# Tillsnyggad dokumentation

senast uppdaterad: 2022-12-13

OBS OBS: Många av kod-delarna i detta dokument är inte kollad gällande slarvfel! Gäller speciellt taggar etc som står i löpande text. Rättar sånt efer hand

## Stoyrns Javascript

Stuff som ska stå i Storyns Javascript-ruta (i twine-editorn alltså). Förhoppningsvis kommer jag ihåg att uppdatera detta allt eftersom.

```
var script = document.createElement('script');
script.src = "apsing.js";

function setupGlobals()
{
    var _state = require('state');
    console.log('require state');
    console.log(_state.variables);
    window.statevar = _state.variables;
}

setTimeout(() => {
    console.log("Delayed statevarupdate for 1 second.");
    setupGlobals();
}, 1000)

setupGlobals();
document.head.appendChild(script)
```

**OBS** Det går bra att radera alla rader med console.log! De står kvar just nu i debugg-syfte.

## Inventory

#### Visa inventory

Stoppa in <write-inventory> Header eller footer.

Sätt classen till: "topLeft" för att få lodrät inventory.

Sätt ID till "mainInventory" för att uppdateringsfunktionen ska veta vilket element som är inventory.

Exempel i footer:

```
<inventory class="topLeft">
<write-inventory id="mainInventory"></write-inventory>
</inventory>
```

### **Uppdatera** inventory

På platser där inventoryn behöver uppdateras, använd <update-inventory>
Detta kan göras direkt i twine-texten eller via (print:"<update-inventory>")

Notera: Sätt variabeln före taggen med uppdatera. Annars uppdaterar den med gamla värdet. Exempel:

```
(link: "ta affisch")[(set: $affisch to 1)(print: "<update-inventory>")]
(link: "ta nyckelknippa")[(set: $nyckelknippa to 1)<update-inventory>]
(link: "ficklampa")[(set: $ficklampa to 1)(print: "<write-inventory
type='update'>")]
```

Samtliga dessa 3 alternativ fungerar i nuläget. Säkrast är att använda alternativet längst upp med print och update-inventory taggen.

#### **Under huven**

Jag hade vissa problem med detta eftersom state-variabeln inte uppdaterades omedelbart. Därför uppdaterade den med gamla värde någon millisekund före den fick nya. Det är just nu (13-dec-2022) löst med en timeout på 200 ms i update-funktionen.

Om man använder (print:) macron uppstår ej detta problem eftersom den då kör klart "set" macron före den kör "print" macron.

Men samtliga metoder fungerar alltså för tillfället.

#### Mer under huven

Inventoryn har en egen array (inventoryltemsList) i JS med namn, icon och värde. Den uppdateras från Harlowe-storyns \_state-variabel.

- Varje gång <update-inventory> körs går scriptet igenom window.statevar, hittar entrys som motsvarar alla namn i listan och sätter värdet till vad den har i window.statevar (som alltså motsvarar harlows state).
- Varje gång <a href="write-inventory">write-inventory</a>> körs går scriptet igenom listan och skapar HTML för att visa, eller inte visa, ikoner beroende på om de har värde 0 eller 1. Om write-inventory anropas via update hittar den elementet med id mainInventory. I annat fall använder den sitt eget element för att veta vilken HTML den ska ersätta på sidan.

En snyggare lösning hade varit att, istället för skriva ut HTML igen bara ändra klasser på befintliga element. Gick dock inte eftersom det är svårt att få kontroll över vad Twine-motorn skriver ut här och där.

## Mätarna, score-bars

Sätt in mätarna i headern.

- Omge det hela med en <span id="updatearea"> för att uppdatera-funktionen ska hitta elementet.
- Använd taggen <score-bar type="a"> eller type=b. a= hypnos. b= död. Där scorebar ska ligga.
- Taggen <score-bar type="update"> där uppdatering ska ske. Gärna med en (print:) macro.
- Taggen <a href="huvud">huvud>...</a> runt alltihop för att flytta och låsa scorebar längst upp på sidan. Om man vill det.

#### Exempel: i header:

```
<huvud>\
<span id="updatearea">\
<score-bar type="a">$hypnos</score-bar><score-bar type="b">$död</score-bar>\
</span>\
</huvud>\
```

**NOTERA:** Backslash \ i slutet av varje rad tar bort radbryt. Använder det för att få mer överskådlig kod, allt kan skrivas på samma rad lika gärna.

**NOTERA 2:** Tidigare fanns script-tagg som satte window.hypnos och död i headern. Dessa ska inte behövas längre. *(december 2022)* 

#### Uppdatera mätarna

Där mätarna ska uppdateras, använd: <score-bar type="update">. Använd Print: macron för detta, så den säkert hinner uppdatera i rätt följd. Exempel:

```
(link: "Titta på pendeln")[(set: $hypnos to it + 1)(print: "<score-bar
type='update'>")]

(link: "Set hypnos till 4")[(set: $hypnos to 4)(print: "<score-bar
type='update'>0j oj, du känner dig hypnotiserad!</score-bar>")]

(link: "Minska död")[(set: $död to it-1)(print: "<score-bar type='update'>Ett steg
närmare döden!</score-bar>")]
```

#### **Under huven**

Score-bar använder css-variabler som uppdateras baserat på window.statevar. Skriver ned mer här senare, om jag ids.

Java-scriptet räknar om hur stor del som ska synas utifrån att maxvärdet för båda är 5.

## **Toppen eller botten**

Gjorde en temporär (eller permanent) CSS klass för att testa lägga scorebar på botten. Sätt class="botten" i huvud-taggen så ska den hamna i botten istället. <huvud class="botten">

### **BUGG-KOLL**

När man har låst score-bar uppe eller nere i skärm ligger den ovanpå allt. Det gör att text och länkar kan hamna under och ej gå att nå även om man scrollar. Speciellt om man lägger score-bar i botten. Jag har justerat marginaler för att undvika detta, men nåt vi får testa i olika skärmupplösningar.

En annan "bugg" är att död-siffran ej syns när värdet är 5 (dvs start-värdet (tror jag)). Försöker se hur detta kan åtgärdas. Det har nåt att göra med att död mätaren går åt "fel" håll.

# Ljud och musik

Inget har förändrats i hur ljud och musik spelas. Skriver in i denna dokumentation senare.

## Visual FX

Inget av detta har ändrats senaste tiden. Skriver in i denna dockumentation senare.