




# @Input





## @Input - декоратор для передачи входящих параметров компонента

<https://angular.io/guide/template-syntax#input-and-output-properties>

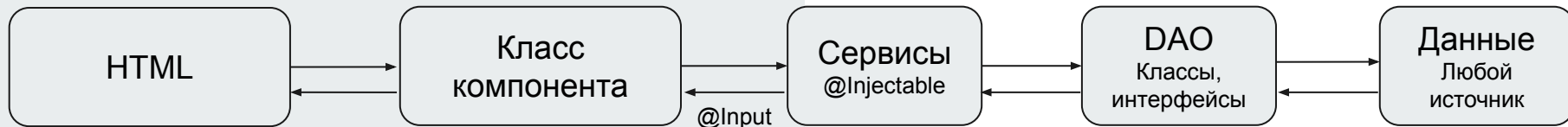
Сервис передает нужные данные компоненту (после какого-либо события или при инициализации)

**@Input** - автоматически подписывается на изменение данных (как с Observable и subscribe) и заменяет по смыслу

Видны все входящие данные для компонента (все **@Input** параметры)

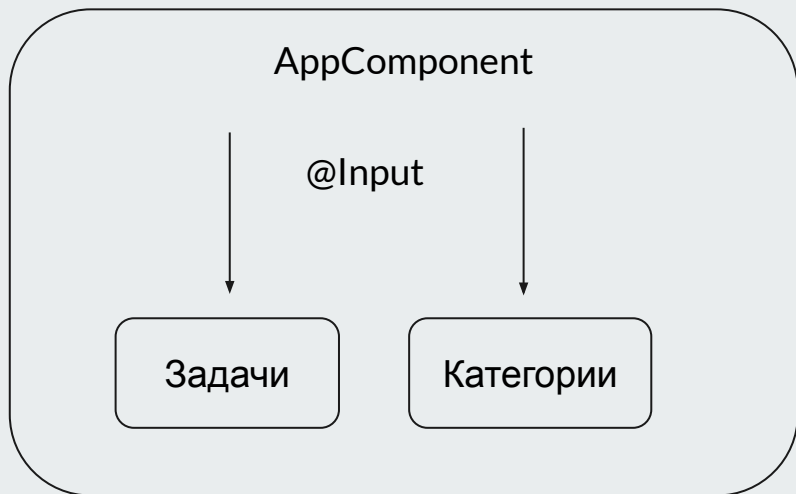
Компонент может иметь ряд @Input переменных, нужные для отображения данных

Компонент сам напрямую не запрашивает данные, ему их передает сервис (по необходимости)





# Parent и Child КОМПОНЕНТЫ



**Parent** - родитель - в нашем случае AppComponent

**Child** - дочерний - в нашем случае TasksComponent

## AppComponent (parent)

1. Подписывается на все необходимые данные (subscribe) из DAO
2. "Раздает" данные всем дочерним элементам
3. Собирает итоговую страницу из компонентов как "конструктор"

## TasksComponent (child)

1. Ждет входящие данные через @Input (как только данные изменятся, @Input это увидит и обновит переменную)
2. Отображает данные в своем view



# Домашнее задание

- Реализовать отображение категорий с помощью @Input
- Сохранить в закладку "Cheat sheet" (шпаргалка) <https://angular.io/guide/cheatsheet>
- Прочитать только до раздела @Output  
<https://angular.io/guide/template-syntax#input-and-output-properties>