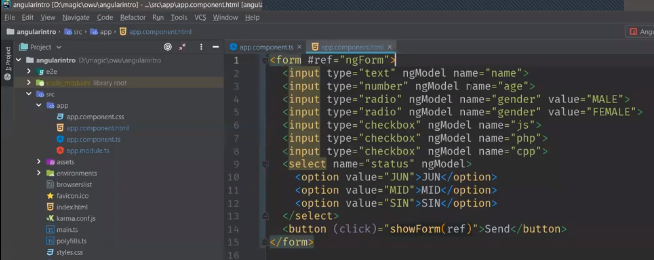
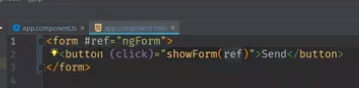
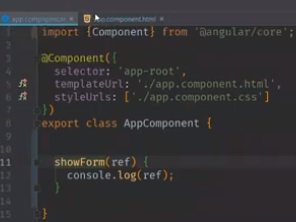
**FORMS**

Односторонній bainding (коли не потрібні попередні дані, а лише хочете щось створити)

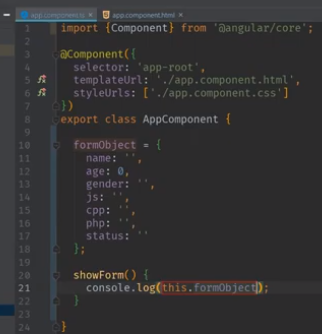


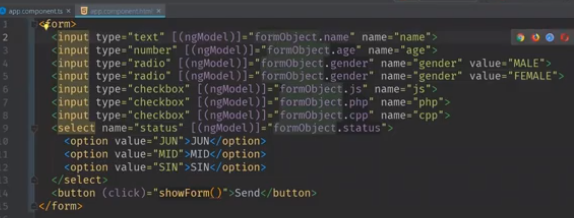
вказуємо ссилку на Ангулярівську форму: #ref =”ngForm”





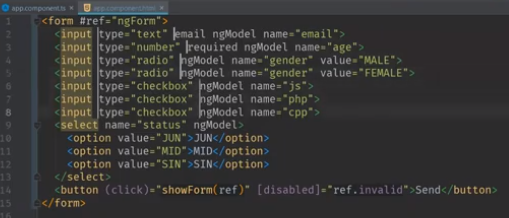
2-x сторонній bainding: (якщо вже є якись обєк і хочемо засунути його у форму і почати його змінювати + можемо поставити дефолтні значення)



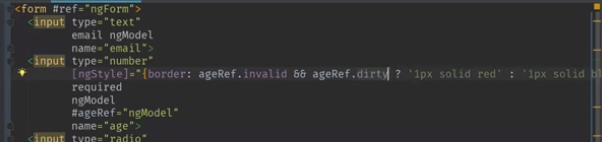


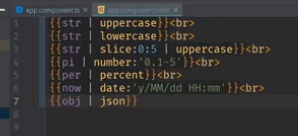
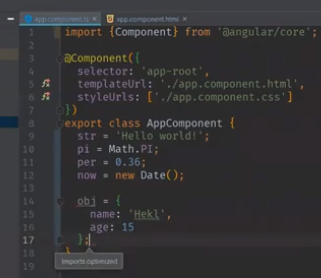
**Валідатори**

Email/ required/ disabled



Блокує кнопку, якщо не всі поля заповнені

Якщо хочемо отримати інформацію з конкретного елемента, то дописуємо 

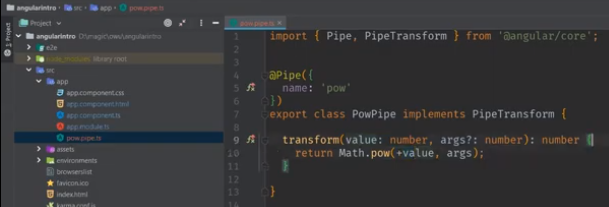
PIPE  


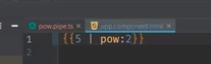
-функціонал, який пропонує Ангуляр і дозволяє змінити вивід вже існуючих даних

Створюємо свій власний pipe.

Пишемо в терміналі: ng g pipe pow (як називатиметься).

Привязуємо його (як і нові компоненти) до модуля в папці в app.module.ts (import + declarations)



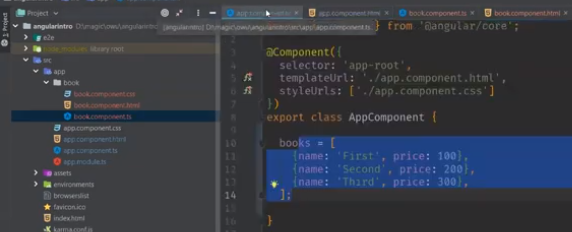


**Components**

Пишемо в терміналі: ng g component book

(+ не забуваємо прописувати в деклараціях)

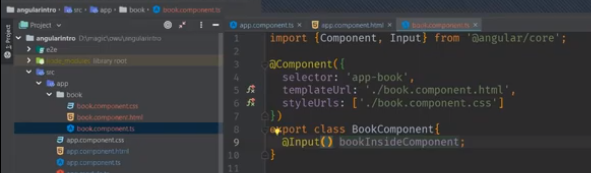
Батьківський елемент: app.components.ts



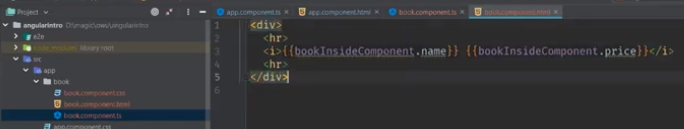
Аби передати дані з батьківського елемента в дочірній в App.components.html створюємо <app-book> і якусь нову внутрішню змінну дочірнього елемента:



Приймаємо змінну в book.components.ts (робим її відкритою для використання): @input( )

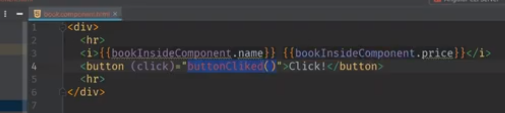


Передаємо в HTML у book.components.html



**Передаємо тепер від батьківського компонента дочірньому.**

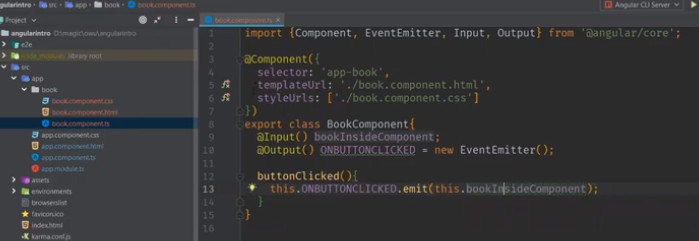
(створюємо зв'язок між компонентами)



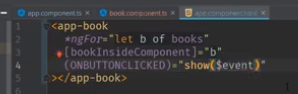
Віддаємо дані: @output( )+назва події = new EventEmitter()

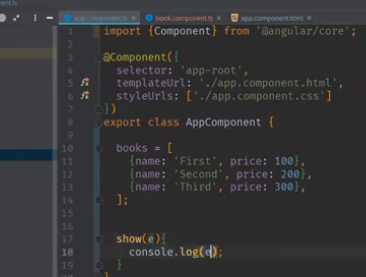
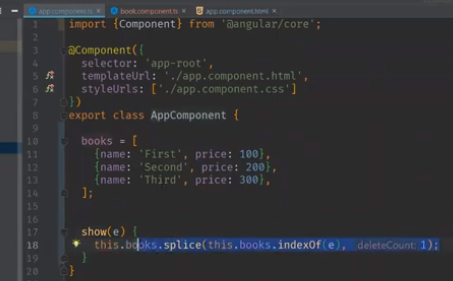
book.components.ts

(Передаємо всю книжку)



Аби батьківський компонент відреагував на подію і прийняв дані: $event

 app.component.html

 Виводимо дані в консоль Видалаяємо дані, що надходять ( в даному випадку -книжки)