1. **0.25%** Afhverju er getElementById() fljótlegasta aðferðin?  
   getElementById() er fljótlegasta aðferðin þar sem að í fyrsta lagi getur bara verið eitt einstakt id á einni síðu þ.e.a.s ég get ekki verið með 2 div tög sem eru með id-ið cool, þannig að um leið og þýðandinn hefur fundið þetta getur hann strax skilað þessu hann þarf ekki að halda áfram að leita og athuga hvort það séu fleiri
2. **0.25%** Hvað gerir querySelector()? Komdu með dæmi.  
   querySelector() notar CSS greinirinn eða class attribute-ið til að velja elementið, nema querySelector velur bara þann fyrsta sem hann finnur t.d. í dæminu hér fyrir neðan er h2 og p með klasnn rautt en ef það er bara notað querySelector() þá tekur hann bara h2 elementið því það er það fyrsta sem hann finnur hinsvegar ef það hefði veriðnotað querySelectorAll("**.rautt**") þá hefði það valið bæði  
     
   <!DOCTYPE html>

<html>

<body>

<h2 **class="rautt"**>h2 með classa sem heitir rautt class=" rautt "</h2>

<p>A paragraph with **class="rautt"**.</p>

<p>Ef þú smellir á takkan þá tekur þetta bara h2 sem er með klasann rautt.</p>  
<button onclick="myFunction()">Try it</button>

<script>  
function myFunction() {  
 document.querySelector("**.rautt**").style.backgroundColor = "red";  
}

</script>  
</body>  
</html>

1. **0.5%** Hvað er málið með auða hnúta (e. Whitespace nodes) og vafra? Hvernig leysum við slíkt?  
   Þess háttar auðir hnútar geta birts í vöfrum einfaldlega ef það er tómt bil eða „whitespace“ milli elementa, þá túlka sumir vafrar það sem Element og getur það haft áhrif á skipanir eins og nextSibling, lastChild og þær skipanir sem eru ekki beint að velja neitt Element hvað varðar id eða class selector heldur eru að flakka á milli, þá t.d. getur nextSibling allt í einu skilað tómu í chrome en skilar kannski gildi í IE.  
     
   Lausnin við þessu er annahvort að nota library eins og JQuery hann hjálpar til með þetta en önnur leins er að trima skránna eða strippa hana af öllum bilum og þjappa, þetta er fín lausn hvað varðar keyrslutíman og stærð á skránni en eftir að svona er framkvæmt getur verið óþægilega leiðinlegt að vinna í eða úr henni, en til eru þó lausnir sem „reverta“ svona breytingar eða þá bara að eiga vinnsluskrá sem þú fiktar í og síðan compressar maður hana og uploadar á serverinn
2. **0.5%** Hvað gerir eftirfarandi kóði, útskýrðu línu fyrir línu  
   getElementsByTagName('li') tekur öll <li> tögin sem eru á síðunni og raðar þeim upp í nodelista sem virkar eins og array eða byrjar á 0 indexi og vinnur sig upp eftir því hvað mörg eru tilstaðar og í þeirri röð sem þau birtast á síðunni, semsagt fyrsta li sem þú sérð á síðunni yrði nr 0 næsta li nr 1 osfrv. þannig í þessu tilviki er þetta að gerast:  
      
   var elements = document.getElementsByTagName('li');   
   //Finnur öll li á síðunni og setur í elements breytuna(  
   if (elements.length > 0) {   
   //opnar if setningu og segir að ef að elements (li tögin) eru fleirri en 0 þá gerir það eftirfarandi  
    var el = elements[2];   
   // hér er vistað li tag 3 í el breytunni semsagt li tag nr 3 sem þú sérð á síðunni geymir el breytan  
    el.className = 'cool';   
   // hér er verið að gefa el (li tagi nr 3) nýtt klasanafn.  
   }   
   //endar if
3. **0.5%** Notaðu DOM element og eitthvað af eftirtöldu; parentNode, previousSibling, nextSibling, firstChild, lastChild til að velja HTML elementið sem er **feitletrað** í kóðanum.  
   <ul

><li><em>fresh</em>figs</li

**><li>pine nuts</li**

><li>honey</li

><li>balsamic vinegar</li

></ul>  
var first = document.getElementsByTagName('ul')[0].firstChild

var pinenuts = first.nextSibling.nextSibling.nextSibling