**Block 1**

1. ***function funcA() {***
2. ***console.log(a);***
3. ***console.log(foo());***
4. ***var a = 1;***
5. ***function foo() {***
6. ***return 2;***
7. ***}***
8. ***}***
9. ***funcA();***

הפונקציה funcA תרוץ. בגלל שa הוא var הוא משתנה גלובאלי ולכן הוא יהיה קיים בשורה 2 למרות שהוא מוגדר בפעם הראשונה רק בשורה 4, אבל הוא מקבל ערך רק בשורה 4 ונוצר כundefined ולכן יודפס לקונסול undefined. לאחר מכן יודפס לקונסול 2 כי הפונקציה foo חיה בסקופ של funcA ולכן אפשר לקרוא לה עוד לפני שהיא נכתבה (בעצם מתייחסים אליה כאילו היא הדבר הראשון שנכתב בקוד בסקופ- כלומר היא קיימת בכל רגע נתון תוך הפונקציה funcA גם אם מבחינת סדר כתיבה היא נכתבת אחרי שקראו לה.

**Block 2**

1. ***var fullName = 'John Doe';***
2. ***var obj = {***
3. ***fullName: 'Colin Ihrig',***
4. ***prop: {***
5. ***fullName: 'Aurelio De Rosa',***
6. ***getFullName: function () {***
7. ***return this.fullName;***
8. ***}***
9. ***}***
10. ***};***
11. ***console.log(obj.prop.getFullName());***
12. ***var test = obj.prop.getFullName;***
13. ***console.log(test());***

יודפס 'Colin Ihrig' כי this.fullName מתייחס לאבא של הפונקציה getFullName שהוא prop ולכן המשתנה שווה ל'Colin Ihrig'. לאחר מכן לתוך test מוכנסת הפונקציה obj.prop.getFullName במקום התוצאה שלה כי היא נקראת עם סוגריים. לכן צריך להוסיף סוגריים ואז מה שיוכנס לתוך test יהיה 'Colin Ihrig'. כמו שכתוב עכשיו, בקריאה לtest בשורה 13 יוחזר 'John Doe' כי המשתנה fullName בסקופ של test הוא החלון הוא 'John Doe'.

**Block 3**

1. ***function funcB() {***
2. ***let a = b = 0;***
3. ***a++;***
4. ***return a;***
5. ***}***
6. ***funcB();***
7. ***console.log(typeof a);***
8. ***console.log(typeof b);***

הפונקציה funcB מחזירה את הערך של המשתנה a שחי בסקופ שלה רק. הערך לא נשמר. כדי לשמור אותו צריך את שורה 6 לשנות ל **let a = funcB();.** אחרי השינוי הזה שורה 7 תדפיס 1. b חי בסקופ של הפונקציה ולכן אין גישה אליו מחוץ לפונקציה ושורה 8 מדפיסה ללוג number כי b הוגדר כמשתנה גלובאלי (כי לא הוגדר אחרת...)

**Block 4**

1. ***function funcC() {***
2. ***console.log("1");***
3. ***}***
4. ***funcC();***
5. ***function funcC() {***
6. ***console.log("2");***
7. ***}***
8. ***funcC();***

הפונקציה funcC מוגדרת פעמיים. בגלל שכשמריצים את הקוד הוא קודם כל עובר על הפונקציות ואז רק מריץ מההתחלה עד הסוף את הקוד יודפס גם בקריאה בשורה 4 וגם בקריאה בשורה 8 בקונסול ,׳2׳ כי ההגדרה השניה של הפונקציה דרסה את ההגדרה הראשונה.

**Block 5**

1. ***function funcD1() {***
2. ***d = 1;***
3. ***}***
4. ***funcD1();***
5. ***console.log(d);***
6. ***function funcD2() {***
7. ***var e = 1;***
8. ***}***
9. ***funcD2();***
10. ***console.log(e);***

בתוך הפונקציה funcD1 מוגדר משתנה d בלי מילה לפניו מה שגורם לתוכנה להגיד אותו כמשתנה גלובאלי, לכן משורה 5 יודפס לקונסול ׳1׳. בתוך הפונקציה funcD2 מוגדר משתנה e מסוג var. כשמכריזים על משתנה var בתוך הפונקציה הוא לוקאלי בפונקציה ואי אפשר לגשת אליו מחוץ לה ולכן כרגע חוזר error בושרה 10 שאומר שהמשתנה e לא מוגדר. אם רוצים לגשת לe צריך או להגדיר אותו מחוץ לפונקציה ואז הוא יהיה גלובאלי או להחזיר את הערך שלו ולשמור את הערך שחזר.

**Block 6**

1. ***function funcE() {***
2. ***console.log("Value of f in local scope: ", f);***
3. ***}***
4. ***console.log("Value of f in global scope: ", f);***
5. ***var f = 1;***
6. ***funcE();***

שורה 4 תדפיס בקונסול Value of f in global scope: undefined כי fמוצהר כמשתנה גלובאלי ולכן מוגדר כקיים מההתחלה אבל הוא מקבל ערך רק אחרי ההדפסה של שורה 4 בשורה 5. שורה 6 תדפיס בקונסול Value of f in local scope: 1 כי אין משתנה לוקאלי f אבל יש גלובאלי והערך שלו הוא 1.