



Översikt

- Arbeta med filer.
- Lägg till media.
- Reagera på plats.
- Profilerar webbsida.

.eerec

Lektion 1: Arbeta med filer

- HTML5 interfaces.
- Blob interface.
- File interface.
- FileList interface.
- FileReader interface.
- Läs textfil.
- Läs binärfil.
- Implementera Drag-and-Drop.

.eerec

HTML5 interfaces

- HTML5 har ett standardiserat File API som ger möjlighet för HTML-sida att interagera med fil.
- Fyra interface finns definierade:
 - Blob interface.
 - File interface.
 - FileList interface.
 - FileReader interface.

.eeec

Blob interface

- Representerar oföränderliga information, i formen raw.
- Har ett attribut som beskriver typ av media, exempelvis "text/plain".
- Har metoden slice(), som returnerar en del av blob, användbart om incremental uppladdning skall ske.

.eeec

File interface

- Interfacet ärver från blob och representerar enskild fil.
- Har två ytterligare läsbara attribut som närmre beskriver status för fil:
 - name, representerar namn för fil, utan sökväg, i en sträng.
 - lastModifiedDate, representerar datum när fil senaste modifierades, i form av Data object.

.eeec

FileList interface

- Interfacet representerar samling av File objects.
- Två metoder att hantera FileList object på webbsida:
 - Definierar `<input type="file">` element på HTML-sida, på sidan hanteras händelsen `change`.
 - Implementera Drag-and-drop funktion, när användare för fil till zon, kommer `change` händelse att triggas.

.eeec

FileReader interface

- Interfacet ger möjlighet för applikation att läsa fil eller blob till JavaScript variabel.
- Tillhandahåller tre metoder för att läsa information:
 - `readAsText()`, läser fil eller blob och gör informationen tillgänglig som plain text. Användbart om du vill läsa innehåll för fil.
 - `readAsDataURL()`, läser fil eller blob och gör dess innehåll tillgängligt som data URL. Användbart om du vill läsa innehåll för binärfil såsom en bild.
 - `readAsArrayBuffer()`, läser fil eller blob och gör innehållet tillgängligt som ArrayBuffer. ArrayBuffer representerar ett avgränsat område av bytes.

.eeec

FileReader interface (forts.)

- FileReader objektet läser innehåll asynkront och använder händelser för att indikera processen.
- Händelser som är definierade:
 - `progress`, genereras hela tiden under bearbetningen, för att indikera hur långt processen har gått.
 - `load`, indikerar att informationen har laddats. Innehåll för fil finns i ett attribut, `result`, för FileReader objektet. Resultatet är ett Blob objekt när `readAsText()` eller `readAsDataURL()`, eller ett ArrayBuffer objekt om `readAsArrayBuffer()` har använts.

.eeec

FileReader interface (forts.)

- Händelser som är definierade:
 - abort, indikerar att information som laddas har avbrytits, eventuellt genom anrop till metoden abort().
 - error, indikerar att ett fel har uppstått under laddning. Fel kan uppstå av olika skäl.
 - loaded, indikerar att läsoperationen är komplett, antingen har lyckats eller misslyckats.

.eeec

Läsa textfil

- Metoden `FileReader readAsText()` ger möjlighet att läsa fil i det lokala filsystemet.
- Arbetsgång:
 - hämta File eller Blob object, antingen genom elementet `<input type="file">` eller genom drag-and-drop.
 - Skapa ett `FileReader` objekt och hantera händelser såsom `load` och `error`.
 - Kalla på `readAsText()` för `FileReader` objektet.
 - I funktionen `load event handler`, accessa textinnehållet i egenskapen `result`.
 - För funktionen `error event handler`, implementera hantering för detta.

.eeec

Läsa binärfil

- Metoden `FileReader readAsDataURL()` ger möjlighet att läsa binär fil i det lokala filsystemet.
- Arbetsgång:
 - hämta File eller Blob object, antingen genom elementet `<input type="file">` eller genom drag-and-drop.
 - Skapa ett `FileReader` objekt och hantera händelser såsom `load` och `error`.
 - Kalla på `readAsDataURL()` för `FileReader` objektet.
 - I funktionen `load event handler`, accessa textinnehållet i egenskapen `result`.
 - För funktionen `error event handler`, implementera hantering för detta.

.eeec

Implementera Drag-and-Drop

- HTML5 har stöd för drag-and-drop.
 - Användare kan dra HTML-element eller filer från det lokala filsystemet.
 - Användare kan droppa detta till element som kan ta emot detta.
- För att stödja funktionen:
 - Slå på funktionen `drag` på HTML-element.
 - Slå på funktionen `drop` för HTML-element.
 - Hantera funktionen `dragover` och händelser knutna för funktionen.

.eeec

Implementera Drag-and-Drop (forts.)

Händelsen `dragstart` startas när vi placerar musen över det som skall flyttas, i detta fallet logotyp och vi försöker flytta på den. Händelsen `drag` startas när själva flyttningen sker. Sen kommer händelsen `dragend` som betyder att dragningen är över.



.eeec

Övning Arbeta med filer (övning8a.pdf)



.eeec

Lektion 2: Lägg till media

- Spela video med `<video>` tag.
- Skapa videospelare.
- Interaktion video och JavaScript.
- Händelser för videoobjektet.
- Hantera audio med `<audio>` tag.

.eeec

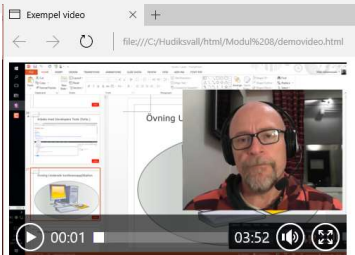
Spela video med `<video>` tag

- HTML5 ger möjlighet att spela upp video utan någon plugin.
- Använd `<video>` tag med eller utan attribut.
- Attribut som kan användas:
 - `width` och `height`, anger bredd och höjd för videospelaren, anges i pixlar.
 - `poster`, ger möjlighet att specificera bild som visas under nerladdning.
 - `controls`, anger om knappar för hantering skall visas.
 - `autoplay`, anger om video automatiskt skall spela upp.
 - `loop`, anger om video skall startas om när den är färdigspeland.
 - `muted`, anger om ljudet skall vara avstängt.
 - `preload`, används för att definiera beteende om vad som skall laddas in när sidan laddas, `preload="metadata"`.

.eeec

Skapa videospelare

```
<video id='myVideo1' width='480' height='280' controls preload='metadata'>
  <source id='mp4' type='video/mp4' src='javascriptvideo.mp4'>
  Your browser does not support the video tag.
</video>
```



.eeec

Skapa videospelare (forts.)

- `src`, anger källa för videofilm.
- `type`, anger MIME-typ för källan, `video/mp4`, `video/webm` och `video/ogg`.
- Om du inkluderar flera format, kommer webbläsare att välja det format som läsaren föredrar. Tänk på att `.mp4` måste komma först om stöd för IOS skall finnas.

```
<video id='myVideo1' width='480' height='280' controls preload='metadata'>
  <source id='mp4' type='video/mp4' src='javascriptvideo.mp4'>
  <source id='webm' type='video/webm' src='javascriptvideo.webm'>
  <source id='ogg' type='video/ogg' src='javascriptvideo.ogv'>
  Your browser does not support the video tag.
</video>
```

.eefcc

Interaktion video och JavaScript

- Videoelement kan skapas med JavaScript.
- Videobjektet har ett antal egenskaper och funktioner som ger möjlighet att kontrollera visning av video:
 - `play()`, funktion som spelar video.
 - `pause()`, funktion som pausar uppspelningen.
 - `paused`, är en egenskap som indikerar var video har blivit pausad.
 - `currentTime`, egenskap som hämtar eller sätter nuvarande tid för video.
 - `duration`, egenskap som innehåller längd på video.
 - `volume`, egenskap som hämtar eller sätter inställning för ljud.
 - `playbackRate`, egenskap som hämtar eller sätter inställning för hastighet när video spelas upp.

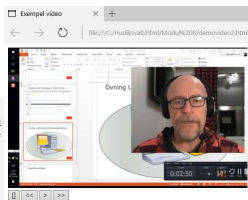
.eefcc

Interaktion video och JavaScript (forts.)

```
function vidplay() {
  var video = document.getElementById("myVideo1");
  var button = document.getElementById("play");
  if (video.paused) {
    video.play();
    button.textContent = "|";
  } else {
    video.pause();
    button.textContent = ">";
  }
}

function restart() {
  var video = document.getElementById("myVideo1");
  video.currentTime = 0;
}

function skip(value) {
  var video = document.getElementById("myVideo1");
  video.currentTime += value;
}
```



.eefcc

Händelser för videoobjektet

- Videoobjektet har ett antal händelser, som ger det möjligt för dig att kontrollera t ex tillgänglighet för innehåll, kontroll av status och nuvarande position.
- Några av de vanligaste:
 - `loadedmetadata`, händelse genereras när tillräcklig mängd metadata har laddats, oftast videolängd.
 - `loadeddata`, händelse genereras när video har laddats.
 - `timeupdate`, händelse genereras under uppspelning, innehåller nuvarande tid.

.ccrec

Hantera audio med `<audio>` tag

- HTML5 tillhandahåller `<audio>` tag, för att spela upp ljudfil, utan att behöva plugin.
- Enklaste sättet att arbeta med audio:

```
<audio src="MinLjudFil.mp3"></audio>
```
- Precis som med video kan du använda JavaScript för att hantera uppspelningen.

.ccrec

Övning Arbeta med video och audio (övning8b.pdf)



.ccrec

Lektion 3: Reagera på plats

- HTML5 Geolocation API.
- Begära information.
- Parametrar.
- Hantera felmeddelande.
- Google Maps.

.eeec

HTML Geolocation API

- Ger möjlighet för läsare att räkna ut longitud och latitude för nuvarande plats.
- Värdenhet kan använda olika tekniker för att inhämta informationen:
 - IP-adress, relativ osäker.
 - GPS positionering, vanlig bland mobilaenheter, kan vara långsam och osäker funktion inomhus.
 - Wi-Fi, triangulering mellan olika accesspunkter.
 - Telefonlokalisering, triangulering baserat på avstånd till master.
 - Användaredefinierad, applikation frågar efter informationen.

.eeec

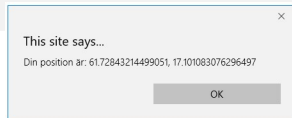
Begära information

- HTML Geolocation API stödjer två typer av förfrågningar:
 - One-shot position request, arbetar med metoden navigator.geolocation.getCurrentPosition().
 - Repeated position updates, arbetar med metoden navigator.geolocation.watchPosition(), metoden kommer att returnera värde som används som ID. För att stoppa ta emot uppdateringar, använda metoden navigator.geolocation.clearWatch() och skicka med ID som parameter.

.eeec

Begära information -exempel

```
<script>
function getCurrentPosition() {
  navigator.geolocation.getCurrentPosition(
    function(position) {
      alert("Din position är: "
        + position.coords.longitude);
    }
  );
}
</script>
```



.eeec

Parametrar

- Call-back funktion, kommer att kallas på av webbläsare när platsinformation är tillgänglig. Om `getCurrentPosition()` används kommer detta att ske en gång, om `watchPosition()` används kommer call-back funktionen att anropas kontinuerligt, tills `clearWatch()` används.
- Som tillägg finns funktion för eventuella rapporteringar av fel.
- `PositionObject`, ett objekt som ger möjlighet att konfigurera noggrannhet.

.eeec

Bearbeta informationen

- När platsinformation är tillgänglig, kommer webbläsare kalla på call-back funktion som har specificerats.
- Call-back funktionen tar emot ett objekt som har egenskapen `coords`.
- `Coords` innehåller: `latitude`, `longitud`, `accuracy` (i meter).
- Kan dessutom innehålla `altitude`, `altitudeAccuracy`, `heading` och `speed`, detta beror på kapacitet för enhet.

.eeec

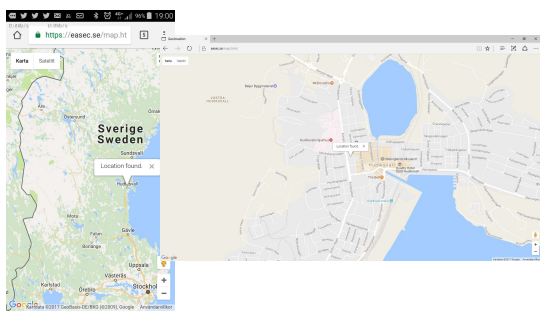
Hantera felmeddelande

- Om ett fel skulle uppstå, kommer webbläsare att kalla på error call-back function som specificerades när anropet gjordes.

```
function(errorObject) {  
    alert("Fel vid inhämtning av position");  
}
```

.essec

Google Maps



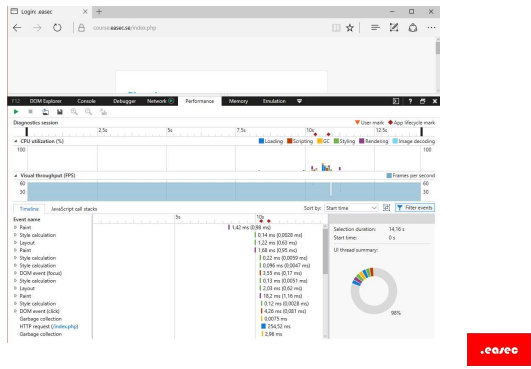
.essec

Övning Arbeta med platsangivelse(ovning8c.pdf)



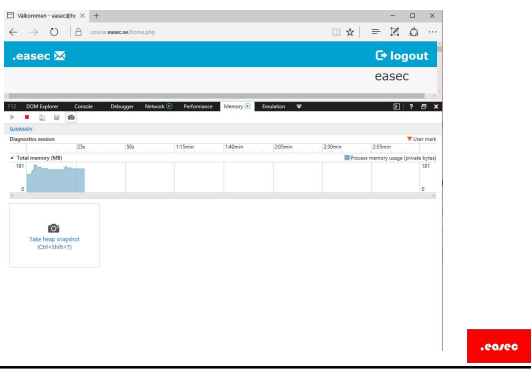
.essec

Utvecklareverktyg -prestanda



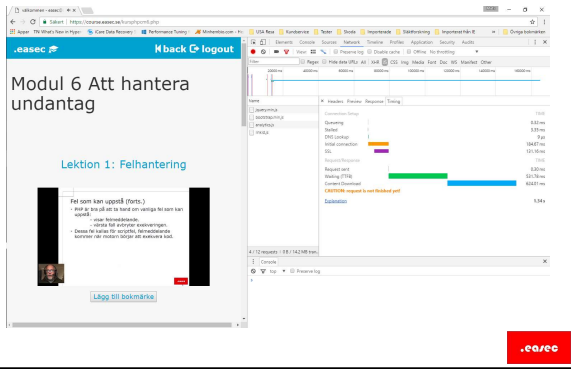
.easesc

Utvecklareverktyg -minne



.easesc

Utvecklareverktyg -Google Chrome



.easesc

Repetitionsfrågor

.course
