

מאפשר ליצור שם משתנה שני לשם משתנה מסוים ואיתו אפשר להשתמש כמו במשתנה reference 1.
(המקורי כרגיל) (גישה ושינוי הערך)

מייצגים עם &

הפונקציה ניגשת למשתנה המקורי ולא אל העותק שלו כך, reference כשמעבירים לפונקציה משתנה עם 2.
אפשר לשנות ערך ישירות מהפונקציה

מאפשר להפחית את כמות ההעתקה שקוראת מאחורי הקלעים - במקום שהעברת המשתנים תגרום
להעתקה של הערך לארגומנט הפונקציה מקבלת "הפניה" למשתנה המקורי

בריפרנס משתמשים במשתנה כמשתנה רגיל ולא צריך לגשת אל הערך עם * כמו בפוינטר, כשיוצרים 3.
ריפרנס חייב לקשר אותו למשתנה קיים ובפוינטר אפשר ליצור פוינטר ריק, ופוינטר צריך להקצות לו זיכרון
בזמן שלריפרנס לא מקצים זיכרון

ריפרנס יותר בטוח בגלל שלא מקצים זיכרון בשבילו בניגוד לפוינטר וכך לא יכול לקרות מצב של זליגת זיכרון
או שאפשר לשכוח לשחרר את הזיכרון

נכון – בגלל שבפונקציה הארגומנט השני הוא ריפרנס צריך להישלח אל הפונקציה משתנה 4.

ולא הארגומנט עלא נכון – לפונקציה נשלח הכתובת של

לא נכון – צריך להישלח לפונקציה לארגומנט השני משתנה ולא ערך

בפונקציה הראשונה המשתנה לא קיים אחרי הפונקציה זאת אומרת שולחים רפרנס למשתנה שלא יהיה 5.
קיים אחרי סיום הפונקציה

בפונקציה השנייה לא משחררים את הזיכרון של המערך הדינמי