

תנאים ב TypeScript

נושאי השיעור

- (1) תנאי מקוצר
- (2) תנאים מקוננים
- (3) switch
- (4) לולאות

תנאים מקוננים

Nested condition - Condition within condition.

אופרטור טרינארי

```
//ternary operator
let grade: number;
let smiley: string;

grade = Number(prompt("Enter your grade: "));

smiley = grade >= 60 ? ":-)" : ":-(";

document.write(smiley);
```

switch

במהלך החיים ניפגש אין ספור פעמים בתפריטים שונים במגוון צורות, לדוגמה התפריט שמוצג בכספומט שכולנו נעזרים בשירותיו פעמים רבות.

בתפריט הראשי כשנכניס את הכרטיס לכספומט יופיעו אופציות שונות:

- הפקדת כסף
- משיכת כסף
- בירור יתרה
- העברה בנקאית
- יציאה מהמערכת

בכדי ליצור את תרשים הלוגיקה של התפריט, אנו צריכים לקלוט את הבחירה של המשתמש, לבדוק את ערכה ולפעול בהתאם.

ניצור תכנית קצרה של התפריט בעזרת מנגנוני ההתניות:

```

let num: number;

num = Number(prompt("5-ל 1 בין מספר הכנס"));

if (num == 1) {
    document.write("משיכה");
}
else {
    if (num == 2) {
        document.write("יתרה ברור");
    }
    else {
        if (num == 3) {
            document.write("ק'צ הפקדת");
        }
        else {
            if (num == 4) {
                document.write("מזומן הפקדת");
            }
            else {
                if (num == 5) {
                    document.write("יציאה");
                }
                else {
                    document.write("5-ל 1 בין רק מספר הכנס! שגיאה")
                }
            }
        }
    }
}
}

```

התכנית הזו היא מסובכת יחסית, ובנויה בצורה שמקשה על הקריאות וההבנה של הקוד. משפט התנאי במבנה הזה נקרא גם משפט רב-בררה, כיוון שנותן לברור ביצוע של קטע ייעודי אחד בתכנית מבין כמה קטעי בררה אפשריים.

לכן, כאשר ניתקל במקרה כזה בו אנו פועלים בהתאם לערך משתנה מסוים, נוכל להשתמש במנגנון קריא יותר וקצר יותר, לשם בדיקת תוכן הערך. המנגנון הזה הוא מנגנון ה-switch שמורכב משתי מילות מפתח: 1. **switch** שלידו נרשום את שם המשתנה אותו אנו רוצים לבדוק 2. **case** כל case מייצג ערך מסוים ואחריו פקודות שיבוצעו רק במקרה שבו תוכן המשתנה שנבדק על ידי ה **switch** שווה ל case הנוכחי

כלומר נרשום מספר **cases** לפני מספר האופציות שנרצה לבדוק האם המשתנה שאנו בודקים ב- **switch** שווה להם.

```

let num: number;

num = Number(prompt("5-ל 1 בין מספר הכנס"));
switch (num) {
    case 1: document.write("משיכה");
            break;
    case 2: document.write("יתרה ברור");
            break;
}

```

```

case 3: document.write("ק'צ הפקדת");
        break;
case 4: document.write("מזומן הפקדת");
        break;
case 5: document.write("יציאה");
        break;
default: document.write("5-ל 1 בין רק מספר הכנס! שגיאה");
        break; // חייבים לא break אחרונה כפקודה
}

```

עכשיו כשיש לנו את שתי הדוגמאות של מבנה הקוד כתובות אחת ליד השנייה, אנו יכולים לראות בבירור את היתרון בקריאות של switch מול if-else, ובקיצור המשמעותי באורך הקוד שמנגנון ה-switch מספק לנו. זו הסיבה הברורה לכך שנעדיף להשתמש בswitch בכל סיטואציה שבה המקרה מתאים לתבנית המסוימת ש-switch נותנת עבורה מענה.

אפקט מפל המים

מפל מים מופיע לרוב בצורה כזו שנקודת ההתחלה שלו נובעת ממקום מסוים בהר, ומאותה הנקודה יזרום עד לתחתית ההר.

המחשב עובד ב-switch באותה צורה:

נקודת ההתחלה תהיה ה-case המתאים לבדיקה, ומאותה נקודת התחלה הוא יתחיל לבצע את הפעולות ולזרום עד למטה- כלומר יבצע את כל הפקודות עד סיום הקטע של ה-switch.

האפקט הזה הוא אפקט שאנחנו צריכים בכל פעם לשקול האם הוא מתאים לנו למקרה הספציפי שלנו או לא. יש מקרים מסוימים ש"מפל המים" מהווה יתרון, כגון האפשרות לתת לערכים מרובים לבצע את אותו הקוד (ולמנוע צורך של שיכפול קוד) אך לא ניכנס למקרים אלו בשלב זה.

לרוב- "אפקט מפל המים" לא יהיה האפקט שנרצה לקבל בעת כתיבת ה-switch, ולשם כך בנוסף למילים **switch** ו-**case** נכיר גם את ה-**break**.

Break - משמעותו שבירה, כלומר יש באפשרותנו להגדיר באופן יזום נקודת סיום אחרת לביצוע הפעולות, מנקודת ברירת המחדל שתהיה תמיד בתחתית הקטע. אחרי כל case שנרצה שבסיום ביצוע הפעולות שלו קטע ה-switch יסיים את פעולתו, ולא ימשיך ל-cases הבאים, נוסיף את המילה break, שמהווה מעין צינור שמעביר אותנו ישירות אל נקודת הסיום, בלי לעבור דרך כל הפקודות הנוספות שבהמשך ה- **switch**.

כללים חשובים ב: **switch**

- (1) default - הוא case שיתבצע כברירת מחדל אם הערך שאנו בודקים לא התאים לשום case אחר.
אם לא נרצה שיתבצע קוד במקרה בו המשתנה שאנו בודקים לא שווה לאף אחד מהcases אז אין צורך לרשום את default.
- (2) לא חובה לרשום את break אחרי ה-default.

let hour: number;

```
hour = Number(prompt("שעה הכנסת 0 ל-23"));

switch (hour) {
  case 9: document.write("בוקר ארוחת");
  break;
  case 12: document.write("צהריים ארוחת");
  break;
  case 18: document.write("ערב ארוחת");
  break;
  default: document.write("בניימ ארוחת");
}
```