

creational		Structural		Behavioral	
תבניות שקשורות לאופן יצירת אובייקטים		תבניות שקשורות למבנה של האוביקטים		תבניות שקשורות להתנהגות של האוביקטים	
		Adapter		Chain of Responsibility	
singleton		תבנית עיצוב שמאפשרת לאובייקטים מסוגים שונים לעבוד בשיתוף פעולה.	השימוש:	מאפשר להעביר בקשה לאורך רצף של מטפלים. עם קבלת בקשה, כל מטפל מחליט אם לטפל בבקשה או להעביר אותה הלאה למטפל הבא ברצף	השימוש:
כאשר נרצה ליצור מופע אחד מסוג המחלקה	השימוש:	מתאם לשקע – מסוג אמריקאי לסוג אירופאי	דוגמא מהאתר או מהשב:	request piepline, נתינת הנחה בחנות	דוגמא מהאתר או מהשב:
יש מקום של הרבה דלתות וכן אדם אחד פותח את כולם	דוגמא מהאתר או מהשב:				
factory		composite		command	
כאשר נרצה ליצור אובייקט, וסוג האובייקט ייקבע לפי פרמטר מסוים. הפרמטר יכול להגיע מכיוונים שונים – סוג מערכת הפעלה, זמן ביום או קלט ממשתמש מסוים.	השימוש:	תבנית המאפשרת לך לחבר אובייקטים למבני עצים ולאחר מכן לעבוד עם מבנים אלה כאילו היו אובייקטים בודדים, גם אם הם עצמם מסתעפים.	השימוש:	הגדרת אוביקט שמייצג פעולה לביצוע האובייקט כולל את הנתונים הדרושים לביצוע הפעולה ואת הפעולה עצמה, הגדרת האוביקט מאפשרת הפרדה בין הגורם שמזמין את ביצוע הפעולה לבין הגורם שמבצע אותה בפועל וכן העברת הפעולה כפרצטר לפונקציה	השימוש:
שליחת דבר דואר במשימת/אוויה/מטוס לפי המיקום	דוגמא מהאתר או מהשב:	תיקיות שבתוכם יש קבצים ותיקיות	דוגמא מהאתר או מהשב	המלצר שמעביר פתקים לטבח, מניות	דוגמא מהאתר או מהשב:
Abstract Factory				commands ומאחסנים בתור כאשר receiver מליט מתי לבצע את command	נוסף
נרצה לייצר מספר אובייקטים ששייכים לאותה משפחה	השימוש:	Decorator		iterator	
ייצרנו מחלקות של רהיטים מודרניים וישנים	דוגמא מהאתר או מהשב:	קיים אובייקט כלשהו. שיכול לקבל תוספות. התוספות אינן משנות את האובייקט עצמו, אלא מוסיפות לו יכולות – משדרגות אותו.	השימוש:	אוביקט שמשתמש למעבר על איברי קבוצה ולגישה אליהם	השימוש:
Builder		פיצה עם סוגי תוספות	דוגמא מהאתר או מהשב		דוגמא מהאתר או מהשב:
כאשר נרצה ליצור מהתחלה אוביקט מסוים (בתהליכים) כאשר התהליך הוא משתנה	השימוש:				
יצירת סוג רכב לפי בחירה אישית	דוגמא מהאתר או מהשב:	Facade		state	
		הממשל להסתיר את המורכבות של מערכת מורכבת מן המשתמשים	השימוש:	כאשר	
Prototype-העתקת אובייקט מסוים עם כל הנתונים שלו – לתוך אובייקט חדש.		תיקון רכב נחשב לתהליך מורכב הכולל מספר שלבים והוצאת מספר רב של רכיבים אתה פשוט רק שולח למוסך בלי להבין איך מתקנים זאת	דוגמא מהאתר או מהשב		
כשייצרת אובייקט היא יקרה מבחינת משאבים – זמן ריצה, זכרון וכו *כשאובייקט נמצא במצב מסוים ורוצים העתק של המצב לפני ביצוע פעולה שמשנה את האובייקט *כשאובייקט נמצא במצב שנדרש תהליך כדי להגיע אליו, רוצים לחסוך את התהליך *חסכון	השימוש:			observer	
כשרוצים ליצור 2 צורות זהות	דוגמא מהאתר או מהשב:				
הערות: factory or Builder? i builder שניהם בסופו של דבר מייצרים אובייקטים שונים. factory מייצר אובייקט על פי בחירה מסוימת, מחזיר אותו להמשך פעילות. לעומת זאת, ב builder מייצרים מוצר סופי כלשהו, בתהליך, באמצעות סדרת פעולות שזהה בכל המוצרים		Flyweight			
		שימושי כאשר מתמודדים עם מספר רב של אובייקטים עם אלמנטים חוזרים ונשנים פשוטים שישתמשו בכמות גדולה של זיכרון אם יאוחסנו בנפרד ולכן נחזיק נתונים משותפים ולא נעשה כל פעם הכל מחדש	השימוש:		
		האליפסות בשעור	דוגמא מהאתר או מהשב		
		Proxi			
		מיופה כח, נמצא: לפני השירות שאליו רוצים להגיע ומחליט מה לבצע. לפעמים יעביר הלאה את הבקשה למישהו נוסף ולפעמים יטפל בעצמו. – ולכן הוא נקרא מיופה כח, כי הוא יכול לבצע פעולה של מישהו אחר, כביכול.	השימוש:		
		InternetProxy	דוגמא מהאתר או מהשב		