

סילבוס אנגולר 19

הקורס המקיף מהיסוד ועד לרמת בכיר

פרק 1:

צעדים ראשונים בעולם של אנגולר 19

1. התקנה או שידרוג לסביבת Angular 19
2. קצת רקע והכנה: יצירת פרויקט חדש מבוסס אנגולר 19
3. ארכיטקטורה בסיסית ורזה יותר בגירסת 19 – ללא הקובץ `app.module.ts`
4. קומפוננטת standalone – הצעד הראשון באפליקציה
5. איך לארגן קומפוננטה בקובץ אחד – מה שהיה גם בגרסאות קודמות
6. עוד קצת רקע באנגולר כללי: `css` גלובלי לכל המערכת לעומת `css` לכמה קומפוננטות
7. רקע בסיסי למתחילים: הגדרת משתנים והצגתם על המסך

פרק 2:

תקשורת והעברת מידע

1. שימוש ב `ngModel` – באנגולר 19 מקבל הצהרה נפרדת לכל קומפוננטת standalone
2. התממשקות standalone עם קומפוננטות נוספות
3. התקדמות לעבודה עם `signal` מסוג `input()`
4. הרחבה על `signal` מסוג `input` – הגבלות ותנאים
5. רקע למתחילים: אירועי מקלדת בסיסיים באנגולר
6. המשך אירועי מקלדת למתחילים

פרק 3:

עיצוב דינמי – שימושי יותר שיטת האובייקטים

1. `ngClass` – עיצוב דינמי בייסיק
2. העברת אובייקטים ל-`ngClass` – קונספט שימושי יותר בגרסאות מתקדמות
3. דרך נוספת להגדיר מחלקות – שימוש ב-`Directive Class`
4. למתחילים: `ngStyle` בסיסי – עיצוב דינמי של `style`
5. `ngStyle` – שליטה באמצעות אובייקט, קונספט שימושי בגירסת אנגולר 19
6. `Attributes` – קישור משתנים, מה שהיה גם בגרסאות קודמות

פרק 4:

עבודה עם לולאות, תנאים ועוד – שידורים

1. שידור ל- `@empty`, `@for` ועבודה עם `track`
2. היכולות של `@let`
3. מעבר לעבודה עם `@switch` `@if` `@else`
4. שימוש בתבנית `output()` להעברת מידע – הפסקת שימוש ב-`@Output`
5. עבודה עם `ng-content` לשיתילת `Html` דינמי, כמו בגרסאות קודמות

פרק 5:

התאמה אישית של קומפוננטות

1. שליטה בתגית של הקומפוננטה עצמה `Host` – דרך אובייקט
2. קוסטמיזציה מותאמת אישית בשילוב `standalone`
3. יצירת רכיב תשתיתי בשילוב `standalone`

פרק 6: שידור משמעותי

עבודה עם `Signal` ונרחיב על `ChangeDetection`

1. הכרת מערכת `Signal` החדשה
2. הכח של שימוש ב- `computed` ויכולת הסרת השימוש ב-`Observable`
3. רקע לפני צלילה: הכרת `map` ו- `reduce`
4. עבודה מורכבת עם `Signal` – ניהול מערכים ותלויות
5. אופטימיזציה – העמקה על `changeDetectionStrategy` בשילוב `signal`
6. המשך והרחבה על `doCheck` במקרים של `onpush`, `default`
7. עבודה עם טיימרים ועדכונים מושהים
8. העמקה נוספת וסיכום על `changeDetectionStrategy`

פרק 7:

טכניקות מתקדמות לגילוי שינויים

1. חקירת הצורך בשימוש ב-`markForCheck`
2. מתקדמים בניהול עדכונים ע"י `detach`, `reattach`
3. ביצועים גבוהים – עבודה מחוץ ל-`ngZone`

פרק 8:

ניתוב ומעברים בין עמודים

1. הגדרות ניתוב – הארכיטקטורה החדשה באנגולר 19
2. ניתוב מתוך רשימה אל הפרטים – דומה לגירסאות קודמות
3. הרחבה על ניתוב – שימוש ב `routerLinkActiveOptions` ועוד
4. ניתובים מקוננים ושימוש ב-`childrens`
5. `Guard` – אבטחת מידע: טיפול בהרשאות

פרק 9:

העברת מידע בין עמודים ושליטה בניתוב

1. העברת פרמטרים בניתוב – עבודה עם `params`, `queryParams` ו-`inject`
2. `Params` מול `ParamMap`
3. אירועי ניתוב: `NavigationCancel`, `NavigationEnd` ועוד
4. `Fragment` – ניתוב לחלק של העמוד ולא לקומפוננטה אחרת
5. טכניקות מתקדמות לניהול פרמטרים – שימוש ב-`match`

פרק 10:

`formControl`, `formGroup` באנגולר 19

1. `FormControl` – יצירה לפי גירסת אנגולר 19
2. עבודה עם `FormGroup`
3. היררכיה של `FormGroup` באמצעות `FormBuilder`
4. הרחבה על ולידציה
5. טפסים בזמן אמת – יצירה, הוספה והסרה של שדות בזמן אמת

פרק 11:

שליטה בטפסים ופעולות

1. טפסים מונחי תבנית `Template Driven Forms`
2. מעקב אחר שינויים בטופס `dirty`, `touched`, `get`
3. ולידציות מורכבות – יצירת חוקים מותאמים אישית

פרק 12: עבודה עם Accessors ועל טפסים דינמיים

1. יצירת Accessor באנגולר
2. הטמעת Accessors בטפסים וחקירת השימוש
3. עבודה מתקדמת עם FormBuilder ומערך של טפסים

פרק 13: עבודה עם HTTP

1. ביצוע קריאות GET באמצעות HttpClient
2. ביצוע קריאות Post באמצעות HttpClient
3. קבלת פרמטרים ע"י ActivatedRoute
4. קריאות API - הרחבה

פרק 14: services והזרקת תלויות באמצעות inject

1. יצירת service והזרקת תלות באמצעות Inject
2. חיבור בין קומפוננטות - איך service מעביר מידע?
3. דרכים נוספות להזרקת שירותים באמצעות useFactory, useValue
4. שימוש ב-useExisting - מתי ולמה זה חשוב?

פרק 15: הרחבה על services

1. תלויות ב-services בארכיטקטורה של standalone
2. הזרקת services - איך עובדים עם Injectable ו-useClass
3. היררכיה בקומפוננטות - שימוש ב-providers לעצירת חילחול המופע
4. שימוש ב-service מחוץ לאיזור הקומפוננטה באמצעות inject