

INTRODUKSJONSKURS .NET

C#, LINQ, ASP.NET MVC, ...

Bootcamp 2015

Lars S & Lars S

05/08/15

- .NET 101
- Grunnleggende C#
- LINQ
- **Lunsj**
- ASP.NET MVC 5 Web Apps

bekkopen.github.io/dotnetkurs

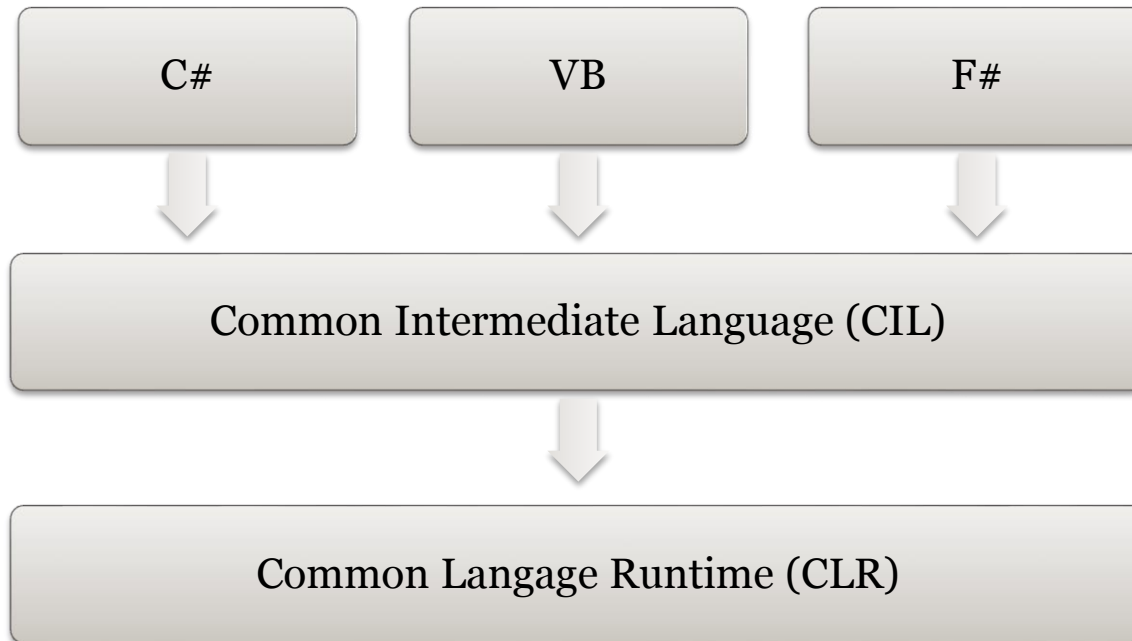
A soft-focus photograph of a person's hands typing on a white keyboard. A white mug is visible on the left side of the frame. The background is bright and out of focus, creating a warm, ambient atmosphere. A black circular overlay is positioned on the left side of the image, containing the text "Demo" and "Hello World" in white.

Demo
Hello World

.NET 101



FRA KODE TIL KJØRENDE PROGRAM





C#




```
public class Person
{
    public Person() {}
}
```

- Getter og Setter

```
public class Person
{
    private int age;
    public int Age
    {
        set
        {
            if (value >= 0) age = value;
            else throw new ArgumentOutOfRangeException();
        }
        get { return age; }
    }
}
```

- Getter og Setter

```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```

- Initialisering

```
Person p = new Person();
p.Name = "Alfred";
p.Age = 89;
```

- Enda enklere

```
Person p = new Person {Name = "Alfred", Age = 89};
```

- Getter og Setter

```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```

- Initialisering

```
Person p = new Person();
p.Name = "Alfred";
p.Age = 89;
```

- Enda enklere

```
var p = new Person {Name = "Alfred", Age = 89};
```

- Standard

```
string Greet(string name) { return "Hello " + name; }
```

- Optional

```
string Greet(string name = "You") { return "Hello " + name; }
```

- Params

```
string Greet(params string[] names) { ... }  
Greet("Ole", "Dole", "Doffen");
```

- Out

```
bool Validate(int age, out string reason) {  
    reason = null;  
    if (age >= 18) {  
        return true;  
    }  
    else {  
        reason = "Too young";  
        return false;  
    }  
}
```

```
string reason;  
var isValid = Validate(17, out reason);  
  
if (!isValid) Console.WriteLine(reason);
```

- Reference

```
void Increase(ref int age){  
    age++;  
}
```

```
int i = 1;  
Increase(ref i);
```

```
Console.WriteLine(i);    // 2
```



- Verdien til en referansetype er referansen

```
void Foo(StringBuilder x){  
    x = null;  
}
```

```
StringBuilder y = new StringBuilder();  
y.Append("Hello");  
Foo(y);  
Console.WriteLine(y == null);    // false
```


- Verdien til en referansetype er referansen

```
void Foo(StringBuilder x){  
    x.Append(" world");  
}
```

```
StringBuilder y = new StringBuilder();  
y.Append("Hello");  
Foo(y);  
Console.WriteLine(y);    // "Hello world"
```

- Metode som gir stringrepresentasjon av objektet

```
public class Student
{
    public int Age { get; set; }
    public string Name { get; set; }

    public override string ToString()
    {
        return string.Format("{0} ({1} years)", Name, Age);
    }
}
```

Demo

Instansiering
av objekter



The background of the slide is a dimly lit office with a green tint. Silhouettes of people are visible working at desks with computers. Large windows in the background show a city skyline. A large black circle is overlaid on the left side, containing the text.

Oppgaver

Personer

Log

(1 til 5)

DELEGATER

- Objektorienterte metodepekere, deklarerer på lik linje som klasse og interface:

```
delegate double Funksjon(double x);
```

- Instans som peker på eksisterende metode:

```
Funksjon f = new Funksjon(Math.Sin);  
double x = f(1.0);
```

- Kan deklarerer anonymt

```
Funksjon f = delegate(double x) { return x + 1; };  
double y = f(1.0);
```

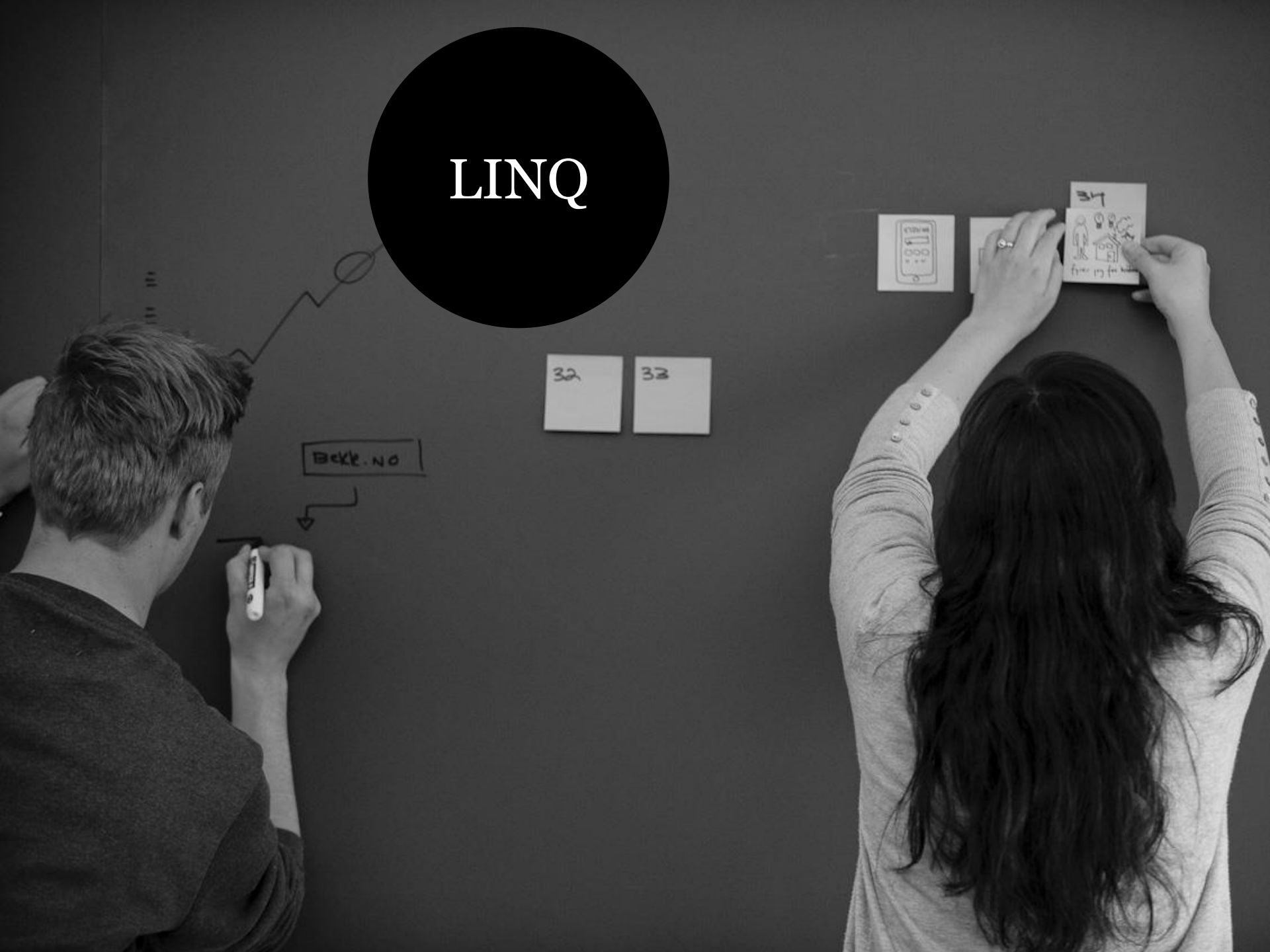
- Innkapsler konsept om at noe har skjedd
- Definere en event:
`public event Action SomethingChanged;`
- Abonnere på en event:
`SomethingChanged += delegate() { Console.WriteLine("Foo"); };`
- Kalle en event:
`SomethingChanged();`



Demo

Delegates &
Events

LINQ



SPØRRINGER FØR LINQ

- Objekter:

```
foreach(Item item in items)
    if(item.Name.StartsWith(firstLetter))
```

- SQL:

```
"select * from students where name like " + firstLetter + "%"
```

- XML:

```
"//Student[starts-with(Name,"+ firstLetter +")]"
```

- C#

```
var studentsAboveDrinkingAge =  
    from student in students  
    where student.Age > 20  
    select student;
```

- Begynner med en from-clause:

```
from itemName in sourceExpression
```

- Avsluttes med en select-clause:

```
select selectExpression
```

- Eller med en groupby-clause:

```
group selectExpression by keyExpression
```

- Eksempel:

```
from student in students  
select student.Name
```

- Filtering med where-clause:

```
where predicateExpression
```

```
from n in numbers  
where n > 10  
select n
```

- Sortering med orderby-clause:

```
orderby (keyExpression (ascending/descending))
```

```
from student in students  
orderby student.Name ascending  
orderby student.Age descending  
select student
```

Demo

LINQ



Oppgaver

Filtrering
Group By

(6 og 7)



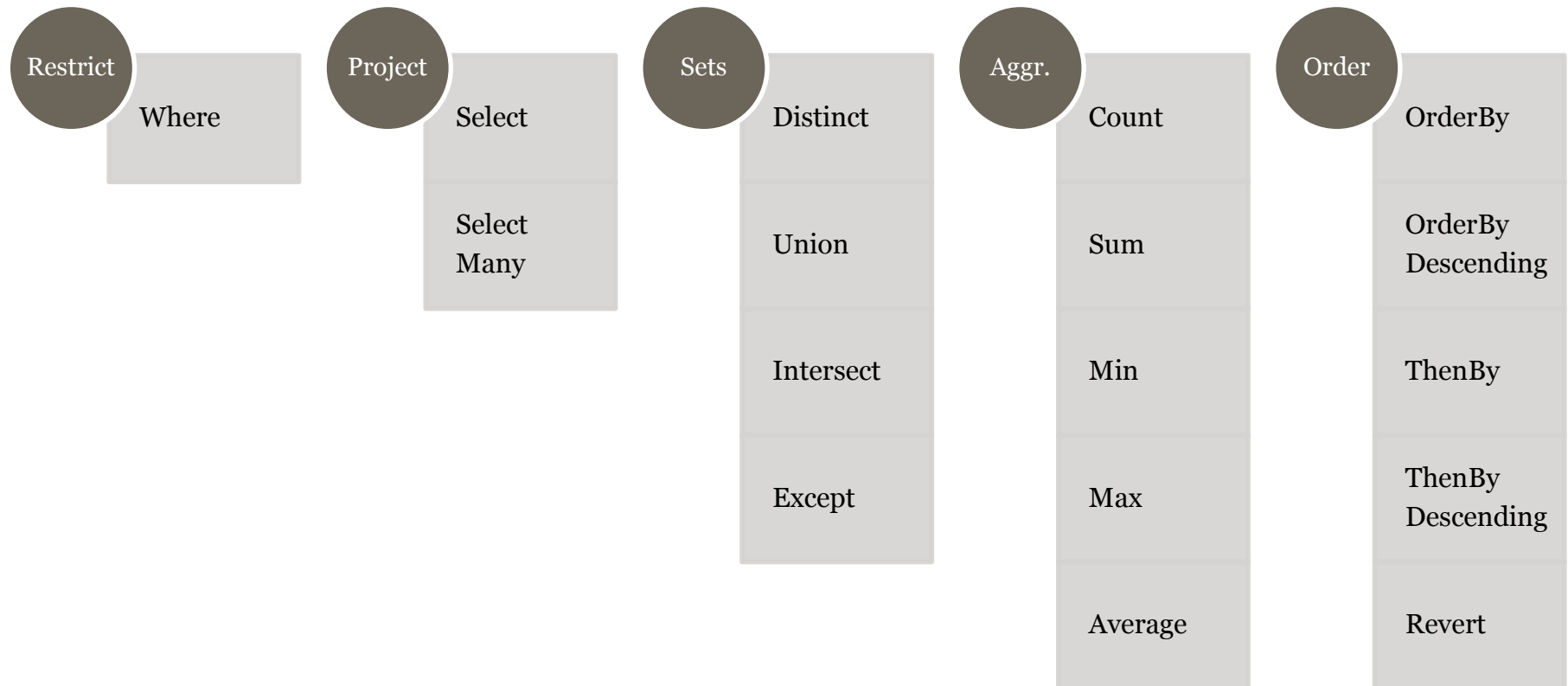
EXTENSION-METODER

- Lar deg legge til ny funksjonalitet til eksisterende typer.

```
public static bool IsNull(this object o)
{
    return o == null;
}
```

```
Student stud = null;
if(stud.IsNull()) {

}
```



- Delegate

```
Predicate<string> isReodor = delegate(string s) {  
    return s == "Reodor";  
};
```

- Lambda-uttrykk

```
Predicate<string> isReodor =  
    s => s == "Reodor";
```

- Filtrere

```
s => s.Code == 123
```

- Trekke ut verdier

```
s => s.Name
```

- Projisere

```
s => new { Name = s.Name, Age = s.Age }
```

- Kombinerer vi extension-metoder og lambda-uttrykk kan vi skrive følgende:

```
numbers.Where(n => n > 10).Select(n => n);
```

- Dette uttrykker akkurat det samme som:

```
from n in numbers  
where n > 10  
select n;
```

- Spørresyntaksen er syntaktisk sukker

- Kombinerer vi extension-metoder og lambda-uttrykk kan vi skrive følgende:

```
numbers.Where(n => n > 10);
```

- Dette uttrykker akkurat det samme som:

```
from n in numbers  
where n > 10  
select n;
```

- Spørresyntaksen er syntaktisk sukker

Oppgaver

Spørreoperatorer
Extension-metoder
Lambda
(8 til 10)



A group of people are seated in a meeting room, viewed from behind. They are looking towards a large window that looks out onto a green landscape. The room has a modern feel with large windows and a whiteboard. A black circle with the word 'Lunsj' in white serif font is overlaid on the left side of the image.

Lunsj

- Hva er ASP.NET MVC?
- Controller
- View
- Model
- NuGet
- Entity Framework & SQL Compact Edition

Hva er ASP.NET MVC?



Web Forms

MVC

ASP.NET

(Request, Response, Session, Cookies,
QueryString, Master Pages)

NDC



5



99X



amende

351

bouvet

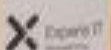


ciber



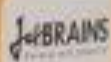
DIPS

EASY

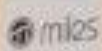


geodata

glas



knowit



NNUG

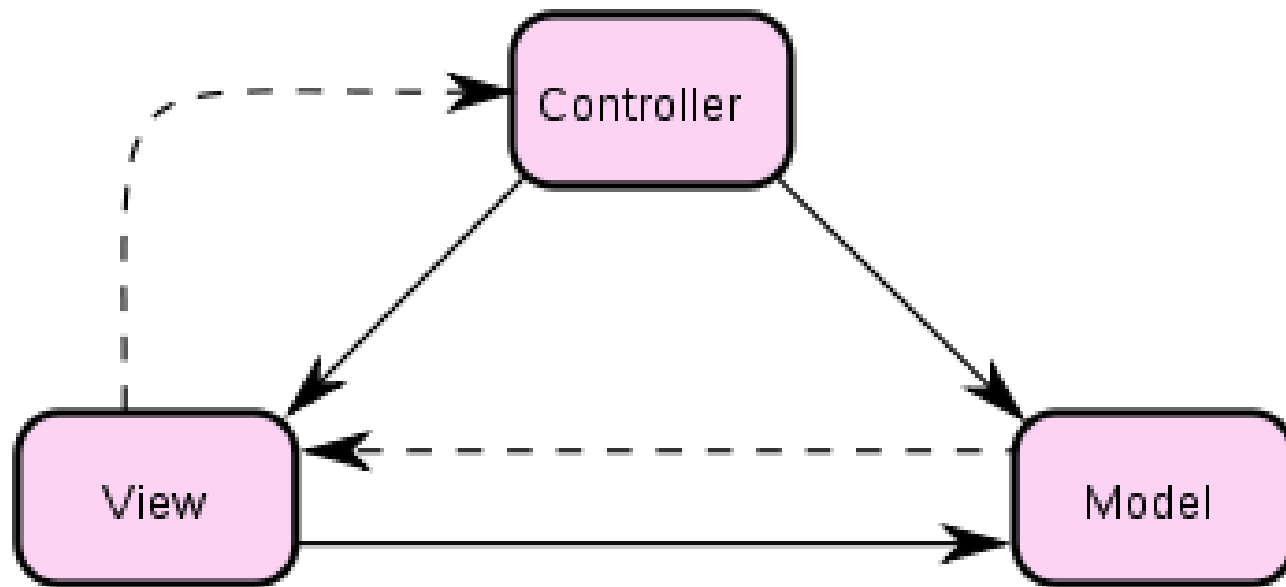


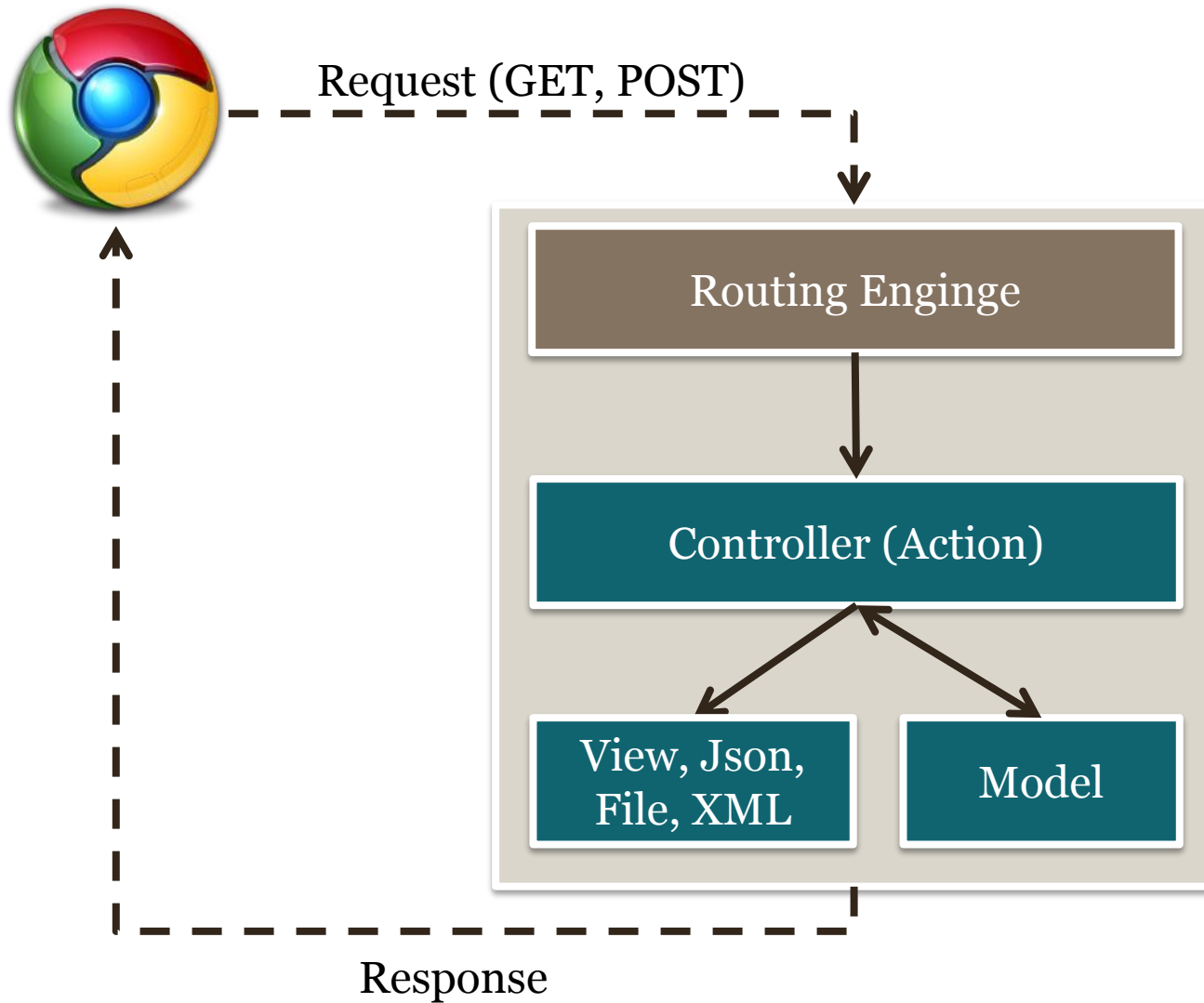
Schlumberger

System

2 V







- Har to ansvar:
 1. Mappe innkommende URL til Controller & Action
 2. Lage utgående URL for lenker etc.

```
routes.MapRoute(  
    "Default", // Route name  
    "{controller}/{action}/{id}", // URL with parameters  
    new { controller = "Home", action = "Index", id = UrlParameter.Optional  
});
```

Url	Controller	Action	Id
/	Home	Index	
/People	People	Index	
/People/Edit	People	Edit	
/People/Edit/123	People	Edit	123

- Mulig å definere egne ruter:

```
routes.MapRoute(  
    "Blog",  
    "Archive/{entryDate}",  
    new { controller = "Archive", action = "Entry" }  
);
```

Url	Controller	Action	entryDate
/Archive/01-01-2015	Archive	Entry	01-01-2015
/Archive/31-06-2015	Archive	Entry	31-06-2015

- Lar deg lage lenker uten å tenke på URL:

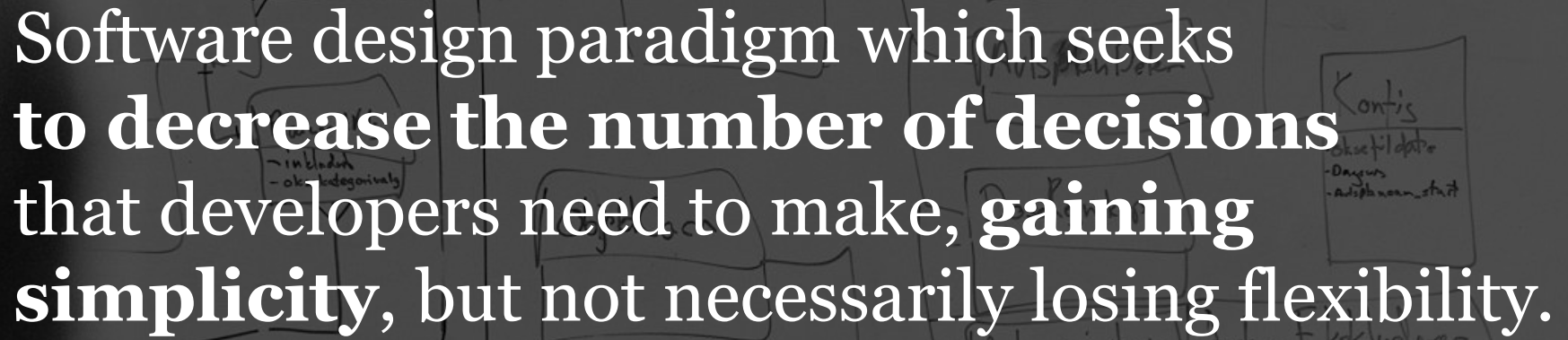
```
@Html.ActionLink("Go Home", "Index", "Home");
```

```
<a href="/Home/Index">Go Home</a>
```

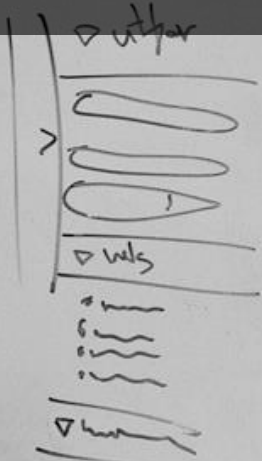
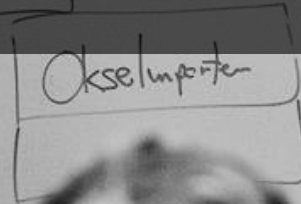
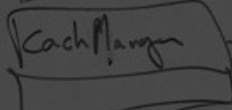
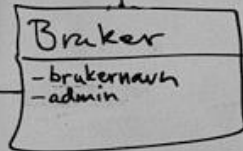
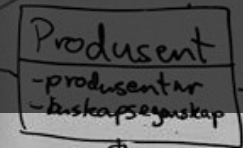
```
@Html.ActionLink("Articles today", "Entry", "Archive",  
    new { entryDate = DateTime.Today}, null);
```

```
<a href="/Archive/05-08-2015">Articles today</a>
```


Convention over Configuration

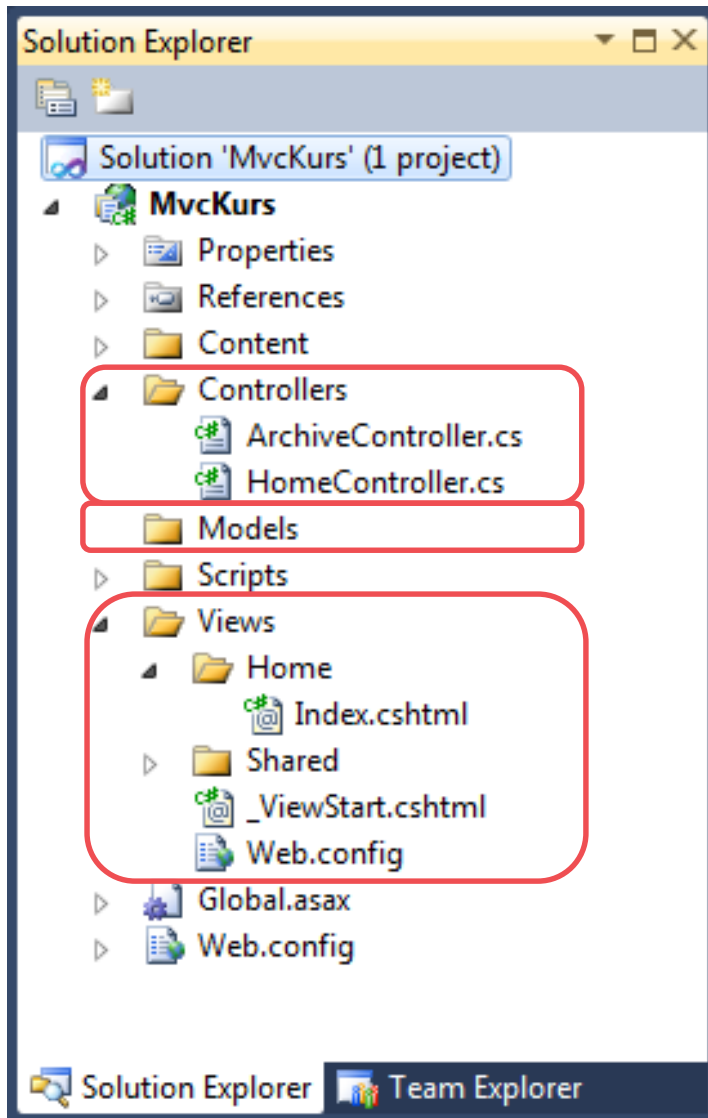


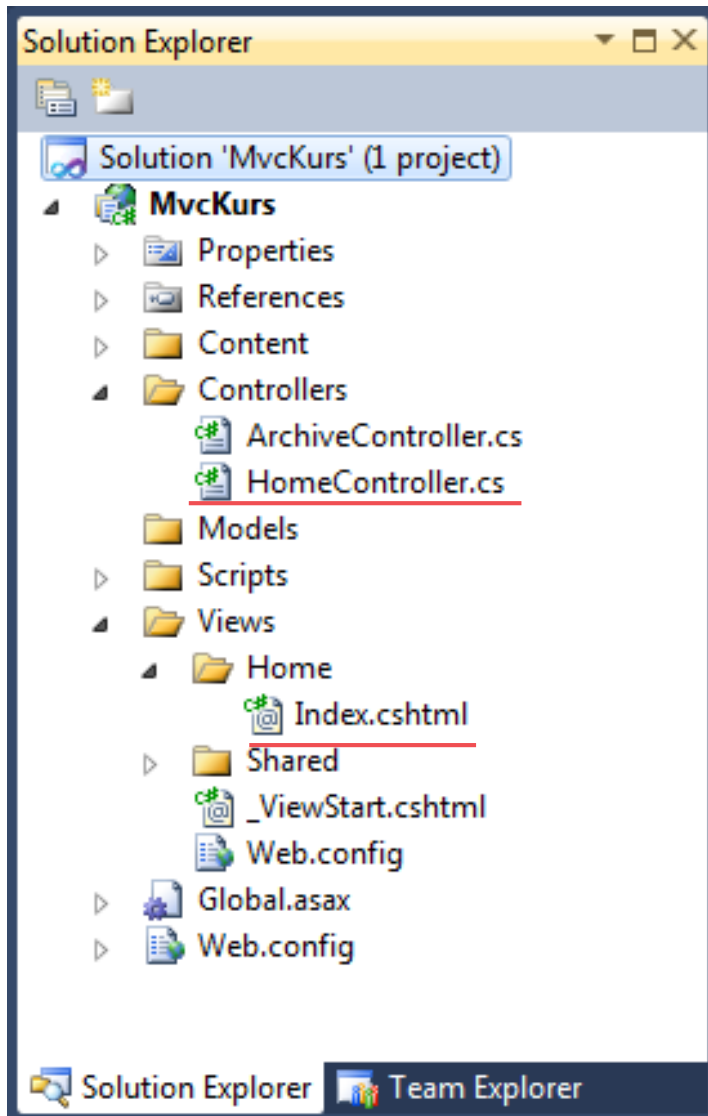
Software design paradigm which seeks **to decrease the number of decisions** that developers need to make, **gaining simplicity**, but not necessarily losing flexibility.



CONVENTION OVER CONFIGURATION

- Konfigurer unntakene:
 - Har du en «Ordre» og en «Ordre»-tabell gjør du ingenting
 - Har du en «Ordre» og en «OrdreData»-tabell konfigurerer du





HTTP GET: /Home/Index

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

Demo
Hello World





Oppgave

Lag to sider
som lenker til
hverandre

VIEWS

- Ansvarlig for generering av HTML
- Konfigurerbart hvilken View Engine du ønsker
 - Web Forms, NHaml, Bellevue, Brail, NDjango, Nvelocity og mange flere
- Kan ha flere View Engines i samme applikasjon
- Razor er den View Enginen flest utviklere bruker i ASP.NET MVC
 - Fokus på bedre flyt mellom kode og markup

```
@if (Model.Products.Count == 0) {  
    <p>We have no products for you...</p>  
} else {  
    <p>We have @Model.Products.Count products for you...</p>  
}
```



```
<h3>Hello @name, it is year @DateTime.Now.Year</h3>
```

```
public ActionResult Index()
{
    ViewBag.Message = "Hello World";
    return View();
}
```

```
<h3>@ViewBag.Message</h3>
```

MODELLER

- Klasser som representerer domene/datamodellen

```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
    public double Height { get; set; }
}
```

- Angi typen som vil bli brukt som Model for et View

```
public ActionResult Index()
{
    var personer = new List<Person> {
        new Person { Name = "Patric Bateman"},
        new Person { Name = "Two Face"}
    };
    return View(personer);
}
```

```
@model IEnumerable<MvcKurs.Models.Person>
<ul>
    @foreach (var person in Model) {
        <li>@person.Name</li>
    }
</ul>
```



Oppgave

Utlisting av
personer

SENDE DATA TIL SERVER

- En Action kan enten være HTTP GET eller HTTP POST
- En Action kan ta i mot et model-objekt som parameter

```
public ActionResult Create()  
{  
    return View();  
}
```

```
[HttpPost]  
public ActionResult Create(Person person)  
{  
    return RedirectToAction("Index");  
}
```

- Metoder for å generere HTML

```
@using (Html.BeginForm("Create", "Home")) {  
    <fieldset>  
        <legend>Person</legend>  
        <div class="editor-label">  
            @Html.LabelFor(model => model.Name)  
        </div>  
        <div class="editor-field">  
            @Html.EditorFor(model => model.Name)  
            @Html.ValidationMessageFor(model => model.Name)  
        </div>  
        <input type="submit" value="Create" />  
    </fieldset>  
}
```

- Metoder for å generere HTML

```
<form action="/Home/Create" method="post">
  <fieldset>
    <legend>Person</legend>
    <div class="editor-label">
      <label for="Name">Name</label>
    </div>
    <div class="editor-field">
      <input class="text-box single-line" id="Name"
name="Name" type="text" value="" />
      <span class="field-validation-valid"
data-valmsg-for="Name"
data-valmsg-replace="true"></span>
    </div>
    <input type="submit" value="Create" />
  </fieldset>
</form>
```


- «Magien» som gjør at en Controller Action kan ta i mot et objekt
- Slipper å manuelt mappe HTTP Request parametere til objekt:

```
[HttpPost]
public ActionResult Create()
{
    var person = new Person();
    person.Name = Request.Form["Name"];
    // osv. osv.

    return RedirectToAction("Index");
}
```

- «Magien» som gjør at en Controller Action kan ta i mot et objekt
- Slipper å manuelt mappe HTTP Request parametere til objekt:

```
[HttpPost]
public ActionResult Create(Person person)
{
    return RedirectToAction("Index");
}
```

- Html Helper metodene sørger for riktig navn på input-feltene
 - `Html.CheckBoxFor()`
 - `Html.DisplayFor()`
 - `Html.DropDownListFor()`
 - `Html.TextBoxFor()`
 - `Html.TextAreaFor()`
 - Osv
- Støtter og mer avanserte brukstilfeller som lister og nøstede objekter

- Baserer seg på metadata og jQuery Validate
- Non-intrusive.
 - Reglene som CSS klasser, JavaScript ligger eksternt.
- Kjører både på klient og server (som en del av Model Binding).
- Utvidbart

```
public class Person
{
    [Required]
    public string Name { get; set; }

    [Range(0, 125)]
    public int Age { get; set; }

    [Range(30.0, 250.0)]
    public double Height { get; set; }
}
```

```
<div class="editor-field">
  <input
    class="text-box single-line"
    data-val="true"
    data-val-number="The field Height must be a number."
    data-val-range="The field Height must be between 30 and 250."
    data-val-range-max="250"
    data-val-range-min="30"
    data-val-required="The Height field is required."
    id="Height"
    name="Height"
    type="text"
    value="" />
  <span
    class="field-validation-valid"
    data-valmsg-for="Height"
    data-valmsg-replace="true">
  </span>
</div>
```



Oppgave

HTTP POST, Model Binder, Validering

A person with short, light brown hair is seen from behind, wearing a dark green t-shirt. They are stretching their arms behind their head, with their hands clasped together. The background is a bright, out-of-focus office space with large windows and another person visible in the distance. A black circle is overlaid on the left side of the image, containing the text "NuGet" in white.

NuGet

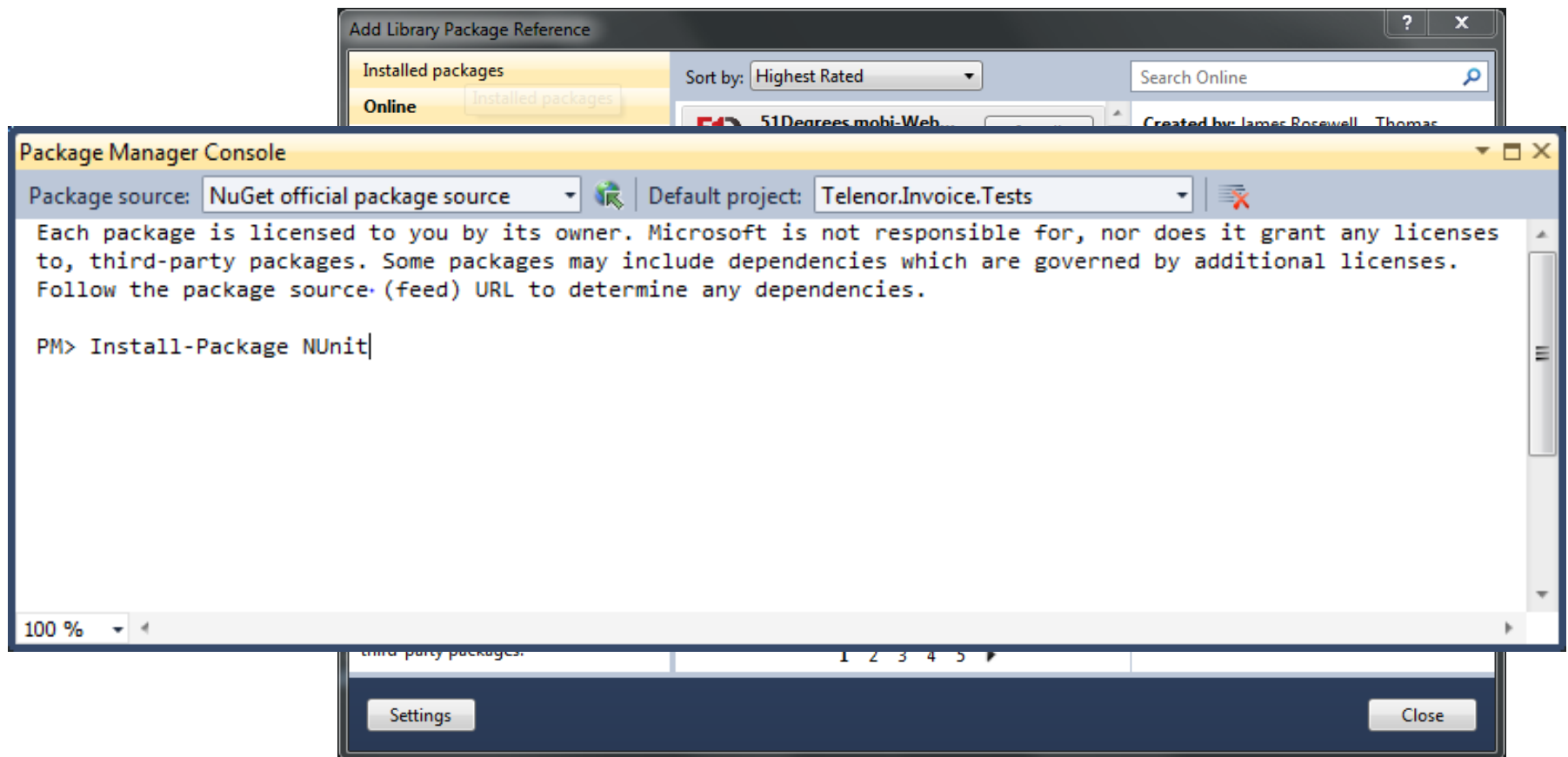
- Pakkehåndtering for .NET
- .NET sitt svar på Ruby Gems, Java Maven, Node NPM, Python PM
- Lansert Januar 2011, 381.478 pakker pr. 01.07.2015
- Open Source prosjekt fra Microsoft, som og tar i mot bidrag fra eksterne.



- Både Open Source pakker fra tredjeparter og Microsoft, og .NET udvidelser
 - NUnit, Elmah, jQuery, Modernizer, Entity Framework (open source)
 - SQL Server Compact Edition (closed source)



- Visual Studio Plug-In
 - UI eller konsoll



- En pakke kan bestå av:
 - Biblioteker (DLL-er)
 - Ressursfiler (JavaScript, CSS, Bilder)
 - Tools (Exe filer)
 - Script som utvider Visual Studio

- Object Relational Mapper
- Mapper .NET objekter mot SQL databaser
- Lar deg bruke LINQ spørringer mot Objekter framfor SQL setninger som må mappes til objekter etterpå
- Sluppet som Open Source tidligere i sommer!

- Installer via NuGet
 - `Install-Package EntityFramework.SqlServerCompact`

- Støtter både «database først» og «kode først» utvikling
 - Begynn med databasen og generer .NET klasser
 - Eller begynn med .NET klasser og generer databasen

- Embedded Database
 - Kjører in-process, in-memory
- Deployes som en binærfil sammen med applikasjonen din
- Installerer som et bibliotek ved hjelp av NuGet
 - `Install-Package Microsoft.SqlServer.Compact`
- Svært enkelt å komme i gang.

- Data Annotations valideres helt ned i Database
- Følger «Id» konvensjoner og gjetter primærnøkkel. Kan og settes eksplisitt.

```
public class Person
{
    public int PersonId { get; set; }

    [Required]
    public string Name { get; set; }

    [Range(0, 125)]
    public int Age { get; set; }

    [Range(30.0, 250.0)]
    public double Height { get; set; }
}
```

- Trenger en DbContext for å kunne jobbe mot databasen:

```
public class MvcKursContext : DbContext
{
    public DbSet<Person> People { get; set; }
}
```

- CREATE

```
var db = new MvcKursContext();
```

```
var newPerson = new Person();  
newPerson.Name = "Lars Skjelbek";
```

```
db.People.Add(newPerson);  
db.SaveChanges();
```

- READ

```
public ActionResult Details(int id)
{
    var db = new MvcKursContext();

    var person = db.People.Find(id);

    return View(person);
}
```

- UPDATE

```
public ActionResult Update(Person person)
{
    var db = new MvcKursContext();

    db.Entry(person).State = EntityState.Modified;
    db.SaveChanges();


    return RedirectToAction("Index");
}
```

- DELETE

```
public ActionResult Delete(int id)
{
    var db = new MvcKursContext();

    var person = db.People.Find(id);
    db.People.Remove(person);
    db.SaveChanges();

    return RedirectToAction("Index");
}
```



Demo
CRUD

Oppgave CRUD



A photograph of a person's hand pointing at a laptop screen. The laptop screen displays a website with a calculator and a calendar. A black circle is overlaid on the right side of the image, containing the text «REST» & Web API in white serif font. The background is blurred, showing a desk and a window.

«REST» & Web API



Oppgave Web API

LYKKE TIL MED C# & .NET!

Lars Skjelbek

Lars Smeby