallad **codeEditor.js**
- När **codeEditor.js** körs:
 - Ska editor importeras och skapa en ny fil
 - Parser ska därefter läsa den filen och direkt efteråt parsas och skiva ut innehållet.

Dvd. om man skriver in "detta är <röd text>" så ska det sparas i en fil, och sen direkt skrivas ut igen som "detta är **röd text**"
- **Tips:** `module` är en global variabel som finns i nodejs, som `window` i browsern. `module.exports` är det som exporteras när man sen importerar filen med **`require()`**

Generella dataformat

- Skrivs som ren text (jämfört med bilder, ljud mm)
- Har en bestämd syntax, som kod, som avgör innehållets typ.
- Har inget bestämt användningsområde.
- **Exempel: XML, JSON, YAML, MARKDOWN**
- HTML, CSS?

Generella dataformat

- **Varför?** För att rena textfiler blir svårlästa.
- För att införa **metadata**
- **Enkelt att läsa av både människor och datorer**
- Ett bra mellanalternativ för mindre mängd information, innan man behöver en databas.

Generella dataformat

- **Varför inte?** Vid för stor mängd blir det oftast klumpigt (som med stora hemsidor)
- Databaser är snabbare
- Binärfiler (mp3, jpg etc) är mycket snabbare att parsa för en dator.

Tack!