git הינה מקום לשמור ולשתף קוד בצורה נוחה ויעילה יותר .  
היתרנות שיש בgit הינם :   
חלוקות העבודה ל commites כלומר העבודה עצמה מתבצעת על ידי חלוקה לזמנים שונים בין הגרסות הנקראים commites   
יתרונות בכך הינם חזרה בכול רגע נתון לגרסה קודמת , ניתן גם לחלק עבודה לענפים שונים בין כמה עובדים   
נקרא branch למעשה כול אחד עובד בעולם משל עצמו ולאחר מכן משלבים בין העולמות של כולם לקבלת מימד אחיד בין כולם

(2  
:Repository  
התיקייה המרכזית המשמשת לכלל קבצי git ניתן דרך הRepository להגיע לכול הגירסות הקודמות לקבל ממנו מידע וכדומה .  
למעשה מתעד את כול מה שעשינו דרך הgit

3)

Branch:   
ענף שבו כול אחד מקבל חלק מהעבודה של הפרויקט ועובד עליו על למיזוג יחד עם הגרסה הגדולה (בדרך כלל של כולם ) , רצוי להשתמש בBranch אישי ולא לעבוד על הגרסה הראשית ((dev\master  
בשביל לשמור גרסה טובה לכולם ללא שגיאות .  
כול Branch מחולק לכמה commites עד למיזוג .  
שימוש מסודר בBranch יתן סדר בקוד ,וימנע כאוס , ובזבוז זמן מיותר בקוד  
  
4)  
Conflict:   
קונפילקט כשמו כן הוא מצב שבו יש "מריבה" בין שני הצדדים הפעם על פיסת קוד , כלומר צד אחד מעונין בקטע או חלק מסוים בקוד ואילו חברו לחילופין לא מעונין בכך , כך שלמעשה הם רוצים דברים שונים זה מזה   
על ידי Conflict ניתן למצוא הבדלים משמעותיים בקוד   
איך פותרים Conflict ?   
על ידי הסבר מדוע השורה או הקטע צריכה להיות בתוך קטע הקוד , או לחילופין מדוע לא נכון שהחלק המדובר יהיה בתוך קטע הקוד הגדול של הפרויקט   
ישנם מצבים שבהם שני הצדדים יפתרו את הConflict שלהם יחדיו עד לקבלת גרסה אחידה המסוכמת על ידי הצדדים השנויים במחלוקת .  
  
5)   
Head  
סימון בGit ל commites מסוים , ניתן לחוזר אחורה לcommites על ידי שימוש בHead חיסור , הHead יושב על הcommite הנוכחי .

6) Remote:   
Repostory Remote :   
רפוזתירי שנמצא על שרת במקום אחר , ואין לי גישה אליו ( לדוגמה עבודה מרחוק ) וכול אחד ניגש אליו למקום אחר , ניתן לשמור עליו ולעשות לו בקשות של pull

(7 Pull Request  
 הורדת תוכן של בקשה -Branch מסוים אליך ,  
כלומר מוריד תוכן מהשרת לBranch עליו אנו עובדים (צריך בירור בהבנה מה התוכן שיורד אליך מהשרת )

8)  
COMMIT - יצירת זמן מסוים "עד ל " ניתן גם לשלוח הודעה קצרה בנוסף ,   
push - העלה של החומר שעליו עבדנו ל Branch Remote  
pull- - הורדה של התוכן הקיים בתוך הBranch Remote  
pr- בקשה שבו תורם לקוד מבקש מאחראי הפרויקט למיזוג הקוד של התורם לקוד הגדול : pull request

- fetch הורדת כול התוכן שיש מהRpository כולל כול הBrancים הקיימים , וכן קבצים נוספים   
rebase- שינוי מתוך הorigin של הbranch הנוכחי לתוך Branch אחר כלומר הוא מקבל את כול המידע שלא היה קיים לו בזמן שאדם אחר עשה Commit ומעדכן את כול המידע בתוך הBranch שבו אנו נמצאים

stash—-למחוק את כול הנתונים שבצעו מאז הcommit האחרון (אצלי( local ולחזור אל הcommit האחרון בעצם אנו יוצרים גירסה חדשה אצלנו שאנו יכולים לחזור אליה בכול פעם שאנו נקלעים לצרות .

add—מוסיף קובץ ,לרשימת הקבצים שצריכים לעלות בcommit הבא שנעשה .לפני ההעלה לכולם -dev -

rebase -i מאחדים את הcommits שיש לנו יחדיו , על מנת לשמור על סדר בין כולם ושלא יהיה לנו יותר מידי commits שיעשו לנו בלאגן , אז כול commits מתאחדים ביניהם .לדוגמה עם היה לנו 5 לאחר מכן כולם יהיו 1 וממילא כול ההודעות הם אחד וכולם מחוברות ביניהם (כנראה לא ניתן לחזור אחורה לאחר פקודה זו )

9)   
rebase-i

עץ השינויים יהיה אחיד ולא נוכל לחזור אחורה בגרסה , ונקבל את כול המידע שהיה לנו חסר בזמן שעבדנו על הלוקל שלנו , השימוש בפקודה נותן לנו שהכול אחיד ונחשב להיות commit אחד   
מסודר יותר , וכך נמנע commitים מיותרים -לא רצוי

הסכנה ניתן לאחד commit טוב יחד עם commit רע .