**1. Það eru til þrjár leiðir að binda event við element, hver þeirra leiða er best og af hverju?**

**Komdu með kóðasýnidæmi. (0.5%)**

þrjár leiðir eru:

* HTML EVENT HANDLERS
* TRADITIONAL DOM EVENT HANDLERS
* DOM LEVEL 2 EVENT LISTENERS

DOM LEVEL 2 EVENT LISTENER er besta leiðin, skal sýna afhverju það er hérna á eftir

**HTML EVENT HANDLERS** er ekki gott að nota það blandar saman HTML og JavaScript saman í kóða eins og kemur fram hérna  
<a onclick="Hide()">  
 **DOM EVENT HANDLERS** Hérna er HTML afskilið og flestir vafrar styðja þennan event handler. En þú getur bara notað eitt functiom við hvern viðburð helsti ókostur við þetta document.onclick = function;  
  
**EVENT LISTENERS** þetta er vinsælasta leiðin til að meðhöndla eventa, er töluvert ólíkara en hvernig fyrri en skrifaður en þetta leyfir einum event að triggera nokkra functiona í einu þannig að það eru töluvert minni líkur að þeir fari að conflicta hvern við annan, getur verið vesen í eldri vöfrum en það eru til lausnir við það

**2. Hvað er callback? Útskýrðu með eigin orðum, komdu með kóðadæmi. (0.5%)** callback kallar á upllýsingar, sækir þá og svo gerir eitthvað við þá. En síðan er líka til eitthvað sem er kallað asynchronous callback, helsti munurinn er að í synhcronous callback þá er kallað í functionið meðan sami stackur er enn virkur en í asynchronous er það queueað og skilast þegar stackurinn er tómur.

function genericPoemMaker(name, gender) {

console.log(name + " is finer than fine wine.");

console.log("Altruistic and noble for the modern time.");

console.log("Always admirably adorned with the latest style.");

console.log("A " + gender + " of unfortunate tragedies who still manages a perpetual smile");

}

function getUserInput(firstName, lastName, gender, callback) {

var fullName = firstName + " " + lastName;

if (typeof callback === "function") {

callback(fullName, gender);

}

}

getUserInput("Michael", "Fassbender", "Man", genericPoemMaker);

Fékk kóða á, Fékk hjálp hjá Óla   
<http://javascriptissexy.com/understand-javascript-callback-functions-and-use-them/>

**4. Leiðréttu eftirfarandi kóða þannig að hann virki (ekki syntax villa). (1%)**

function checkUsername() {

var target = e.target;

}

var el = document.getElementById('username');

el.addEventlistener('blur', checkUsername, false);

**5. Event flow, Hvað þýðir false í Event Listener? (0.5%)**

true er capturing phase á meðan false er blubbling phase sem þýðir að þetta byrji á því sem var klikkað og vinnur sig upp síðan upp um parenting, semsagt ef þú ert með 4 div og það byrjar á div 1 og í div1 er div2 og í div2 er div3 og í div3 er div4 þú ert með onclick handler á þessu öllu saman þá með að gera false ertu að segja að þú viljir bubbla það síðan upp, eða þá að ef þú ýtir á div 3 þá keyrist það click event síðan div 2 og síðan eitt, með true væri það hinsvegar öfugt.

**6. Hvað gera eftirfarandi aðferðir? Komdu með dæmi um notkun (1%)**

a. stopPropagation()

event.stopPropagation(); er akkúrat til að meðhöndla og fara út úr bubbling eða capturing phase ef þú ert með marga listenera á á sama stað eða inní sama div tagi eða í li dropdown það sem event.stopPropagation(); er í raun að segja er bara þegar ég er búinn þá hættiru að halda áfram.  
li.addEventListener(„click“,function(e) {  
alert(„Smelltir á “ + e.target.classname)},true);  
<li class=“1“> 1 </li>  
<li class=“2“> 2 </li>  
<li class=“3“> 3 </li>  
<li class=“4“> 4 </li>  
stopPropagation()er hérna til að stoppa þetta, annars myndi þetta breyta út alla

**7. Quiz**

**Þegar notandi hefur smellt á einhvern svarmöguleika þá á viðkomandi svarmöguleiki að fá rauðan**

**bakgrunnslit. Notaðu Event object og Event Delegation til að ná þessu fram. (3%)**Index.html fylgir með