

Laboration 7

Cookies

– övningar/uppgifter

1M322 Webbteknik 2, 7,5hp

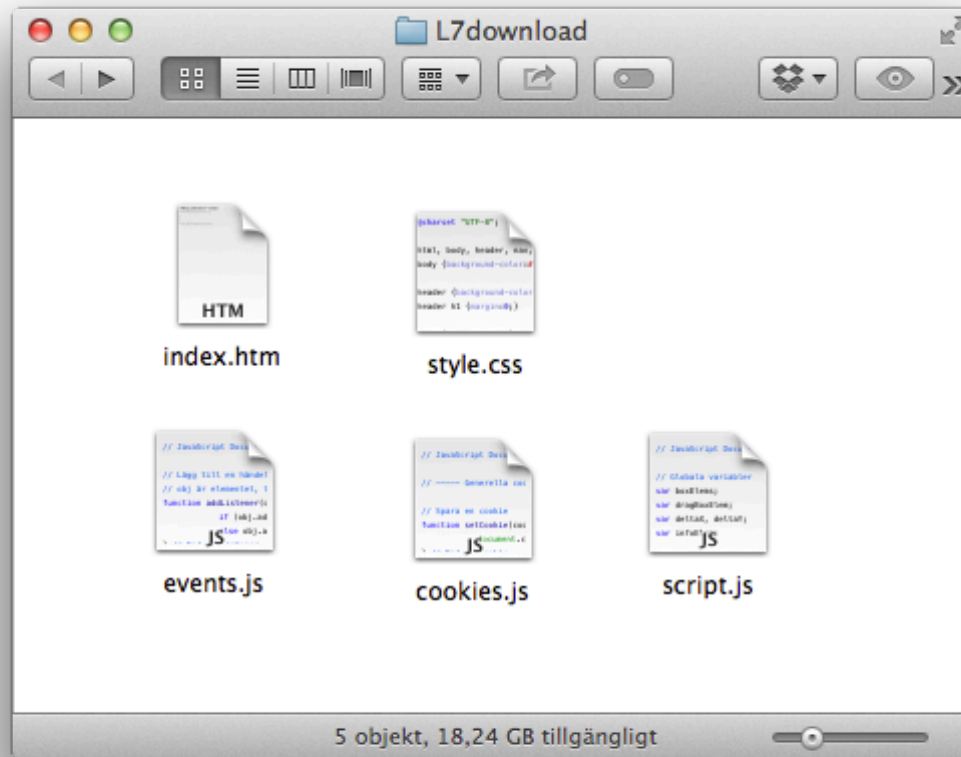
Medieteknik

1. Ladda ner arbetsdokument

Till övningarna i denna laboration finns det ett antal filer som du kan ladda ner i en zip-fil.
Länk till zip-filen finns på laborationens webbsida.

Då du packat upp zip-filen, får du en mapp med fem filer.

Det är i filen script.js som du ska skriva koden i övningarna. Du använder då också funktioner i de andra script-filerna.



2. Orientera dig i de nedladdade filerna

I de nedladdade filerna finns ett program där man kan dra fem olika boxar till olika positioner. Detta är ytterligare ett exempel på drag-and-drop, men det är inte det som det handlar om nu. Titta dock gärna ändå på koden för hur boxarna hanteras och hur koordinaterna för top och left beräknas. Men det kan du göra efter denna laboration.

Det som det nu handlar om är att du ska lägga till kod för att spara boxarnas position, så att de har kvar sina positioner, då man kommer tillbaks till webbsidan.

Tilllägg i övningarna

De tilllägg som ska göras i övningarna är:

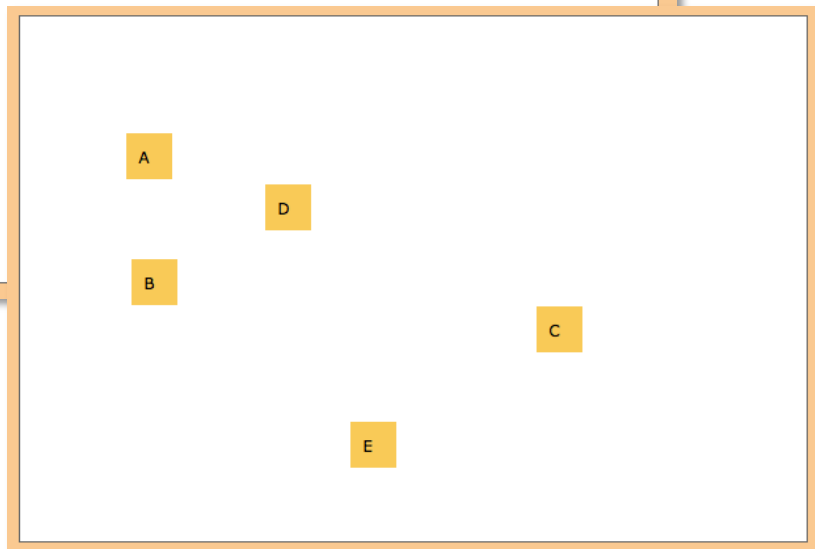
- Då man lämnar sidan, ska top och left för alla boxar sparas i en cookie.
- Då man kommer till sidan, ska cookien läsas in och top och left för boxarna ska återställas till de värden som finns i cookien.

Filen *cookies.js*

I filen *cookies.js* finns det ett antal funktioner för att spara och läsa en cookie. Detta är samma fil och funktioner som finns i exemplen i föreläsning F7. Så om du inte redan satt dig in i hur dessa funktioner fungerar, bör du göra det nu.

1ME322, Laboration 7 - Cookies

Dra de små boxarna till önskad plats inom den vita ytan.



3a. Spara en cookie

Du ska nu skapa funktionen för att spara cookien. Vi tar det i ett antal steg, så att du kan testa och se hur cookien sparas. Vi börjar i detta steg med att skapa en cookie med ett fast värde.

Funktionen *saveBoxCookie*

- Skapa en ny funktion och kalla den *saveBoxCookie*.
 - Den kan ligga var som helst i filen, men lägg den under *init*-funktionen, där det finns några tomma rader i script-filen.
- I funktionen anropar du *setCookie* (som finns i *cookies.js*).
 - Som parametrar skickar du med *"boxpos"* som namn och *"kakan"* som värde.

```
setCookie("boxpos", "kakan");
```

Anrop av *saveBoxCookie*

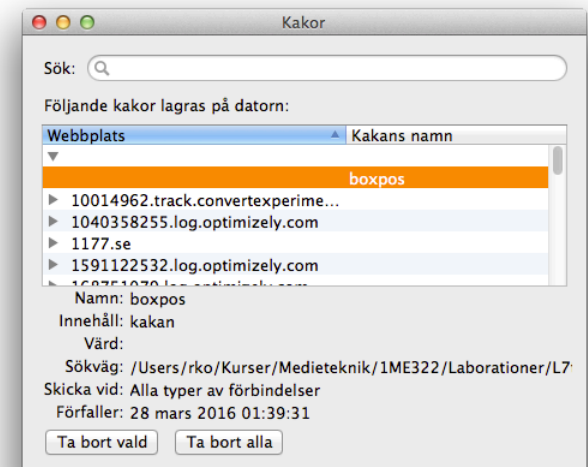
Cookien kan sparas i någon funktion där man gör en förändring av den data som ska sparas eller, om någon sådan funktion inte finns, då man lämnar sidan. I detta fall gäller det sistnämnda. Du ska alltså spara cookien, då man lämnar sidan, dvs då händelsen *"unload"* inträffar.

- Under funktionen *init*, lägger du till en rad med *addListener*, där du som parametrar skickar med händelsen *"unload"* och funktionen *saveBoxCookie*.

Testa

Händelsen *"unload"* inträffar då man går till en annan sida, stänger fönstret med sidan eller då man laddar om sidan.

- För att testa, laddar du om sidan i webbläsaren.
- Ta sedan fram info om cookies och kontrollera att cookien finns och innehåller värdet *"kakan"*.
 - I Firefox väljer du menykommandot *Verktyg → Sidinfo*.
 - I det fönster du då får väljer du fliken *"Säkerhet"*.
 - Där klickar du på knappen *"Visa kakor"*.
 - Du får då ytterligare ett fönster med bl.a. en kolumn kallad *"Webbplats"*. Översta alternativet i den kolumnen ska vara tomt och det är den *"lokala domänen"*, dvs det som gäller då du kör webbsidan lokalt på din dator.
 - Klicka på det alternativet, så att det expanderas och du bör då se cookien *"boxpos"*.
 - Klicka på *"boxpos"*, så ser du dess innehåll. Efter rubriken *"Innehåll"* ska det stå *"kakan"*.
 - Då vet du att du kan spara en cookie.



3b. Spara en cookie

Nu ska du spara boxarnas position i cookien. Du ska alltså för varje box spara värdena för *top* och *left*. Dessa värden ska åtskiljas av ett kommatecken. Det är fem boxar, så det blir fem sådana par av värden. För att åtskilja dem använder du tecknet #.

Cookiens värde kan alltså t.ex. bli *"14,56#398,45#296,356#23,458#285,369"*.

Funktionen *saveBoxCookie*

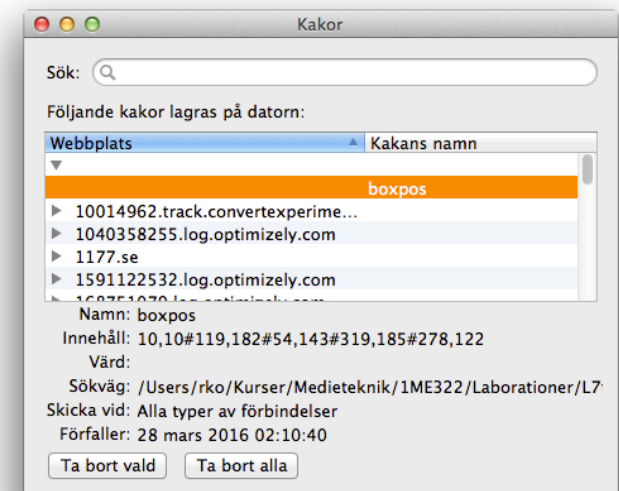
- Deklarera variablerna *i*, *l*, *t* och *cookieValue*, som ska vara en loopvariabel, värden för *top* och *left* samt den textsträng som ska sparas i cookien.
- Börja med att sätta *cookieValue* till en tom sträng.
- Inför sedan en *for*-loop där du går igenom alla boxar, dvs arrayen *boxElems*.
I loopen gör du följande:
 - Ta fram *left* och *top* från *boxElems[i]* och spara i variablerna *l* och *t*.
Använd *parseInt*, för att få bort "px", så att det endast blir de numeriska värdena.
 - Lägg sedan in ett #-tecken och värdena *l* och *t* åtskilt av komma i variabeln *cookieValue*.
 - Använd operatoren += för att lägga till detta till variabelns innehåll.
- Efter loopen tar du bort det första #-tecknet med hjälp av *substring*, se koden i figuren.
- Sista satsen i funktionen är *setCookie*, där du nu byter "*kakan*" till *cookieValue*, utan citattecken.

```
cookieValue += "#" + l + "," + t;
```

```
cookieValue = cookieValue.substring(1);
```

Testa

- Ladda om sidan i webbläsaren, så att du får in det nya programmet.
- Dra boxarna till olika positioner.
- Ladda sedan om sidan i webbläsaren, så "*unload*" inträffar och cookien sparas.
- Ta fram info om cookien (på samma sätt som i föregående övning) och kontrollera att cookien "*boxpos*" nu innehåller en sekvens av siffrvärden.
 - Boxarnas position behålls dock inte, eftersom du ännu inte skrivit kod för det.



4a. Läs in cookien

Nu ska du skriva koden för att läsa in cookien. Även detta tar vi i två steg, så vi börjar med att du läser in cookien och skriver ut den i infofältet under den vita rutan.

Funktionen *getBoxCookie*

- Skapa en ny funktion och kalla den *getBoxCookie*.
- Deklarera variablerna *i*, *cookieValue*, *cookieArr* och *posArr*, som ska vara en loopvariabel, cookiens sträng, värdet uppdelat i en array och varje position uppdelat i en array med *left* och *top*.
- Avläs cookien med funktionen *getCookie* (som finns i filen *cookies.js*) och spara värdet i variabeln *cookieValue*.
- Skriv sedan ut *cookieValue* i infofältet (som det finns en referens till i variabeln *infoElem*).

```
cookieValue = getCookie("boxpos");
```

Funktionen *init*

- Lägg in ett anrop av *getBoxCookie* sist i *init*-funktionen.
 - Funktionen ska alltså anropas då webbsidan blivit inläst.

Testa

- Ladda om webbsidan, så att du dels får in det nya programmet och funktionen för att läsa cookien anropas.
- Cookien läses då in och dess värde skrivs under den stora vita rutan.
- Du vet då att du kan avläsa cookien och få in värdet i programmet.

```
10,10#50,10#90,10#130,10#170,10
```

4b. Läs in cookien

Nu ska du använda cookiens värdesträng och ta ut värdena för *top* och *left* för de olika boxarna. Dessa ska användas för att positionera boxarna.

Funktionen *getBoxCookie*

- Ta bort utskriften av *cookieValue* och lägg istället in en *if*-sats, där du kontrollerar att *cookieValue* är skilt från *null*.
 - Du kontrollerar alltså att cookien finns. Om den inte finns returnerar *getCookie* värdet *null*.
- I *if*-satsen gör du följande:
 - Dela upp de olika positionerna i en array, splitta alltså på skiljetecknet *#*. Spara i variabeln *cookieArr*.
 - Gå igenom *cookieArr* i en *for*-loop.
I *for*-loopen gör du följande:
 - Dela upp positionen i en array, splitta alltså på skiljetecknet komma. Spara i variabeln *posArr*.
 - Sätt *top* och *left* för boxen till värdena i *pos*.
 - Glöm inte att lägga till enheten "px".

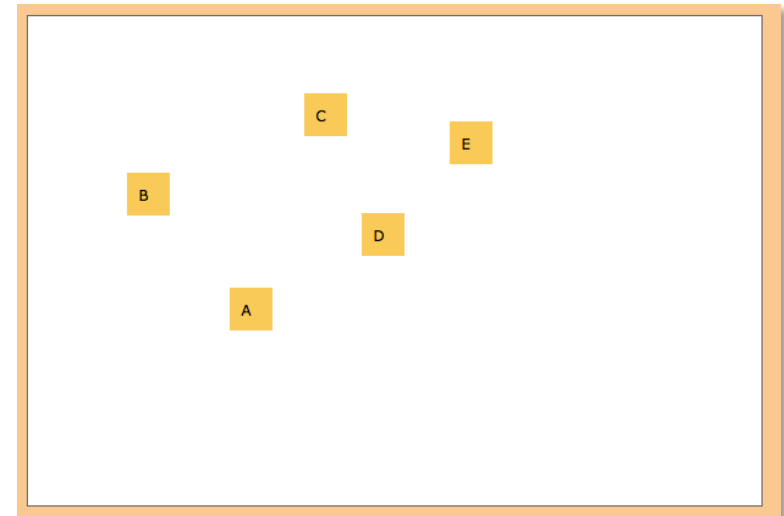
```
cookieArr = cookieValue.split("#");
```

```
posArr = cookieArr[i].split(",");
```

```
boxElems[i].style.left = posArr[0] + "px";  
boxElems[i].style.top = posArr[1] + "px";
```

Testa

- Ladda om sidan i webbläsaren, för att få in det nya programmet.
- Dra boxarna till olika positioner.
- Ladda sedan om sidan igen, för att dels få händelsen "*unload*", så cookien sparas, och dels få in sidan igen så att cookien läses in.
 - Boxarna ska då behålla sina positioner.
- Prova också att stänga sidan och sedan öppna den igen.
 - Boxarnas ska då återställas till de positioner de hade, då du stängde sidan.



5. Klart

Du är nu klar med laborationen. Denna laboration är endast en övning och inte någon direkt del av en uppgift. Du behöver därför inte publicera ditt resultat.

Du bör nu ha sett hur man kan spara en cookie och hur man kan läsa den. Hur man får fram värdet som ska sparas i cookien och vad man gör med värdet, då man läst in cookien, beror på vad det är man sparar. I detta fall handlade det om positioner för boxar, men i något annat fall kan det vara något annat. Grundstrukturen för programmet blir dock densamma.