

Informatik

Tentamen i JavaScript för webbutveckling (ISGB15-provkod 1500) fredagen den 19 oktober 08.15-13.15.

Ansvariga lärare:	Peter Bellström	
Betygsgränser:	Max:	40
	Godkänd:	20
	Väl Godkänd:	30

Maxpoäng på respektive uppgift anges inom parantes.

Inga hjälpmedel!

Förklara bilder och begrepp noggrant!

OBS! Nytt blad för varje uppgift och glöm inte skriva anonymitetskod.

LYCKA TILL!

Uppgift 1 (6 poäng)

Redogör *exakt* för vad du som användare ser innanför body-elementet (när du trycker F12 i Firefox och väljer fliken "Inspekterare" samt klickar på alla pilar för att visa all text efter att sidan har laddats och JavaScript-koden har exekverats (se HTML5-koden och JavaScript-koden nedan). Observera att du skall redogöra för vad du ser och *inte* hur webbläsaren tolkar det du ser!

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Uppgift 1.</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8          <div id="divOne">
9              <ol>
10                 <li>Första li utan data-url och abbr</li>
11                 <li>
12                     data-url="https://developer.mozilla.org/en-US/docs/Web/HTML">
13                     <abbr title="HyperText Markup Language">HTML</abbr>
14                 </li>
15                 <li data-url="https://developer.mozilla.org/en-US/docs/Web/CSS">
16                     <abbr title="Cascading Style Sheets">CSS</abbr>
17                 </li>
18                 <li data-url="https://developer.mozilla.org/en-US/docs/Web/HTTP">
19                     <abbr title="Hypertext Transfer Protocol">HTTP</abbr>
20                 </li>
21                 <li>Andra li utan data-url och abbr</li>
22             </ol>
23         </div>
24         <div id="divTwo"></div>
25         <script src="uppgift1.js"></script>
26     </body>
27 </html>

```

```

1  "use strict";
2
3  function createContent(nodeOne, nodeTwo) {
4      let abbr = nodeOne.querySelector("abbr");
5      let text = abbr.firstChild.nodeValue;
6      let url = nodeOne.getAttribute("data-url");
7      let title = abbr.getAttribute("title");
8      let tdNode = document.createElement("td");
9      tdNode.appendChild(document.createTextNode(text));
10     nodeTwo.appendChild(tdNode);
11
12     tdNode = document.createElement("td");
13     tdNode.appendChild(document.createTextNode(title));
14     nodeTwo.appendChild(tdNode);
15
16     tdNode = document.createElement("td");
17     tdNode.appendChild(document.createTextNode(url));
18     nodeTwo.appendChild(tdNode);
19 }
20
21 function prepareGUI() {
22     let liNodes = document.querySelectorAll("li[data-url]");
23     let tableNode = document.createElement("table");
24     document.querySelector("#divTwo").appendChild(tableNode);
25
26     for(let i = 0; i < liNodes.length; i++) {
27         let liNode = liNodes.item(i);
28         let trNode = document.createElement("tr");
29         tableNode.appendChild(trNode);
30         createContent(liNode, trNode);
31     }
32 }
33 prepareGUI();

```

Uppgift 2 (6 poäng)

Skriv JavaScript-funktionen `changeParagraphs()` i vilken du skall modifiera det användaren ser för alla p-element med attributet `data-description`. Efter att funktionen har exekverat skall det användaren ser i varje p-element, texten som finns mellan start och slut elementet, vara samma värde som finns i `data-description` attributet för aktuellt p-element. Din lösning skall också vara skriven så att den fungerar med p-element utan befintlig text och med p-element med befintlig text inklusive tab, enter och mellanslag. Att använda egenskaperna `textContent`, `innerHTML` eller `outerHTML` är inte tillåtet utan du skall använda dig av metoder och egenskaper som används och rekommenderas i kurslitteraturen. Det betyder att om det finns ett text-objekt i p-element-objektet skall det användas och om inte måste ett nytt text-objekt skapas och placeras i p-elementet-objektet. Bilden nedan kan vara till hjälp men skall bara ses som ett exempel på antalet p-element och dess innehåll.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Uppgift 2.</title>
5      <meta charset="utf-8">
6    </head>
7    <body>
8      <div id="oldContent">
9
10         <p data-description="HTML">HyperText</p>
11         <p>Ett första p-element utan attribut.</p>
12         <p data-description="CSS"> </p>
13         <p data-description="JavaScript"></p>
14         <p>Ett andra p-element utan attribut.</p>
15       </div>
16       <script src="uppgift2.js"></script>
17     </body>
18 </html>
```

Uppgift 3 (10 poäng)

Utgå ifrån HTML5-koden nedan och skriv:

- JavaScript-funktionen `mouseoverP()` som tar emot händelseobjektet `e`. I funktionen skall du i konsolen skriva ut värdet (texten) som finns för första barn-elementet som finns i elementet där händelsen inträffande (skilj på var händelsen inträffar och var den fångas). (2p).
- JavaScript-funktionen `createElements()`. I funktionen skall du skapa textobjekt och placera innanför `body`-elementet. Antalet textobjekt skall vara lika många som aktuellt värde för `input type=range`-elementet. Texten i första textobjektet skall vara 1 följt av ett mellanslag och sista textobjektet värdet för `input type=range`-elementet följt av ett mellanslag. (2p)
- JavaScript-funktionen `rangeValueChanged()`. I funktionen skall aktuellt värde för `range`-elementet visas i elementet med `id=sliderValue`. (2p)
- Skriv JavaScript-funktionen `prepareGUI()`. I funktionen skall lägga till lyssnare för när du som användaren ändrar värdet i `input type=range` elementet anropas funktionen **rangeValueChanged** som du skapade i en tidigare uppgift. I funktionen skall du också lägga till en lyssnare för när du som använder klickar på `input type=button` anropas funktionen **createElements** som du också skapad i en tidigare uppgift. Avslutningsvis skall du lägga till lyssnare för när användaren rör pekaren över något av alla `p`-elementen så skall du anropa funktionen **mouseoverP** som du skapade i en tidigare uppgift. Observera att din lösning skall vara skriven så att antalet `p`-element inte spelar någon roll samt att din lösning skall vara **unobtrusive!** (4p)

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Uppgift 3.</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8     <p>1234</p>
9     <p>2345</p>
10    <p>3456</p>
11    <span id="sliderValue">1</span>
12    <input type="range" min="1" max="100" value="1" />
13    <input type="button" value="Create" />
14    <script src="uppgift3.js"></script>
15  </body>
16 </html>
```

Uppgift 4 (6 poäng)

Utgå från HTML-koden nedan och skriv Javascript-koden för en funktion som heter `changeTable()`. I funktionen ska du göra följande tillägg/ändringar i tabellen:

- Lägg till (texten som finns skall kvarstå) texten HTML sist i första td-elementet i första tr-elementet i tbody-elementet.
- Lägg till ett nytt td-element med texten CSS sist i andra tr-elementet i tbody-elementet.
- För sista td-elementet i tredje tr-elementet i tbody-elementet ändra så texten blir HTTP.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Uppgift 4.</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8     <table>
9       <thead>
10        <tr><th>url</th><th>abbr</th><th>title</th></tr>
11      </thead>
12      <tbody>
13        <tr>
14          <td>https://developer.mozilla.org/en-US/docs/Web/</td>
15          <td>HyperText Markup Language</td>
16          <td>HTML</td>
17        </tr>
18        <tr>
19          <td>https://developer.mozilla.org/en-US/docs/Web/CSS</td>
20          <td>Cascading Style Sheets</td>
21          <td></td>
22        </tr>
23        <tr>
24          <td>https://developer.mozilla.org/en-US/docs/Web/HTTP</td>
25          <td>HTTP</td>
26          <td></td>
27        </tr>
28      </tbody>
29    </table>
30    <script src="uppgift4.js"></script>
31  </body>
32</html>
```

Uppgift 5 (7 poäng)

HTML-dokumentet nedan innehåller ett formulär för att registrera personinformation. Skriv Javascript-koden för funktionen `validateRegistration()` som körs när formuläret skickas. Funktionen ska uppfylla följande krav:

1. Du ska validera fältvärdena enligt följande:
 - *E-postadress* får inte lämnas tomt
 - *Kod* får inte lämnas tomt
 - *Kod* måste vara ett giltigt tal
 - *Jag samtycker till att lagra personuppgifter (GDPR)*, måste vara ikryssad
2. Undantagshantering ska användas.
3. Om något av villkoren inte uppfylls ska ett undantagsobjekt kastas som innehåller ett lämpligt felmeddelande.
4. Om ett undantag kastas ska ett felmeddelande visas i div-elementet med id *error-message*.
5. Om ett undantag kastas ska funktionen returnera *false*.
6. Om inget undantag kastas ska funktionen returnera *true*.

```

1 <!DOCTYPE html>
2 <html lang="sv">
3   <head>
4     <meta charset="utf-8">
5     <title>Uppgift 5</title>
6   </head>
7   <body>
8     <div id="error-message"></div>
9     <form action="uppgift5.html" method="post" onsubmit="return validateRegistration();">
10      <div>
11        <label for="email">Mejl:</label>
12        <input type="email" id="email" />
13      </div>
14      <div>
15        <label for="code">kod:</label>
16        <input type="text" id="code" />
17        <a href="uppgift5.html">Har du ingen kod erhåller du koden här!</a>
18      </div>
19      <div>
20        <input type="checkbox" id="terms" />
21        <label for="terms">Jag samtycker till att lagra personuppgifter (GDPR).</label>
22      </div>
23      <div>
24        <button type="reset">Rensa</button>
25        <button type="submit">Skicka</button>
26      </div>
27    </form>
28    <script type="text/javascript" src="uppgift5.js"></script>
29  </body>
30 </html>

```


Uppgift 6 (5 poäng)

När användaren öppnar HTML-dokumentet nedan skapas ett antal DOM-element. Redogör för hur innehållet i main-elementet ser ut efter att sidan laddats. Du ska alltså förklara hur DOM-strukturen ser ut, dvs det som syns i inspekteraren/granskaren i webbläsarens utvecklarverktyg när du fäller ut main-elementet och eventuella underelement.

```
1 <!DOCTYPE html>
2 <html lang="sv">
3   <head>
4     <meta charset="utf-8">
5     <title>Uppgift 6</title>
6   </head>
7   <body>
8     <main id="container"></main>
9     <script type="text/javascript" src="uppgift6.js"></script>
10  </body>
11 </html>
```

```
1 "use strict";
2
3 addEventListener("load", init);
4
5 function createOption(value, text) {
6   let optionNode = document.createElement("option");
7   optionValues(optionNode, value, text);
8   return optionNode;
9 }
10
11 function optionValues(optionNode, value, text) {
12   optionNode.innerHTML = text;
13   optionNode.setAttribute("value", value);
14   if(value % 2 === 0) {
15     optionNode.style.color = "pink";
16   } else {
17     optionNode.style.backgroundColor = "lightblue";
18   }
19 }
20
21 function init() {
22   let selectNode = document.createElement("select");
23   document.querySelector("#container").appendChild(selectNode);
24   for(let i = 1; i <= 4; i++) {
25     let newOption = createOption(i, i * i);
26     selectNode.appendChild(newOption);
27   }
28 }
```