**C# Map Samples**

**Program.cs**

1. demo\_urls

**Map.cs**

1. demo\_map\_imperative
2. demo\_map\_oo\_10\_11
3. demo\_map\_oo\_20
4. demo\_map\_func\_extension

return urls.Select(url => new Uri(url).Host);

1. demo\_map\_func\_query

return from url in urls select new Uri(url).Host;

**Program.cs**

1. demo\_main\_map

**F# Map Sample**

**Map.fs**

let GetHosts urls = Seq.map (fun url -> (new Uri(url)).Host) urls

**C# Reduce Sample**

**Reduce.cs**

1. demo\_reduce\_imperative
2. demo\_reduce\_oo
3. demo\_reduce\_func

return domains.Aggregate((state, item) => state + ", " + item);

**Program.cs**

1. demo\_main\_reduce

**F# Reduce Sample**

**Reduce.fs**

let HostsToCsv urls = urls |> Seq.reduce (fun state item -> state + ", " + item)

**C# MapReduce Sample**

**MapReduce.cs**

* demo\_reduce\_func

urls.Select(url => new Uri(url).Host).Aggregate((state, item) => state + ", " + item)

**MapReduce.fs**

let UrlsToCsv urls = urls |> Seq.map (fun url -> (new Uri(url)).Host) |> Seq.reduce (fun state item -> state + ", " + item)