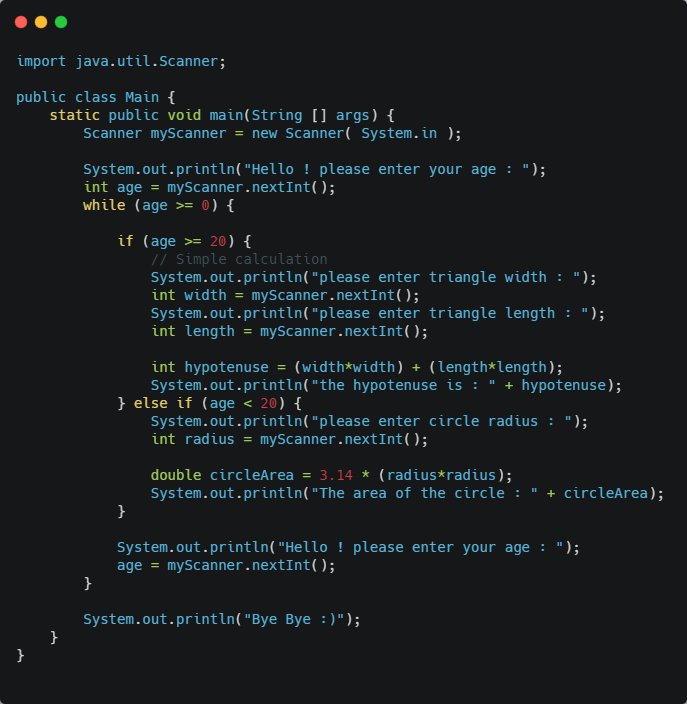
**פונקציות**

שימו לב לקטע הקוד הבא:

[](http://shorturl.at/FJKPU)

מה הבעיה בקוד הנ"ל? כמה זמן לקח לכם להבין מהי מטרת הקוד?

**הקוד ארוך** ולא מובן!

עד כה, כתבנו את הקוד בתכנית שלנו באותו המקום, שורה אחר שורה, בתוך פונקציית הmain.

הבעיה מתחילה כאשר גודל הפרויקט גדול, עד כדי אלפי שורות קוד.

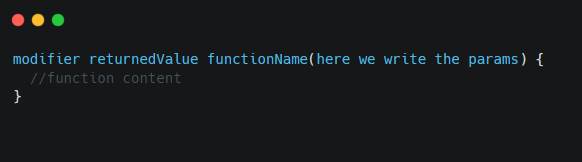
במצב כזה, כאשר כל הקוד יהיה כתוב שורה אחר שורה בתוך הmain, ניתקל בקוד מסורבל, ארוך, מסובך שמקשה גם על תחזוקה שלו לטווח ארוך.

מה הפתרון? **פונקציות**!

**פונקציה:** קטע לוגי סגור המבצע פעולה מוגדרת. פונקציה יכולה גם **לקבל** פרמטרים, לבצע עליהם פעולה מסוימת, ו**להחזיר** ערך כלשהו.

נגדיר פונקציה בצורה הבאה:

{תוכן הפונקציה} (פרמטרים) [שם הפונקציה] [ערך החזר][modifier]



**Modifier**

בחלק זה יצא לכם לגעת בהמשך, כרגע חשוב להכיר שלכל פונקציה נגדיר את "רמת הסיווג" שלה, כלומר איפה בקוד ניתן להשתמש בה. יתכן ותיתקלו בשלב זה בmodifier public.

בפונקציות שתכתבו בשלב זה תתקלו במילה static, כרגע נכתוב אותה בכל פעולה בצורה הזו ובהמשך הקורס נבין מה היא אומרת.



**פרמטרים**

הפונקציה יכולה לקבל משתנים בתור פרמטרים. היא יכולה להשתמש בהם ולבצע באמצעותם פעולות מסוימות – בדומה למה שיכולנו לבצע עד כה.

פונקציה יכולה לקבל פרמטר אחד, שניים, שלושה... כמה שנרצה. הפרמטרים יכולים להיות מטיפוסים שונים. (אחד int, השני char, השלישי double וכו').

* ניתן להשתמש בפונקציה שלא מקבלת פרמטרים כלל

**שם פונקציה**

על כל פונקציה להיקרא בשם ייחודי, באמצעותו נקרא לפונקציה ספציפית.  
ניתן לבחור בכל עבור פונקציות– תמיד נבחר שם בעל משמעות **המשקפת את תפקידה** של הפונקציה.

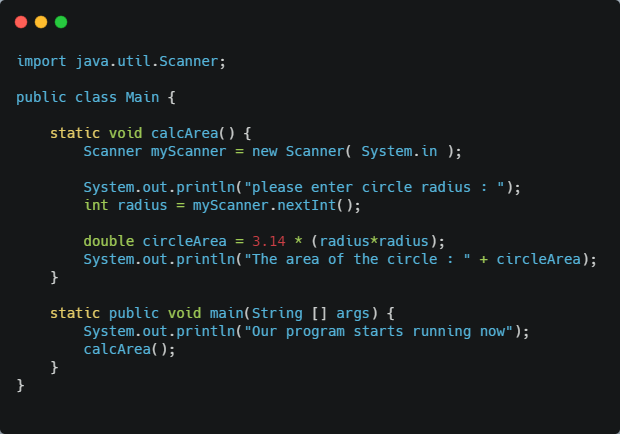
**תוכן הפונקציה**

בתוכן הפונקציה נכתוב את הפעולות אותן נרצה לבצע עבור כל קריאה לפונקציה.  
ברגע שנקרא לפונקציה שלנו, הקומפיילר אוטומטית ייכנס לתוכן הפונקציה ויבצע את הפעולות שהיא כוללת, שורה אחר שורה. לאחר שריצת הפונקציה תסתיים, הקומפיילר יחזור למקום הקודם ממנו הגיע לפונקציה.

**ערך החזר**

ניתן להחזיר ערך מסוים בסיום ריצת הפונקציה. בסוף ביצוע הפעולות הנכללות בפונקציה, ניתן להחזיר תוצאה מסוימת שככל הנראה משקפת את ביצוע הפעולות. שימו לב, בניגוד לפרמטרים של הפונקציה, ניתן להחזיר רק ערך החזר **אחד**.

**דוגמה לשימוש בפונקציה:**



בדוגמה זו הוגדרה פונקציה בשם "calcArea".

פונקציה זו לא מקבלת פרמטרים ולא מחזירה ערך החזר. לכן טיפוס ערך ההחזר הוא void (ריק) מילה שמורה המציינת כי לפונקציה אין ערך החזר כלל.  
הפונקציה קולטת ערך של רדיוס ולפי נוסחה מחשבת את שטח המעגל.

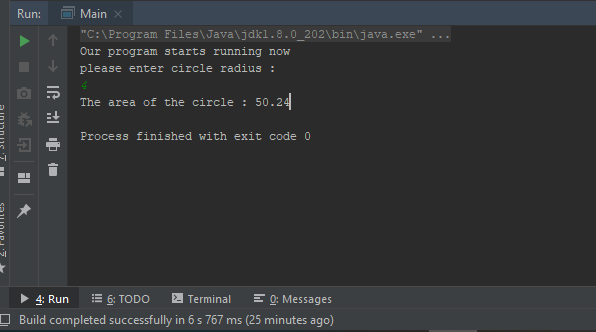
**ריצת הקוד תמיד תתחיל מפונקציה הmain!**

אז למה שנשתמש בפונקציות?

1. קריאות התוכנית (בלי פונקציית main ראשית מסורבל וקשה להבנה).
2. שימוש חוזר בקוד – ניתן להשתמש בפונקציה כמה פעמים במהלך הקוד. במקרה כזה צריך להוסיף שורת קוד אחת שקוראת לפונקציה וגורמת לה לפעול. במידה ולא היינו משתמשים בפונקציה, היה עלינו לרשום את אותו קטע הקוד בשנית אם היינו רוצים להשתמש בו במקומות שונים בקוד.
3. מאפשר עבודת צוות – כל חבר צוות יעבוד על פונקציה שונה.

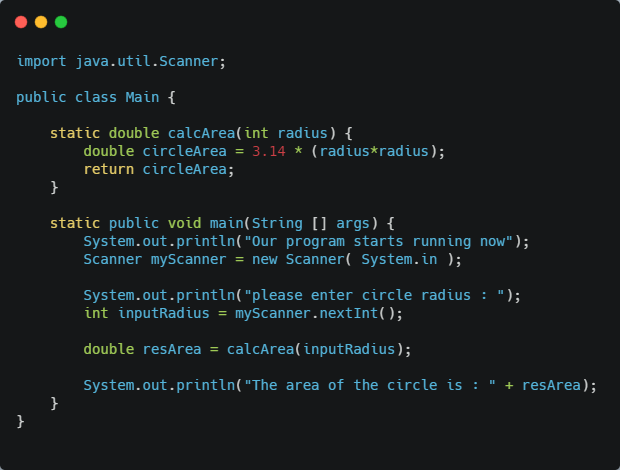
במידה ונריץ את הפרויקט, נראה שקודם כל תודפס ההודעה הראשונה הכתובה בmain, ורק לאחר מכן רצות תבוצענה הפעולות הנמצאות בפונקציה " calcArea".

לאחר ריצת הפונקציה חלון הconsole ייראה כך:



**קבלת פרמטרים בפונקציות**  
נגדיר את שם וטיפוס כל פרמטר, על מנת שנדע איך להשתמש בו בתוך הפונקציה עצמה.

**החזרת ערך**

נציין את **סוג המשתנה** בלבד ועל מנת לבצע החזר של אותו משתנה, נשתמש במילה השמורה return.  
[](https://carbon.now.sh/?bg=rgba(178,192,204,0.98)&t=seti&wt=none&l=text/x-csrc&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%20java.util.Scanner;%0a%0apublic%20class%20Main%20%7b%0a%0a%20%20%20%20static%20double%20calcArea(int%20radius)%20%7b%0a%20%20%20%20%20%20%20%20double%20circleArea%20%3D%203.14%20*%20(radius*radius);%0a%20%20%20%20%20%20%20%20return%20circleArea;%0a%20%20%20%20%7d%0a%0a%20%20%20%20static%20public%20void%20main(String%20%5b%5d%20args)%20%7b%0a%20%20%20%20%20%20%20%20System.out.println(%22Our%20program%20starts%20running%20now%22);%0a%20%20%20%20%20%20%20%20Scanner%20myScanner%20%3D%20new%20Scanner(%20System.in%20);%0a%20%20%20%20%20%20%20%20%0a%20%20%20%20%20%20%20%20System.out.println(%22please%20enter%20circle%20radius%20:%20%22);%0a%20%20%20%20%20%20%20%20int%20inputRadius%20%3D%20myScanner.nextInt();%0a%0a%20%20%20%20%20%20%20%20double%20resArea%20%3D%20calcArea(inputRadius);%0a%0a%20%20%20%20%20%20%20%20System.out.println(%22The%20area%20of%20the%20circle%20is%20:%20%22%20%2B%20resArea);%0a%20%20%20%20%7d%0a%7d%0a)

הפונקציה הבאה, מקבלת את המשתנה "inputRadius" הנקלט מהמשתמש דרך פונקציית הmain.

לאחר מכן מתבצע **זימון** (קריאה) לפונקציה, אליה מועבר בתור פרמטר המשתנה "inputRadius".

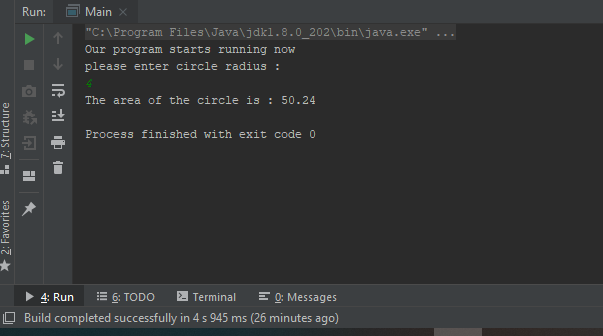
בתוך הפונקציה "calcArea" משתנה זה יקרא "radius".

לבסוף, ערך המשתנה “circeArea” מוחזר באמצעות פקודת return בתוך הפונקציה.



לאחר ריצת שורת קוד זו, המשתנה "resArea" יכיל את הערך שהפונקציה calcArea מחזירה.

פלט הערך המוחזר יהיה:



**פונקציות מובנות ב JAVA**

**מה הן פונקציות מובנות?**

פונקציות מובנות הן פונקציות שמגיעות עם השפה שלנו וניתן להשתמש בהן מבלי לכתוב אותן בעצמנו.

**למה שנשתמש בפונקציות מובנות?**

פונקציות מובנות נועדו על מנת להקל עלינו, המתכנתים, בתהליכים בסיסיים שעלולים לחזור על עצמם, כך שנוכל להשתמש בפונקציות פשוטות ונוחות במקום לממש דברים בעצמנו בכל פעם.

כמו כן, גם פעולות בסיסיות שיהיה מורכב לממש בעצמנו לעתים יונגשו בצורה פשוטה הרבה יותר בתור פונקציות מובנות.

**דוגמאות לפונקציות מובנות בשפה JAVA**

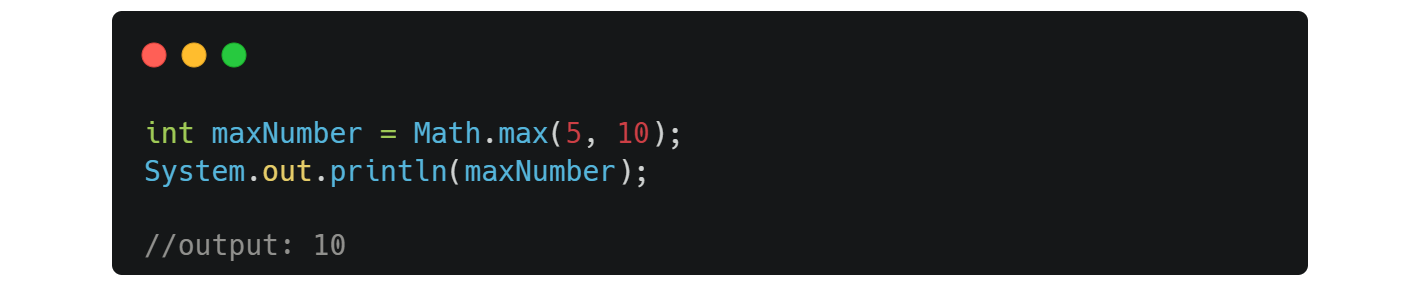
**הפונקציה charAt** – ניתן להפעיל על String, מקבלת אינדקס במחרוזת ומחזירה את התו במקום האינדקס



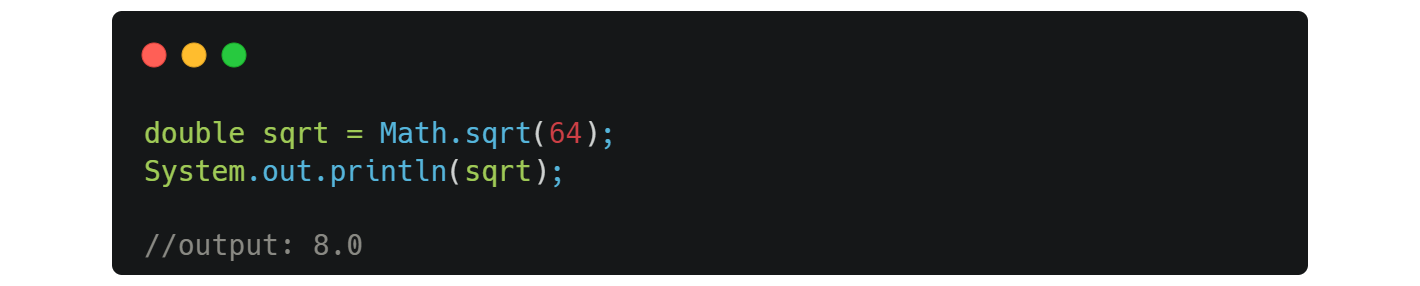
**הפונקציה length** – ניתן להפעיל על String, מחזירה את אורך המחרוזת

****

**הפונקציה max של המחלקה Math** – מקבלת שני מספרים ומחזירה את הגדול מבניהם

****

**הפונקציה sqrt של המחלקה Math** – מקבלת מספר ומחזירה את השורש שלו (כ double כיוון שיכולים להיות מספרים שהשורש שלהם לא שלם)

****

**כמובן שיש עוד המון פונקציות מובנות בשפה ששווה לקרוא עליהן ולהשתמש בהן**

על פונקציות נוספות הקשורות למחרוזות ניתן לקרוא [כאן](https://www.w3schools.com/java/java_ref_string.asp)

על פונקציות נוספות של המחלקה Math ניתן לקרוא [כאן](https://www.w3schools.com/java/java_math.asp)

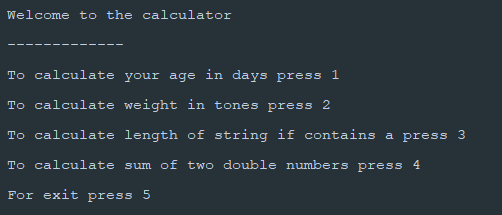
**תרגול מחשבון**

עלינו לכתוב תוכנית המדמה מחשבון!

המחשבון שלנו בכל פעם מציג תפריט למשתמש (כלומר את הפעולות שניתן לבצע במחשבון).  
המשתמש בוחר איזו פעולה הוא מעוניין לבצע, מכניס פרמטרים והתוכנית שלנו תחשב את התוצאה ותציג אותה למשתמש.

* התוכנית תדפיס בכל פעם את התפריט שוב ושוב מחדש למשתמש, וכאשר המשתמש יבקש לצאת, התוכנית תדפיס הודעה מתאימה והריצה תסתיים.

התפריט שיוצג למשתמש בכל פעם (עד אשר אשר המשתמש יבחר לצאת) ייראה כך:



בבחירת המספר 5 (יציאה מהתוכנית), יודפס למשתמש "Bye Bye".

**להלן פירוט על הפונקציות שיש לכתוב לצורך מימוש התוכנית:**

1. פונקציית main: תבצע את ניהול התוכנית מבחינת קבלת הקלטים והריצה כל עוד לא נקלט 5. (שימו לב! זוהי לולאת זקיף, קראו עליה בשיעור לולאות שוב אם יש צורך בכך)
2. פונקציית חישוב גיל המשתמש בימים. עלינו לכתוב פעולה המקבלת את גילו בשנים ומחשבת את גילו בימים. (בהנחה שיש 365 ימים בשנה).  
   הפעולה הזו לא מחזירה ערך החזר אלא מדפיסה את גילו של המשתמש בלבד.  
    תופעל בקבלת המספר 1 בתפריט.
3. חישוב משקל המשתמש בטונות. עלינו לכתוב פעולה המקבלת את משקל המשתמש בקילוגרמים. הפעולה תחזיר בערך החזר את משקלו בטונות **וגם** תדפיס את משקלו בטונות. תופעל בקבלת המספר 2 בתפריט.
4. פעולה שקולטת מהמשתמש את שמו המלא, ובודקת האם הוא מכיל בתוכו מחרוזת שמתקבלת כפרמטר. הפונקציה מדפיסה למשתמש את אורך המחזורת במידה והיא מכילה את המחרוזת שהתקבלה כפרמטר, אחרת, במידה והמחוזרת לא מכילה את המחרוזת שהתקבלה כפרמטר הפונקציה מדפיסה 999 (הפונקציה לא מחזירה ערך החזר).  
   תופעל בקבלת המספר 3 בתפריט, עם הערך "a". חתימת הפונקציה היא:

public static void doesStringContain(String substring)

1. פונקציה המקבלת שני פרמטרים מסוג double, ומחזירה את הסכום שלהם בחזקת 4  
   תופעל בקבלת המספר 4 בתפריט.

**בהצלחה!**