Válaszolj az alábbi kérdésekre minél részletesebben!

1. Mi a különbség a <span>, <div>, és az <article> elemek között?

A span nagyon hasonló a divhez mindkettő egy html elem. A különbség az, hogy míg a div egy blokk szintű elem, addig a span egy in-line elem. Egyébként mind a kettőt a tartalmak formázására, stílusok meghatározására szokták használni. Az article egy olyan html elem, ami egy dokumentumban, vagy az oldalon egy szabadon újrafelhasználható tartalmat hordoz és ezért inkább szemantikai jelentése van.

1. Milyen különbségeket és hasonlóságokat ismersz a cookie-k és a localStorage között?

A sütiket a felhasználó gépén egy a [web szerver](https://hu.wikipedia.org/wiki/Web_szerver) hozza létre a [böngésző](https://hu.wikipedia.org/wiki/B%C3%B6ng%C3%A9sz%C5%91) segítségével, ahol azok egy meghatározott könyvtárba kerülnek. Abban hasonló a localStorage, hogy itt is a web szerver által küldött információ tárolódik el a böngésző segítségével, de míg a sütik egy könyvtárba mentődnek, addig a localStorage csak a böngésző futásáig tárolja az adatokat. És emiatt sokkal több adatot tud itt átmenetileg tárolni.

1. Milyen http metódusokat ismersz és melyiket milyen célra használják?

GET – Egy meghatározott információ, adat letöltését kezdeményezi.

POST – Feldolgozandó adatokat küld a szervernek.

PUT – Feltölti a megadott adatokat.

DELETE – Törli a megadott adatokat.

HEAD – ugyanazt csinálja mint a GET, csak az üzenettestet hagyja ki a válaszból.

TRACE – Visszaküldi a kapott kérést.

OPTIONS - Visszaadja a szerver által támogatott HTTP metódusok listáját.

CONNECT - Átalakítja a kérést transzparens TCP/IP tunnellé.

(Utóbbi négyet még nem használtam.)

1. Mi lesz az ’x’ értéke és miért?

var a = b = x = 0;

x = ++a + b++;

Az értéke 1 lesz. Mert a ++a kifejezés értéke 1, míg az b++ -é 0.

1. Mi kerül az x változóba és miért?

var x = (a = 3);

Az x értéke 3 lesz, mert az x értékét megfeleltetem a-nak.

1. Mit ír ki és miért?

var bar = function() { console.log("bar"); }

var foo = function() { console.log("foo"); }

console.log("baz");

setTimeout(foo, 1000);

setTimeout(bar, 500);

A logolás sorrendje: baz -> bar -> foo.

Először a baz fut le, mert ott nincs setTimeout beállítva. Ezt követi 500 ms múlva a bar, majd szintén 500 múlva a foo.

1. Mi a különbség a két változó között?

var v1 = document.getElementById(’inputText’);

var v2 = $(’#inputText’);

Nincs különbség köztük. A v1 egy sima javascript selevtor, a v2 pedig egy jQuery selector. Mindkettőben az ID alapján határozzuk meg a kiválasztandó elemet, ami az ’inputText’.

1. Mely mód(ok)on lehet elérni a „John Smith” szöveget tartalmazó változót?

var data = {

’person.name’: ’John Smith’

};

Ezzel ki lehet logolni, ha egy elemű az objektum: console.log(data);

De ez is jó lehet: console.log(Object.values(data));

Vagy ez: for(key in data) {

if(data.hasOwnProperty(key)) {

var value = data[key];

console.log(value);;

}}

1. Mit ír ki és miért?

var callbacks = [];

for (var i = 0; i < 5; ++i) {

callbacks.push(function() { console.log(i); });

}

callbacks[3]();

Amit kiír: 5

Az i értékét logolja ki, ami elmegy ötig és itt látja, hogy ez nem felel meg a ciklusban foglaltaknak, ezért leáll.

1. Mit ír ki és miért?

var a = 2, b = { c: 4 };  
var x = a, y = b;

x = 5;  
y.c = 6;

console.log(a + b.c);

Az eredmény 8 lesz.

1. Mi a különbség a két CSS selector között?

.element .symbol {}

.element.large .symbol {}

.element .symbol {} – az element és a symbol osztályokat is kijelöljük, de ebben az esetben nem feltétlenül van az osztályok között parent-child kapcsolat, vagy kiválasztja azokat a html elemeket, amelyek rendelkeznek mindkét osztállyal egyszerre.

.element.large .symbol {} – az element osztály child osztálya a large és ezen belül lévő symbol osztályt/osztályokat jelöljük ki

1. Milyen színű lesz a felirat és miért?

<div class="form-square">

<div class="seven-col">

Hello World!

</div>

</div>

div.form-square > div {

color: blue;

}

.seven-col {

color: red;

}

A színe kék lesz. Mivel ez: div.form-square > div a css selector hierarchiában egy sima osztály felett áll.

1. Adatbázis tervezési feladat

Tervezd meg egy webáruház adatbázis sémáját – relációs adatbázisban (táblákban gondolkodj)!

Termékeket szeretnénk tárolni úgy, hogy az adminok a felületen vehessék fel a termékeket. A termékhez bármennyi, különböző típusú attribútum tartozhat. Új terméktípus, vagy új termék-attribútum felvételéhez ne kelljen sémát módosítani!

Példa a termékekre és attribútumaikra:

Dell Vostro X500  
- memória: 16gb   
- processzor magok száma: 4   
- bevezetés ideje: 2015-12-31

Braun A1800 hajszárító- szín: fekete, fehér  
- erősség: 1800W  
- hideglevegő fújás: igen   
- garanciális hónapok száma: 12

A cél tehát olyan adatstruktúrát kialakítani, hogy ha ezentúl notebookoknál a kijelző felbontását is tárolni szeretnénk, akkor ehhez ne kelljen az adattáblák struktúráját módosítani. Ha több alternatív megoldást is találsz, írhatsz többet is. Az adatstruktúrát bármilyen formában leírhatod – csak értsük meg ☺

**Products**

Product\_id

Product\_name

Ebben a táblában tárolom a termék azonosítóját és nevét (ID –> auto increment).

**Attributes**

Attribute\_id

Attribute\_name

Attribute\_value

Ebben a táblában az attribútumokat tárolom. Itt is kell egy azonosító az attribútumok azonosítására, valamint egy oszlop az attribútum névnek és értéknek.

**Product\_map**

Map\_id

Product\_id

Attribute\_id

És kell egy tábla ahol az adott terméknévhez tudom hozzárendelni a hozzá tartozó attribútumokat.