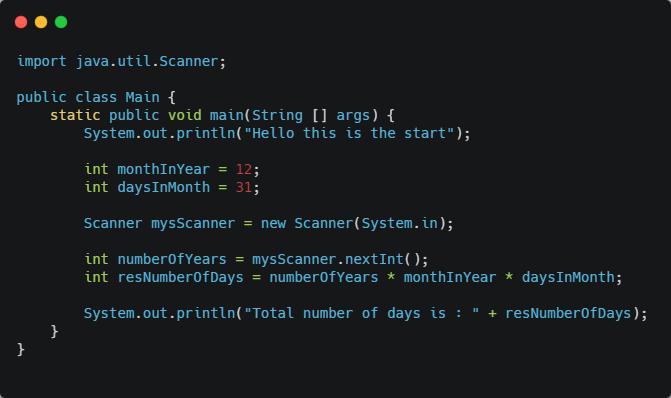
**לולאות**

עד עכשיו למדנו לכתוב קוד בצורה יחסית פשוטה:

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-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%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520this%2520is%2520the%2520start%2522)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520monthInYear%2520%253D%252012%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520daysInMonth%2520%253D%252031%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520mysScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520numberOfYears%2520%253D%2520mysScanner.nextInt()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520resNumberOfDays%2520%253D%2520numberOfYears%2520*%2520monthInYear%2520*%2520daysInMonth%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Total%2520number%2520of%2520days%2520is%2520%253A%2520%2522%2520%252B%2520resNumberOfDays)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)הקוד רץ בצורה כרונולוגית - 'מלמעלה למטה', שורה אחר שורה לאורך ריצת התוכנית, עד סוף קטע הקוד שלנו.

הבעיה מתחילה כאשר אנחנו רוצים לחזור על קטע קוד מסוים כמות מסוימת של פעמים – באמצעות הידע שברשותנו עד כה, נאלץ לכתוב את אותו קטע הקוד מחדש ובכך לחזור על קוד!

[Text

Description automatically generated](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-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%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)למשל, נניח ונרצה להדפיס ‘HelloWorld’ 5 פעמים, נכתוב את הפקודה System.out.println(“HelloWorld”) 5 פעמים!

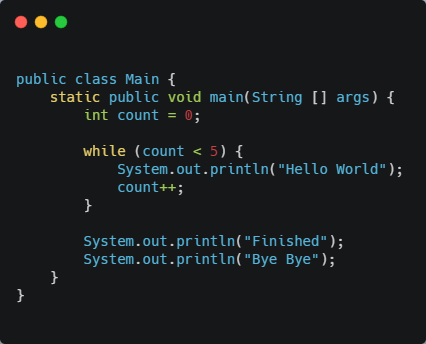
* מה יקרה אם נרצה להדפיס הודעה מסוימת **1000 פעמים**?
* מה יקרה אם נרצה להדפיס מספר פעמים לפי בחירתו של המשתמש? (לפי קלט מסוים).

בדיוק בשביל זה יש **לולאות.**

לולאה = קטע קוד החוזר על עצמו שוב ושוב, מספר רב של פעמים, עד שלב מסוים בו הלולאה נעצרת.

ישנם כמה סוגים של לולאות, אנחנו נלמד על שני הסוגים הנפוצים.

לולאות for ו-while

**[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-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=%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520count%2520%253D%25200%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520while%2520(count%2520%253C%25205)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520World%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520count%252B%252B%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Finished%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Bye%2520Bye%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)ללולאה מסוג while**

בקוד הזה תודפס המחרוזת  
‘Hello World’ 5 פעמים.

אומנם כתבנו את ההדפסה פעם אחת בלבד!

זאת מפני שהקוד אינו רץ מלמעלה למטה לפי הסדר, אלא בכל פעם הוא בודק האם התנאי בלולאת ה-while עדיין מתקיים - ואז חוזר על קטע הקוד שבתוך הלולאה מחדש.

* While == 'כל עוד'

כל עוד count < 5 הלולאה תתקיים

**איך נשתמש בלולאת while?**

1. נגדיר ערך מסוים בו נשתמש לקבוע את המצב בו נכנס/לא נכנס ללולאה, למשל, נגדיר משתנה count שיאותחל בערך 0.
2. לאחר מכן, נגיע ללולאת while ובתוך הסוגריים יוגדר 'תנאי העצירה' של הלולאה.
   * **במידה והתנאי מתקיים** (במידה ו-count קטן מ-5) – הקומפיילר נכנס לתוך הבלוק ומבצע את ההדפסה.
   * **במידה והתנאי אינו מתקיים** (במידה ו-count גדול או שווה ל-5) – הקומפיילר **מדלג** על הבלוק (סוגריים מסולסלים {}) של לולאת ה-while. והריצה ממשיכה כרגיל בסדר כרונולוגי.

**שימו לב!**  
הפקודה count++ הכרחית לביצוע הלולאה – פקודה זו מקדמת את המשתנה count ב-1.

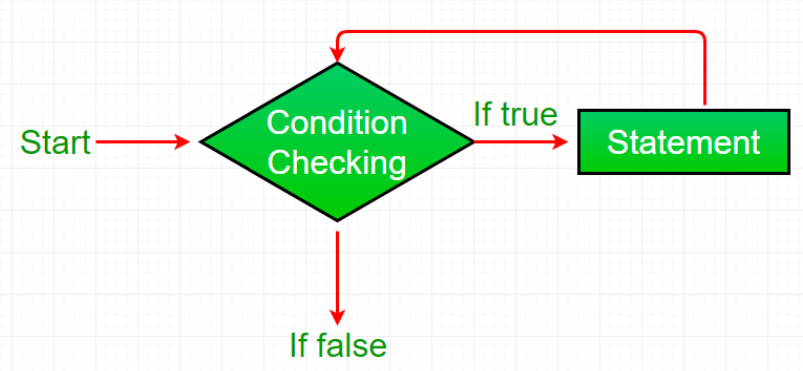
בשלב מסוים (לאחר 5 ריצות חוזרות של הלולאה) תנאי הלולאה כבר לא יתקיים וריצת הלולאה תסתיים.

כל עוד נתכנן את הקוד שלנו בצורה נכונה ונדאג להשתמש במבנה לולאות נכון, קטע קוד בלולאה יסתיים בשלב מסוים – נרצה להימנע ממצב בו לא השתמשנו במבנה לולאה נכון ונגרום למצב של **לולאה אינסופית**.

לולאה אינסופית – לולאה שתמיד תתבצע, התנאי עצירה שלה לעולם לא יתקיים

הסיבה לכך היא שלולאה אינסופית תפגע בריצה התקינה של הקוד ותמנע מהמשך הקוד שלנו לרוץ ותבצע את קטע הקוד שבתוך הלולאה פעם אחר פעם.

* האם לולאה אין סופית היא שגיאה? שגיאת זמן ריצה? שגיאת קומפילציה?  
  בדקו באינטרנט.



תרשים זרימה המתאר את מהלך לולאת הWhile

**פירוט ריצת הקוד של לולאת הwhile בדוגמה**

* המשתנה count מאותחל בערך 0.
* מתבצעת בדיקה האם count < 5, התשובה היא כן ולכן נכנסים ללולאה.
* מודפסת ההודעה: Hello World **פעם אחת** ומקדמים את המשתנה count, מכאן שערכו: count = 1.
* מתבצעת בדיקה האם count < 5, התשובה היא כן ולכן נכנסים ללולאה.
* מודפסת ההודעה: Hello World **פעם שניה** ומקדמים את המשתנה count, מכאן שערכו: count = 2.
* מתבצעת בדיקה האם count < 5, התשובה היא כן ולכן נכנסים ללולאה.
* מודפסת ההודעה: Hello World **פעם שלישת** ומקדמים את המשתנה count, מכאן שערכו: count = 3.
* מתבצעת בדיקה האם count < 5, התשובה היא כן ולכן נכנסים ללולאה.
* מודפסת ההודעה: Hello World **פעם רביעית** ומקדמים את המשתנה count, מכאן שערכו: count = 4.
* מתבצעת בדיקה האם count < 5, התשובה היא כן ולכן נכנסים ללולאה.
* מודפסת ההודעה: Hello World **פעם חמישית** ומקדמים את המשתנה count, מכאן שערכו: count = 5.
* בודקים האם count < 5, התשובה היא **לא** ולכן **יוצאים** מהלולאה
* ממשיכים כרגיל בהמשך ריצת התוכנית עד סופה.

[Text

Description automatically generated](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-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%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520aNumber%2520%253D%25200%252C%2520bNumber%2520%253D%25200%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520while%2520(aNumber%2520!%253D%2520-1%2520%2526%2526%2520bNumber%2520!%253D%2520-1)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Please%2520enter%2520A%2520number%2520(enter%2520-1%2520for%2520exit)%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520aNumber%2520%253D%2520myScanner.nextInt()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Please%2520enter%2520B%2520number%2520(enter%2520-1%2520for%2520exit)%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520bNumber%2520%253D%2520myScanner.nextInt()%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520int%2520sum%2520%253D%2520aNumber%2520%252B%2520bNumber%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522The%2520sum%2520is%2520%253A%2520%2522%2520%252B%2520sum)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522FINISHED%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Bye%2520Bye%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

מה מטרת קטע הקוד הבא? הריצו אותו בסביבת העבודה וגלו! (תדבגו את הקוד ותבינו מה מטרתו)

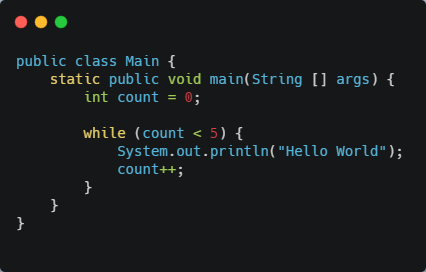
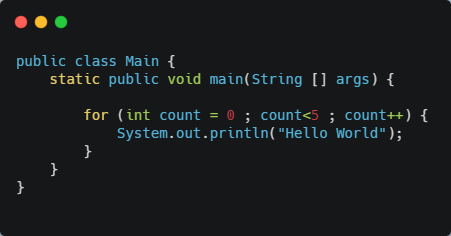
**לולאה מסוג for**

לולאה שנייה ונפוצה לא פחות היא לולאת הfor.

בלולאת for נגדיר את המשתנה של הלולאה בתוך כותרת הלולאה, ונקדם את המשתנה (++) גם בתוך הכותרת של הלולאה

* For == 'עבור כל'

עבור כל count שקטן מ5 הלולאה תתקיים

שתי הלולאות הנ"ל מדפיסות Hello World 5 פעמים, אחת while והשנייה for

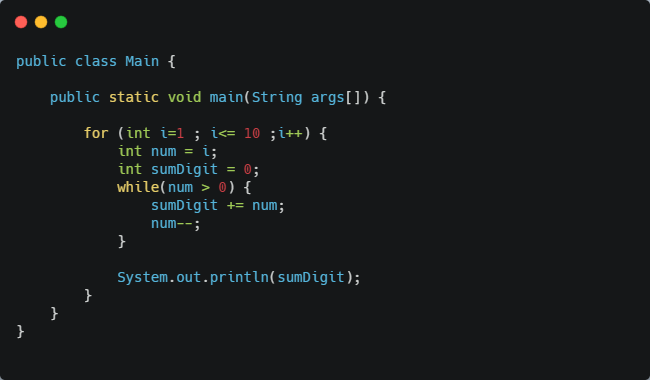
הפלט של שתי הלולאות יהיה:  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World

שימו לב שבדוגמה הנ"ל שימוש בלולאת for הינה נוחה ופשוטה יותר.

* כל לולאת for ניתן לפתור גם באמצעות לולאת while  
  שימוש בלולאת for יכול להיות לעיתים פשוט ונוח יותר במצבים מסוימים.

לולאות for היא סוג של לולאת while, למעשה מדובר בלולאת while עם מונה שמתקדם בצורה אוטומטית.  
כאשר מספר הפעמים לריצת הלולאה ידוע מראש, נעדיף להשתמש בלולאת **for**.  
כאשר מספר הפעמים לריצת הלולאה אינו ידוע מראש, או אינו תלוי במונה מסוים (למשל תלוי במשתנה בוליאני בלבד), נעדיף להשתמש בלולאת while.

**ניתן אף להגדיר לולאה בתוך לולאה! (לולאות מקוננות)**

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=public%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520%250A%2520%2520%2520%2520public%2520static%2520void%2520main(String%2520args%255B%255D)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%250A%2520%2520%2520%2520%2520%2520%2520%2520for%2520(int%2520i%253D1%2520%253B%2520i%253C%253D%252010%2520%253Bi%252B%252B)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520int%2520num%2520%253D%2520i%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520int%2520sumDigit%2520%253D%25200%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520while(num%2520%253E%25200)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520sumDigit%2520%252B%253D%2520num%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520num--%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(sumDigit)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%257D%250A%257D%250A)

(שאלת קליטה: מה התוכנית הבאה עושה? נסו להריץ בעצמכם ובדקו!)

[Text

Description automatically generated](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%250A%2520%2520%2520%2520public%2520static%2520void%2520main(String%2520args%255B%255D)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520grade%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520grade%2520%253D%2520myScanner.nextInt()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520while%2520(grade%2520!%253D%2520-1)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522The%2520grade%2520is%2520%253A%2520%2522%2520%252B%2520grade)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520grade%2520%253D%2520myScanner.nextInt()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522BYE%2520BYE%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)**לולאת זקיף**  
לעיתים נרצה להשתמש בלולאה המבצעת קלט – כל עוד הקלט עונה על תנאי הלולאה – כלומר הקלט שהתקבל לא שווה למספר מסוים.  
למשל, תכנית הקולטת ציונים של תלמידים עד קבלת הקלט 1-.  
במצב כזה נבנה לולאת זקיף במבנה הבא:

**לולאה עם משתנה בוליאני בתנאי עצירה**  
תנאי העצירה של הלולאה תמיד יהיה בוליאני.  
בתוך תנאי העצירה של הלולאה שלנו, ניתן להשתמש באופרטור חשבוני (<,>,= וכו') או בבדיקה של משתנה בוליאני (בדוגמא: isContinueLoop) המכיל תוצאה של תנאי כלשהו.

שימוש לב בדוגמה הבאה:

[Text

Description automatically generated](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%250A%2520%2520%2520%2520public%2520static%2520void%2520main(String%2520args%255B%255D)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520grade%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520boolean%2520isContinue%2520%253D%2520true%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520grade%2520%253D%2520myScanner.nextInt()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520if%2520(grade%2520%253D%253D%2520-1)%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520isContinue%2520%253D%2520false%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520while%2520(isContinue)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522The%2520grade%2520is%2520%253A%2520%2522%2520%252B%2520grade)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520grade%2520%253D%2520myScanner.nextInt()%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520if%2520(grade%2520%253D%253D%2520-1)%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520isContinue%2520%253D%2520false%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522BYE%2520BYE%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

האיטרציה הלולאה תיעצר רק כאשר הערך משתנה isContinue יהיה false.

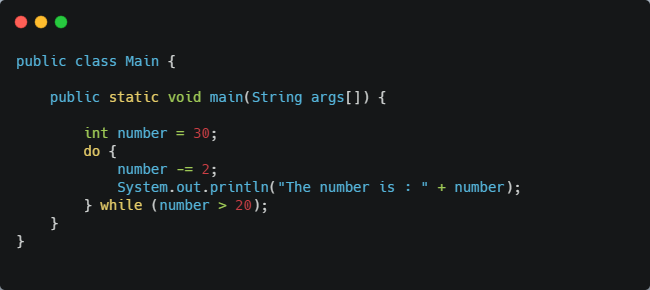
איטרציה – מעבר/ריצה של הלולאה

**Extra**

חשוב להדגיש שבקורס **לא נשתמש** בסוג הלולאות שנלמד עליהן בקטע זה, אך חשוב להכיר אותן לידע כללי.

Do While:

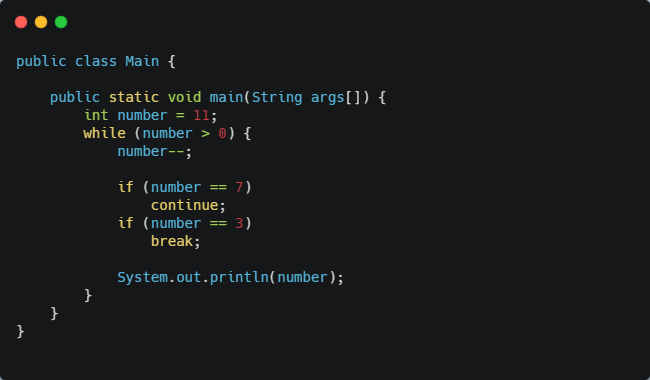
בלולאה מסוג Do While, ראשית תתבצע ריצת הלולאה ורק לאחר מכן תתבצע בדיקה האם התנאי מתקיים. במידה והתנאי מתקיים, הקוד שבתוך הבלוק (הקוד שבתוך הסוגריים המסולסלות) ימשיך לרוץ.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=public%2520class%2520Main%2520%257B%250A%250A%2520%2520%2520%2520public%2520static%2520void%2520main(String%2520args%255B%255D)%2520%257B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520number%2520%253D%252030%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520do%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520number%2520-%253D%25202%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522The%2520number%2520is%2520%253A%2520%2522%2520%252B%2520number)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%2520while%2520(number%2520%253E%252020)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

(מתי נרצה להשתמש בלולאה מסוג do while? בדקו באינטרנט!)

Continue/Break:

* פקודת **continue** מאפשרת לנו בצורה ישירה לקפוץ לאיטרציה הבאה של הלולאה.  
  ישנם מצבים בהם נרצה שהמשך הלולאה לא יתבצע ולקפוץ אוטומטית לריצה הבאה. במצבים אלו נשתמש בפקודת continue.
* פקודת **break** מאפשרת לנו לצאת מן הלולאה באמצע הריצה, גם אם התנאי עדיין מתקיים.  
  ישנם מצבים ספציפיים בהם נרצה לגרום לריצת הלולאה פשוט להיעצר באמצע, במצבים אלו נשתמש בפקודת break.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=public%2520class%2520Main%2520%257B%250A%250A%2520%2520%2520%2520public%2520static%2520void%2520main(String%2520args%255B%255D)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520number%2520%253D%252011%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520while%2520(number%2520%253E%25200)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520number--%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520if%2520(number%2520%253D%253D%25207)%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520continue%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520if%2520(number%2520%253D%253D%25203)%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520break%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(number)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%257D%250A%257D%250A)

(מה ידפיס קטע הקוד הבא? תריצו בעצמכם ובדקו!)

**סיכום**

לולאות הם מרכיב הכרחי וחשוב בכתיבת קוד שלנו, הן מאפשרות לנו לכתוב מערכות מורכבות יותר, לחסוך בקוד/עבודה מיותרת, ומאפשרות יכולות שלא היו אפשריות לפני כן ע״י מימוש בצורה סבירה.

**תרגול**

1. כתוב תכנית אשר קולטת מספרים מהמשתמש עד הקלט 999-, ומדפיסה את סכום המספרים שנקלטו  
   לדוגמה:

Enter numbers (-999 to stop):  
5  
6  
-999  
Sum: 11

1. כתוב תכנית אשר קולטת מהמשתמש מספר חיובי, ומדפיסה ריבוע כוכביות שצלעו כאורך המספר שנקלט. למשל עבור הקלט 5 יודפס:

Enter a number: 5  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

1. כתוב תכנית אשר קולטת מהמשתמש מספר חיובי גדול מ – 2 ומדפיסה האם המספר ראשוני או לא.  
   תזכורת: מספר הוא ראשוני אם הוא מתחלק ללא שארית רק בעצמו וב – 1, ולא באף מספר נוסף. למשל, 2 ראשוני - מתחלק רק בעצמו וב – 1, אולם 4 לא-ראשוני כיוון שמתחלק גם ב – 2.
2. כתוב תכנית אשר קולטת מהמשתמש מספר שלם וחיובי, ומדפיסה את האיבר המתאים בסדרת פיבונאצ'י. סדרת פיבונאצ'י היא סדרה בה שני האיברים הראשונים הם 1, וכל איבר אחר הוא סכום של שני האיברים הקודמים בו.  
   האיברים הראשונים בסדרה:  
   1, 1, 2, 3, 5, 8, 13, 21....  
   רמז: חישבו מה עליכם לעשות כדי להתמודד עם מספרים גדולים במיוחד.  
   השתמשו בטיפוס משתנה מסוג long.
3. חשב והדפס את לוח הכפל – כפולות המספרים מ1 עד 10

רמז – יש להשתמש בלולאה מקוננת (לולאה בתוך לולאה)

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100