מילון מונחים בשפת JAVA SCRIPT

|  |  |  |  |
| --- | --- | --- | --- |
| מונח בJS | הסבר | דוגמא בתוך קוד | הערות |
| מונחים כלליים:   * פקודה – שורת קוד. * זיכרון – המקום בו מאחסן המחשב את המידע. * זרימת תוכנה – אוסף של פקודות המריץ קוד באופן סידרתי. * CPU – center process unit. * משתנה – פקודה שמקצה מקום בזיכרון ונותנת למקום שם. * Case sensitive – האם התוכנה רגישה לשוני בין אותיות גדולות וקטנות באנגלית. JS רגישה לשוני ולכן, תא בשם 1A הוא לא זהה לתא בשם 1a . * כתיבה בשפת JS - <https://codepen.io/haimnewclass/pen/poLEGKw?editors=0010> * יש לכתוב בשורת הconsole את השורה הזו: window.CP.PenTimer.MAX\_TIME\_IN\_LOOP\_WO\_EXIT = Number.POSITIVE\_INFINITY כדי למנוע את מנגנון "מניעת לולאה אינסופית" של codepen * Lvalue – left value , בכל תוכנה/שפת קוד הערך בצד שמאל מקבל לתוכו את הערכים שכתובים מצד ימין למשוואה. | | | |
|  |  |  |  |
| ; | סיומת של כל שורת קוד. |  |  |
| Var \_\_ | פקודה להגדרת משתנה. | Var a;  a=10; | שם של משתנה יהיה אותיות באנגלית בשילוב מספר. התיו הראשון תמיד יהיה אות. קו תחתון הוא חוקי לשימוש אבל לא סימן אחר. |
| Console.log(\_\_) | פקודה המדפיסה למסך הconsole הנמצא בתחתית המסך את הפלט המבוקש. | Var c = "bla";  Console.log(c);  🡨 למסך יודפס bla |  |
| + (חיבור) - (חיסור)  / (חילוק)  \* (כפל)  % (מציג רק שארית חלוקה) | פעולות חשבון בתוך הקוד | Var num1, num2, num3, sum;  Sum = num1+num2-num3;  Sum = num1/num2\*num3; | אפשר לחפש עוד פעולות בגוגל |
| Math.random() | פקודה שבוחרת מספר רנדומלי בין 0 ל1. לא כולל 1 (עד 0.999..) | Var a = Math.random();  Console.log(a\*100); |  |
| Integer – מספר שלם  Float – מספר עשרוני  String – מחרוזת (נכתב בין מרכאות) | טיפוסים שונים של משתנים (פרטי מידע) שהתוכנה קולטת. | Var a = 12;  Var b = "shalom";  Console.log(a+b);  🡨 המחשב ידפיס "12shalom" | בשפת JS לא צריך להגדיר מראש את המשתנה, התוכנה מבינה לבד והconsole יודע מה לעשות איתם. |
| parseInt()  נקרא גם "type casting". | פונקציה שהופכת משתנה מסוג float או string למשתנה מסוג int (מספר שלם). | Var f = Mathrandom()\*100;  Var g = perseInt(f);  Console.log(g);  או  Var f = "12";  Var g = perseInt(f);  Console.log(g); | הפונקציה חותכת את המספרים אחרי הנקודה העשרונית, לא מעגלת אותם!  \*\*יש גם parseFlaot שעושה פעולה דומה רק הופך את המשתנה למספר עשרוני |
| Return \_\_; | פקודה שמחזירה ערך (Lvalue) ושומרת אותו במקום בזיכרון, כך שיהיה אפשר להשתמש בו אחר כך. | ראו דוגמא בהמשך. | \*בניגוד לconsole.log שמדפיסה למסך ופה נגמר התהליך איתה. \*הפקודה מסיימת פונקציה! ויוצאת ממנה. |
| Function \_\_\_ (\_,\_)  {  \_\_\_\_  {  1.שם  2.פרמטרים  3.גוף הפונקציה | פונקציה – כל פונקציה היא יחידה עצמאית של קטע קוד, בעלת שם, המקבלת פרמטרים ומחזירה תוצאה ע"פ המבוקש. | Function Sum (a,b)  {  Var c = a+b;  Console.log(c);  }  Sum(100,200); | \*ניתן לקרוא לפרמטרים ב2 פונקציות שונות באותו השם.  \*כדי לקרוא לפונקציה, צריך לכתוב את שמה יחד עם הפרמטרים. ניתן לקרוא לה גם משורת הקלט של הconsole. |
| פונקציה בתוך פונקציה |  | Function GAN()  {  Var a = Math.random()\*100;  a = parseInt(a);  return a;  }  Function Sum (a,b)  {  Var c = a+b;  Console.log(c);  }  Function Sum (GAN(),GAN()); |  |
| If (\_\_)  }  \_\_  {  Else  }  \_\_  {  1.תנאי  2.גוף הif  3.גןף הelse | תנאי if – קוד המאפשר בדיקה של תנאי מסוים ולאחריו מבצע פעולות בהתאם להאם התנאי מתקיים או לא. | Function Mor (a,b)  {  If (a>b)  { console.log("yes"); }  Else  { console.log("no"); }  }```` | סוגי אופרנטים (תנאים):  \*< (גדול)  \*> (קטן)  \*=== (שווה ל)  \*=! (לא שווה ל)  \*=< (גדול שווה)  \*=> (קטן שווה). |
| If בתוך if | ניתן להכניס לולאה לתוך לולאה. ובכך ליצור לולאה מורכבת. |  |  |
| הרחבה לתנאי if: && או || או !not | הוספת "וגם" או "או" או not לתוך התנאי | Function bla (num1,num2,num3)  {  If (num1>num2 && num1>num3)  {  return "num1 is biggest";  }  If (!(num1>num2 && num1>num3))  {  Return "num1 is not bigest";  } | \***&&** - וגם - כלומר הif מתקיים רק אם שני תנאים שהגדרנו מראש מתקיימים שניהם  \***||** - או - כלומר הif מתקיים רק אם לפחות אחד מהתנאים שהגדרנו מראש מתקיים  \***! –** היפוך -  !True=False  !False=True  לשים לב:  \*קדימות לסוגרים  \*קדימות ל AND |
| משתנה boolean | משתנה המכיל בתוכו רק את הערכים true או false. כלומר, ניתן להכניס לתוכו תנאי כלשהו והמשתנה ישמר בתוכו רק אם התנאי היה נכון או לא נכון. | Function Booltest()  {  Var bool = true;  Var bool2 = false;  *אין צורך לכתוב את שתי השורות הנ"ל, רק לצורך המחשת הרעיון.*  Var a = 1;  Var b = 2;  Bool = a>b  *לשים לב שהתנאי הבוליאני לא נכתב במרכאות או אם ; אחריו.*  Console.log(bool);  }  🡨 המחשב ידפיס false |  |
| Prompt("\_\_")  1.הוראה למשתמש | פונקציה המקבלת קלט מהמשתמש. | Var a = prompt("enter a num");  Var b = prompt("enter a num");  Var con = a>b *//Boolean!*  If (con)  {  Console.log("a is big");  }  Else  {  Console.log("a is small");  } |  |
| Alert ('\_\_\_') | פונקציה שמקפיצה הודעה (לא בקשה!) למשתמש. | Alert('end of loop'); | לשים לב להשתמש במרכאות ' ולא מרכאות '' |
| Loop – לולאת while:  While (\_\_)  {  \_\_\_  { | קטע קוד החוזר על עצמו עד שהתנאי שבראשו מפסיק להתקיים. | Function loop()  {  Var i = 0; *// the counter!*  While (i<10)  {  Console.log(i);  I=i+1;  }  Console.log("end of loop");  } |  |
| Loop – לולאת for:  For (\_\_; \_\_; \_\_\_)  {  \_\_\_  {  1.מונה הלולאה  2.תנאי לקיום  3.מקדם המונה | קטע קוד החוזר על עצמו עד שהתנאי שבראשו מפסיק להתקיים. | For (var x=1; x<=20; x++)  {  Iconsole.log(x);  } | דרך מקוצרת לכתיבת לולאת while |
| לולאה מכוננת | לולאה בתוך לולאה. ניתן לכתיבה בכל סוגי הלולאות. |  |  |
| מערך Array  Var \_\_ = new array[];  או כתיבה מקוצרת  Var \_\_ = []; | משתנה מסוג מערך, המכיל בתוכו מספר משתנים. | Function List ()  {  Var students = new array[];  Students [0] = "moshe";  Students [1] = "yossi";  Students [2] = "liat";  Var i = 0;  While (i<5)  { console.log(students[i]);  i++; *//=i+1*  }  } | לכל משתנה בתוך המערך יש מיקום, הנספר מ0 ואילך, ומאוזכר ע"י שם המערך ומספר המיקום שלו בתוך []. |
| \*הערה למערכים | \*הערה למערכים: ניתן לתת שם למערך שהוא מילה ולהכניס לתוכו ערך מספרי.  הJS לא מגביל בשם האינדקס שניתן לתת לתא מסויים במערך. | Function List ()  {  Var food = new array[];  Students [apple] = 120;  Students [banana] = 100;  Alert (arr["banana"]);  } | יש לשים לב, שאם עושים את זה לא ניתן לרוץ על המערך לפי מספרים (בלולאה). |
| \_\_\_.length  1.שם המערך | פונקציה המייצגת את אורך המערך שאת שמו הכנסנו (מספר המקומות בו). | Var arr = [1,2,3,4];  Console.log(arr.length);  🡨 המחשב ידפיס 4 | אם לדוגמא אורך המערך הוא 4, האינדקס האחרון בו הוא 3 (כיוון שהמערך מתחיל באינדקס במיקום 0) |
| Document. getElementById ("\_\_\_").value | פקודה המקשרת בין מיקום בJS לid הנמצא בhtml בתגית המשויכת לinput (קבלת קלט מהמשתמש)  אם כותבים: var x = docu.. אז מה שהמשתמש כותב בתיבת טקסט נכנס למשתנה x  אבל, אם כותבים: docu.. = var x אז תיבת הטקסט תקבל לתוכה את הערך של המשתנה x |  | |
| \_\_.push(\_\_\_)  1.שם המערך  2.טקטס חופשי | פקודה המכניסה טקסט כל שהוא (מספרים או "אותיות") לאיבר האחרון של המערך אותו בחרנו, בלי צורך לדעת או לציין את אורך המערך | Var arr = [];  Arr.push("kjhsxj"); | |
| Document. getElementById ("\_\_\_").InnerText | פקודה המקשרת בין מיקום בJS לid הנמצא בhtml בתגית המשויכת לp (פסקה)  אם כותבים: var x = docu.. אז מה שהמשתמש כותב בתיבת טקסט נכנס למשתנה x  אבל, אם כותבים: docu.. = var x אז הטקסט יקבל לתוכו את הערך של המשתנה x |  | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |