

Class Notes

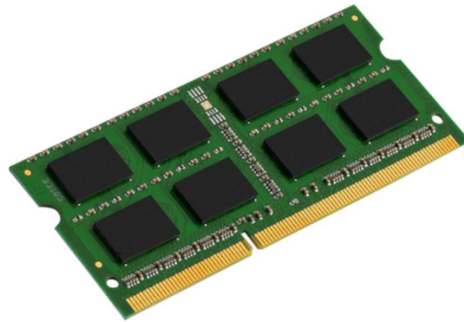
משתנים = תאי זיכרון.

משתנה אחד = תא זיכרון אחד.

RAM – Random Access Memory

זהו זיכרון שאנו כמתכנתים עובדים איתו ישירות.

אלו כרטיסי הזיכרון הירוקים



לצורך עבודה עם ה-RAM יש לפתוח תאי זיכרון.

תא זיכרון כזה נקרא "משתנה" – Variable.

כל תא זיכרון מכיל נתון אחד.

לדוגמה: שם פרטי.

לדוגמה: גיל.

לדוגמה: כתובת.

לכל תא זיכרון יש שם. השם חייב להיות משמעותי וייחודי.

קיימים חוקים לגבי חוקיות שמות משתנים:

א. שם משתנה יכול להכיל רק ספרות, אותיות באנגלית (קטנות או גדולות), קו תחתון.

ב. שם משתנה לא יכול להתחיל בסיפרא.

ג. שם משתנה לא יכול להיות כשם של מילה שמורה של שפת התכנות.

קיימות מוסכמות לגבי שמות משתנים:

א. שם משתנה חייב להיות משמעותי ולייצג את המידע שהוא מכיל.

ב. שם משתנה יתחיל באות קטנה, מילים מחוברות, כל מילה שנייה והלאה יתחילו באות גדולה.

דוגמאות חוקיות:

firstName, age, lastName, mobile, myBestFriendEver

דוגמאות לא חוקיות:

first name, 9age, last-name, super+girl

פקודת השמה

פקודה המאפשרת להכניס ערך למשתנה.

הערך יכנס לתוך תא הזיכרון.

```
firstName = "Moishe"
```

תמיד הערך שמימין יכנס לתא הזיכרון שמשמאל.

משמאל חייב להיות תא זיכרון

הפקודה הבאה לא חוקית:

```
"Moishe" = firstName
```

שירשור – Concatenation

הפעולה של לחבר מחרוזות עם עוד ערך ביחד.

לדוגמה מחרוזות עם מחרוזות.

לדוגמה מחרוזות עם מספר.

היא מבוצעת ע"י +

דוגמה:

```
"Hello" + " This is Cool!"
```

דוגמה נוספת:

```
"Red " + "Green " + "Blue " + "Yellow"
```

דוגמה נוספת:

```
"Red " + "Green " + "\nBlue " + "Yellow"
```

Hard-Coded

זהו מידע הקיים בקוד שלנו שאינו משתנה לעולם.

לדוגמה:

```
firstName = "Moishe"
```

לעולם בתוך המשתנה firstName תהיה המחרוזת "Moishe" וזה לא משנה כמה פעמים המשתמש יריץ את התוכנית.

אם המשתמש מכניס מידע לתוכנית שלנו, המידע הזה אינו Hard-Coded

סימן שמבצע פעולה ומחזיר תוצאה.

לדוגמה * זה אופרטור הכפל. הוא מבצע הכפלה בין שני מספרים ומחזיר תוצאת כפל ביניהם.

הערך שהאופרטור עובד עליו נקרא אופרנד.

אופרנד

הנפעל ע"י האופרטור

לדוגמה: $3 * 5$

3 ו-5 הם אופרנדים,

ה-* זה האופרטור (אופרטור הכפל) שמבצע מכפלה ומחזיר 15.

אם תהיה פעולת השמה, היא תקבל 15. לדוגמה: $result = 3 * 5$

ישנם מספר סוגי אופרטורים.

אופרטורים מתמטיים: +, -, *, /, **, %

אופרטורי השוואה: >, <, <=, >=, ==, !=

אופרטורים לוגיים (נגיע בהמשך)

ויש עוד...

מספר האופרנדים:

יש אופרטורים שפועלים על אופרנד אחד. אופרטור כזה נקרא אופרטור אונרי.

דוגמה:

```
result = -3
```

```
num = 8
```

```
result = -num
```

יש אופרטורים שפועלים על שני אופרנדים. אופרטור כזה נקרא אופרטור בינארי.

דוגמה:

```
num1 = 10
```

```
num2 = 20
```

```
result = num1 * num2
```

יש אופרטור אחד ויחיד שמסוגל לפעול על שלושה אופרנדים!!

הוא נקרא אופרטור טרינארי Ternary Operator =

(נלמד בג'אווה סקריפט)

אופרטור המודולו – Modulo

זהו אופרטור השארית.

הוא מחשב ומחזיר את השארית שתישאר בחלוקה של שני מספרים.

דוגמה:

result = 4 / 3 → result is 1.3333333333

result = 4 % 3 → result is 1

שיטת החישוב:

$$4 \% 3 = ? \quad 4 / 3 = \frac{4}{3} = 1\frac{1}{3} \quad \text{This is the Modulo}$$

שבר שבר
מדומה ממשי



$$6 \% 5 = 1$$

$$7 \% 2 = 1$$

$$10 \% 3 = 1$$

$$12 \% 6 = 0$$

$$4 \% 2 = 0$$

$$22 \% 7 = 1$$

$$11 \% 3 = 2$$

0 עבור מספר זוגי, 1 עבור מספר אי-זוגי חיובי = num % 2

סיפרת האחדות = num % 10

$$21 \% 10 = 1$$

$$38 \% 10 = 8$$

$$50874 \% 10 = 4$$

משפט תנאי

פקודה הבודקת תנאי בולאני כלשהו.

תנאי בולאני זהו ביטוי השווה ל-true או ל-**false**, כלומר הוא או נכון, או לא נכון.

לדוגמה, הביטוי $4 > 3$ הוא תמיד true.

לדוגמה, הביטוי $num > 3$ יהיה true רק אם num אכן יותר גדול מ-3. אם num יהיה שווה 3 או פחות, הביטוי יהיה false.

משפט תנאי זוהי פקודה שבודקת תנאי בולאני ואם התנאי שווה ל-true היא מבצעת סדרת פקודות א', אם התנאי שווה ל-**false** היא יכולה לבצע סדרת פקודות ב'.

אופרטורי השוואה: $==$, $!=$, $<$, $>$, $<=$, $>=$.