תשובות לשאלות האמריקאיות\פתוחות בבחינה:

1. **מה יודפס ?**

let x = 5;

function a(b) {

console.log(x);

}

function b() {

x = 6;

console.log(x);

a(b);

}

b();

1. 5,5
2. 6,5
3. 6,6
4. 6, RefrenceError : “x” is not defined
5. **תאר את ההבדל בין var ל let לבין const**

**Var:  
- מגדיר משתנה** לוקאלי **לכל דבר ועניין, אם מוגדר** בתוך פונקציה**.   
- מגדיר משתנה** גלובלי **אם מוגדר בבלוק קוד כלשהו שאינו פונקציה (לדוגמא בתוך if שיש לי בסקריפט שלא כחלק מפונקציה).**

**Let: מגדיר משתנה שהוא mutable ולוקאלי לscope בו מוגדר (לדוגמא, scope גלובלי, scope לוקאלי לפונקציה, scope לוקאלי בין {}).**

**Const: מגדיר משתנה לוקאלי ל-scope ההגדרה (כמו ב-let), אך שאינו mutable. ניסיון לשינוי (ישיר, על ידי השמה) יגרום לבעיה.**

1. **מה יהיה הערך של x?**

**const x = [1 , 2 , 3].indexOf(“a”);**

* 1. 0
  2. 1
  3. -1
  4. RefrenceError
  5. TypeError

1. **מה יהיה הערך של x**

**const x = [1 , 2 , 3].indexOf(“1”);**

* 1. 0
  2. 1
  3. -1
  4. RefrenceError
  5. TypeError

1. **מה יהיה הפלט של הקוד הבא?**

**console.log( 1 ==**= **2 && “howdy” )**

* 1. false “howdy”
  2. false
  3. “howdy”
  4. SyntaxError

**(אני מתעלם מזה שחסר ";" בסוף)**

1. **מה יהיה הפלט של הקוד אם נחליף את התנאי ל 1 === 1 ?**
   1. false “howdy”
   2. false
   3. “howdy”
   4. SyntaxError

**(אני מתעלם מזה שחסר ";" בסוף כמובן גם כאן )**

1. **האם זה חוקי לשנות href של אלמנט a? פרט**

**אני לא מכיר שום דבר שמונע שינוי כזה.**

1. **הסבר מה החשיבות של גרסת דפדפן לכתיבת קוד javascript. כיצד ניתן להתמודד עם הקשיים העולים מכך?  
     
   javascript דוגלת בתאימות לאחור, כלומר שאני יכול תמיד להריץ קוד ישן בגרסאות חדשות יותר של מנועים. לכן פה אין חשיבות (קוד ישן יותר מהתאריך של היום ירוץ על כל מנוע מהיום לעתיד, בהבטחה מהסטנדרט של JS).  
     
   היא לא דוגלת ב"תאימות לפנים", כלומר – לא נכון שברגע שיש feature חדש, הוא יכול לרוץ על מנועים ישנים יותר. זו בעיה כי אני לא יכול (אולי) להריץ קוד מודרני על מנועים מיושנים.  
     
   הפתרון הוא לבצע transpiling לקוד שלי, כלומר "לקמפל" אותו לגרסאת javascript ישנה יותר. יש טכניקות לעשות זאת ידנית, אבל אין צורך כי יש גם כלים אוטומטיים למטרה זו (והם לעיתים מייצרים קוד יותר יעיל).**

1. **לפניך קטע קוד מה יהיה הפלט? נמק.**

**const a = [“h” , “e” , “l” , “l” , “o”]**

**const b = a.join().split()**

**console.log(a === b)**

1. true
2. false

הקוד לא רץ כי אין ";" אבל מתעלם.

הפונקציה split ללא ארגומנט לא עושה כלום. לכן הערך של -b הוא הערך של: a.join() .  
הפונקציה join ללא ארגומנט, לפי דיפולט, יוצרת את המחרוזת המתקבלת מconcatenation של תאים במערך עם ה"מפריד" שהוא פסיק (התיו “,” ). לכן, הערך של b הוא:  
h,e,l,l,o כמחרוזת. אלו לא שווים, לכן יוחזר false.

1. **מה יהיה הערך של x?**

**const x = “a>d”.split(‘>’);**

* 1. [“a>”,”d”]
  2. [“a” , “>” , “d”]
  3. [“a” , “>d”]
  4. [“a” , “d”]
  5. [“a”, “”, “d”]