Getting data form the controller into the view

1. Strongly-typed views (@model) 🡪 return view(Model)
2. ViewData, ViewBag 🡪 two different ways to look at your object that’s being instantiated. (loosely, weakly-typed views)
3. ViewData 🡪 has **keys**. ViewData[“keys”] = value;
   1. Like a dictionary.
   2. Object type
   3. Non-generic, so, may need to cast when accessing
   4. Mapping object to object.
   5. Disappears across requests
4. ViewBag 🡪 different way to access ViewData; both for reads and for wrights
   1. Dynamic types 🡪 turns off compile type check entirely
5. TempData 🡪 used just like ViewData, once inserted its values stay across requests.
   1. TempData.Peek(“kye”) but if you peek it won’t be marked for deletion
   2. Uses cookies by default

Request 🡪

Authorization Filter (check for user validity) 🡪

Resource Filters (just to improve performance. Uses caching) 🡪

Exception filters (exception handling for everything after this point. Model binding.) 🡪

Action Filters 🡪

Action Methods 🡪

Action Filters 🡪

Result Filter 🡪

Result (view)

Partial Views 🡪 @html.RenderPartial(Model.InnerModel)

* Can be nested
* Look into cascade delete