**Block 1**

1. ***var b = 1;***
2. ***function someFunction(number) {***
3. ***function otherFunction(input) {***
4. ***return b;***
5. ***}***
6. ***b = 5;***
7. ***return otherFunction;***
8. ***}***
9. ***var firstResult = someFunction(9);***
10. ***var result = firstResult(2);***

כשb נוצר בפעם הראשונה הוא מכיל 1. בשורה 9 מורצת someFunction שבה משנים את הערך של b ל5. שורה 10 קוראת לפונקציה הפנימית שמחזירה את b שעדיין מכיל 5 כי לא שונה.

**Block 2**

1. ***var a = 1;***
2. ***function b2() {***
3. ***a = 10;***
4. ***return;***
5. ***function a() { }***
6. ***}***
7. ***b2();***
8. ***console.log(a);***

כשa מאותחל הוא משתנה גלובאלי שמכיל 1. כמורצת b2 יש הויסטינג ובגלל זה בשורה 7 היא לא משנה את הערך הגלובאלי של a אלא משנה את הערך הלוקלי להיות 10 (נוצר בסקופ של הפונקציה a). שורה 8 תדפיס לקונסול 1.

**Block 3**

1. ***let i;***
2. ***for (i = 0; i < 3; i++) {***
3. ***const log = () => {***
4. ***console.log(i);***
5. ***}***
6. ***setTimeout(log, 100);***
7. ***}***

עבור כל i שירוץ בין 0 ל2 בקפיצות של 1 (יהיה 0 ואז 1 ואז 2 ואז יסיים את הריצה של הלולאה) יריץ את הפונקציה log שמדפיסה לקונסול את i. בגלל שלוג הוא משתנה בתוך הלולאה הוא שייך לסקופ של הלולאה והוא מאותחל כל פעם מחדש וזה בסדר שהוא const. לוג רץ אחרי שהוא מחכה 100 מילישניות כי בשורה 6 מוגדר timeout. בפועל לקונסול יודפס 0 ואז יודפס 1 ואז יודפס 2 בהפרשים של 100 מילישניות.