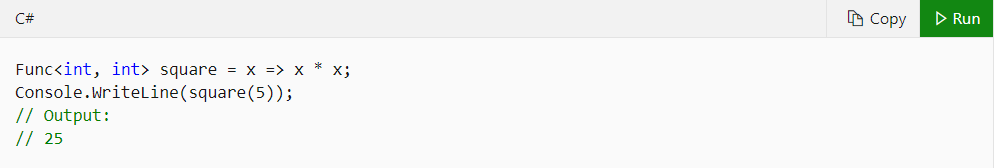
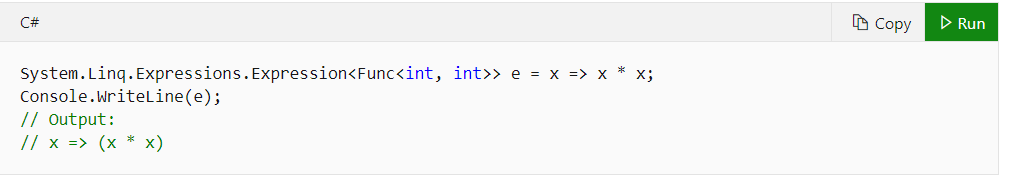


Khi trong ngoặc đơn chỉ có 1 param thì ko cần ngoặc đơn cũng đk

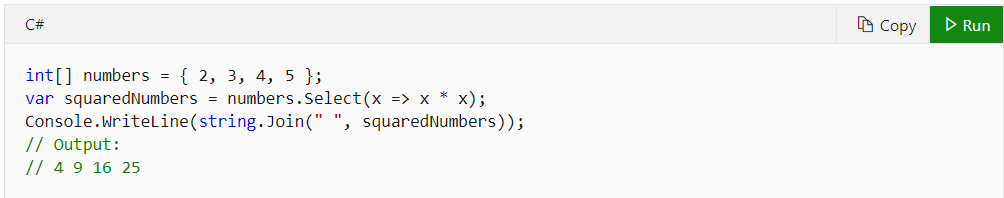
[Action<T1,T2>](https://docs.microsoft.com/en-us/dotnet/api/system.action-2): action có 2 paramater và ko có giá trị trả về, hoặc action có 1 param là T1 và giá trị là T2. Dùng cách này để đưa kiểu dữ liệu đầu vào đầu ra cho lambda, vì trong lambda không chỉ ra kiểu dữ liệu



Lambda tương tự với arrow function

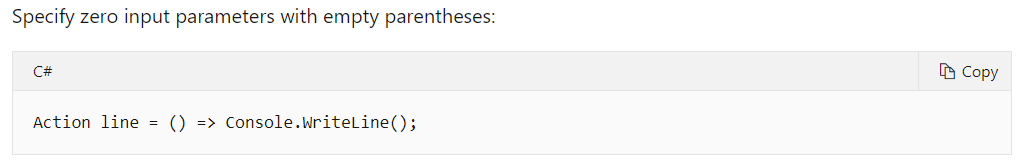


Dùng expressions để in ra expression

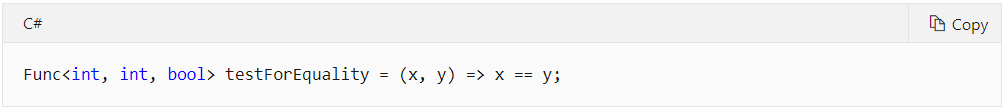


Có thể dùng lambda làm param cho function ( thự ra thì cái select ni toàn dùng lambda)

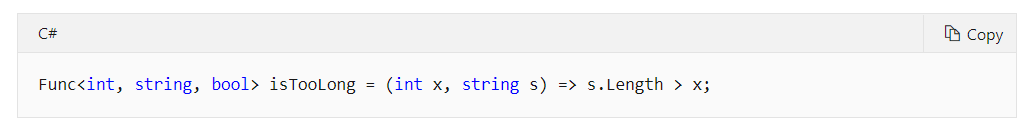
Lambda dùng ko đi kèm với action như ở đây thì nghĩa là đầu vào là x, còn đầu ra là x\*x



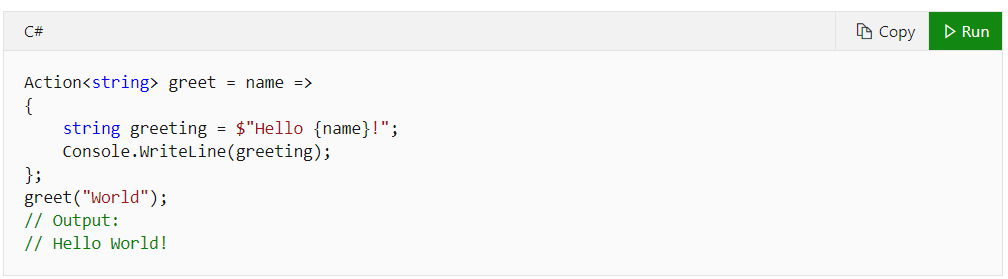
Tương tự arrow function thì nó ko có param, ko có giá trị return vẫn ok



Giá trị trả về ở đây kiểu bool

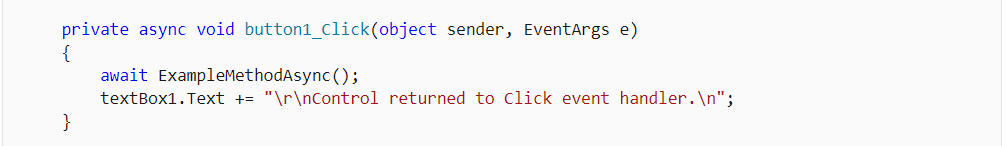


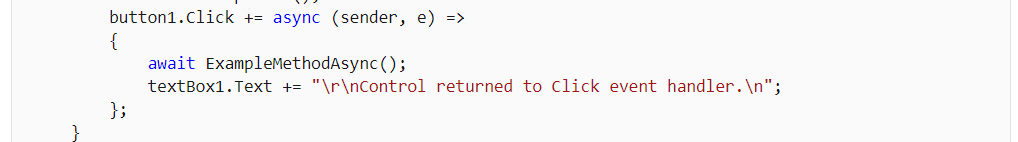
Đôi lúc khó để suy luận ra kiểu của input, thì đưa vào trong input của lambda luôn



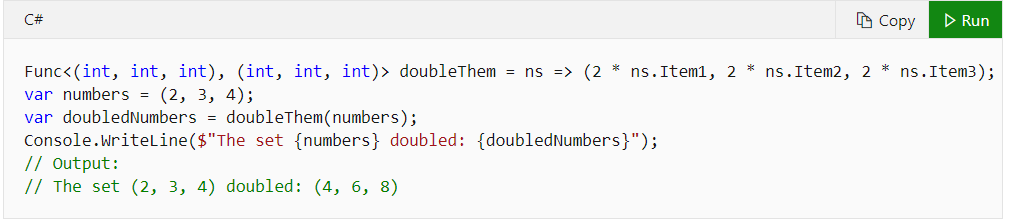
Y chang arrow function luôn

**Async lambdas**



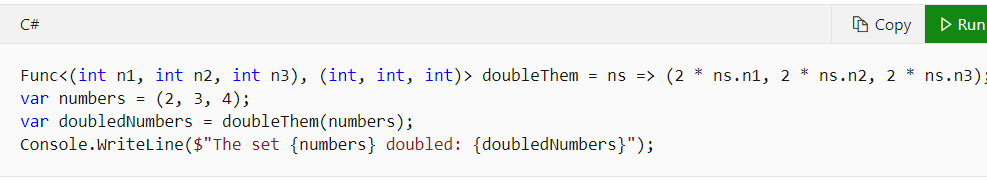


Dùng lambda với async function

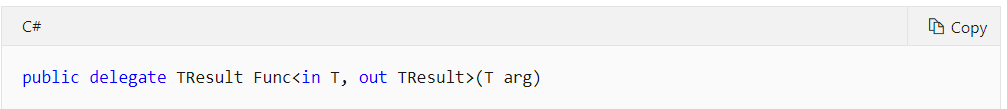


Sử dụng lambda với tuple và tên của đầu vào mặc định là Item1, Item2,…

Typle được khởi tạo = cách (1,2,3)



Có thể đổi tên mặc định đó khi truyền vào

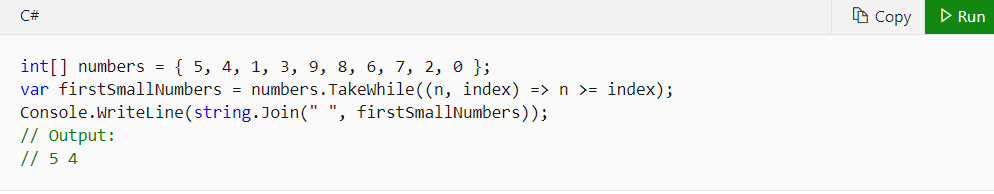


Trong fun sử dụng generic

Sử dụng với delegate

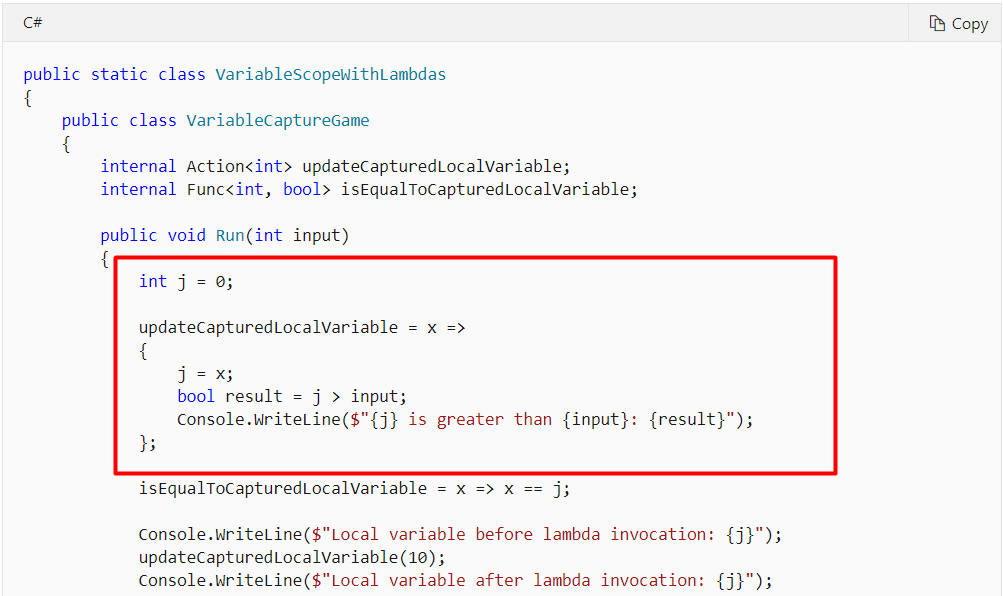


Dùng labmda truyền vào cho funtion



Kiểu dữ liệu truyền vào của lambda sẽ được ngầm hiểu theo kiểu dữ liệu của input

Ngoài ra có thể dùng action<t1,t2> để xác định trước kiểu dữ liệu trong lambda



Hàm của lambda có thể sử dụng các biến trong scope ( như js luôn), ví dụ như biến j ở đây