

Övningsuppgift 6

I denna övningsuppgiften skall vi testa på bindings i en separat Angular App. **Obs.** Gör inte detta i Star Wars appen från Övning 5 - vi kommer att arbeta vidare med Star Wars appen i Övning 7.

1. Skapa en ny Angular App med "ng new"
2. Skapa en Component (med ng g c) som du kallar för **bindings**
3. I app.component.html - ta bort standard HTML:en som du fick på köpet. Anropa din nya "bindings"-Component från html:en.

Nu kommer app-component vara förälder till bindings (parent / child).

Property Binding

1. Skapa ett nytt attribut i "bindings" som du kallar för counter (number, initiera det till 0).
2. I html:mallen för "bindings" skriv ut texten "Counter är nu {{ counter }}". När du tittar på din app skall du se "Counter är nu 0". **Detta är property binding**
3. Vi vill nu initiera counter från app.component.html när componenten skapas. Detta gör du genom att
 1. I binding.component.ts - importer "Input"
 2. I binding.component.ts - dekorera "counter" med @Input()
 3. I app.component.html - sätt input till "5" genom att skapa ett [counter] attribut i <app-bindings>-taggen.

När du nu kör din kod kommer den visa "Counter är nu 5". **Detta är också property binding** fast mellan Components. Vi satte counter till 5 med ett statiskt tal - detta går också att göra med ett attribut från app.component.

Event binding

1. Skapa en metod i "binding.component.ts" - kalla den btnClickAddOne(). Metoden skall öka värdet av counter (attributet i Componenten) med 1.
2. Skapa en knapp i binding.component.html - när knappen trycks skall btnClickAddOne() köras. Detta gör du genom att sätta (click)="btnClickAddOne()" på knappen.
3. När du trycker på knappen skall du se att "Counter är nu 5" räknas upp med 1 för varje gång du trycker på knappen. **Detta är event binding.**
4. Vi vill nu att app-component får en signal om att countern har ändrats. Detta gör vi genom att skicka ett event från binding- till app-component.
 1. I binding.component.ts - importera "Output, EventEmitter"
 2. I binding.component.ts - skapa ett nytt attribut som du dekorerar med @Output(), det skall heta counterChanged och tilldelas en ny EventEmitter<number>.**Spoiler:** @Output() counterChanged = new EventEmitter<number>();

3. I binding.components.ts - Uppdatera så att btnClickAddOne() sänder ut (emit) det nya värdet på counter.

Spoiler: counterChanged.emit(this.counter);

5. Nu skickar binding-component eventet men app.component tar inte emot det. Låt oss fixa det.

1. I app.component.ts - i class:en - skapa en ny metod som heter onCounterChanged och har en parameter som heter counter och är av typen number. Metoden skall rapportera värdet av counter med hjälp av alert till användaren.
2. I app.component.html - koppla ihop counterChanged eventet med din onCounterChanged() - metod i taggen för <app-binding>.

Spoiler: (counterChanged)="onCounterChanged(\$event)"

När du trycker på knappen - skall du se det nya numret i en alert ruta och samtidigt skall värdet på sidan uppdateras. **Detta är också event binding** fast mellan Components.

