CODE

1. ng-model er en Angular-funksjon
2. ng er vanlig Angular-forkortelse
3. Viktig å få fram hva som skjer i templaten her
4. Vi UTVIDER html med nye features (her vha et attributt)
5. Start med å skrive inn litt tekst
6. Okey, sjekk ut "ng-app". Hva skjer om vi fjerner den?
7. Flytt den til `<html>`
8. Ikke vis når `name` er tom. `ng-if` to the rescue.

what happened?

1. We have used Angular's "extended HTML"
2. Tested the simplest thing we can do in Angular
3. Haven't written a line of JavaScript

Angular module

1. As soon as we want to write some JS we need a module
2. It's a container for everything your app consists of
3. We'll later look at how to depend on third-party angular code
4. Usually you'll have the "create" in its own file

Controllers and scopes

1. It's time for JavaScript!
2. Notice `ng-controller` og `ng-app`
3. $scope is whats available in the template/html
4. Both variables and functions!
5. Wrap i en form + en ng-submit
   http://jsbin.com/tehete/4/edit
6. Legg til en array og bruk ng-repeat: "name in names"
   http://jsbin.com/tehete/5/edit

Two-way

1. Now you've seen on of the most important features of angular: TWO-WAY BINDING
2. We want to keep to different "worlds" in sync
3. Had we used jQuery or similar libs, this would have been quite a bit of job
4. TEGN PÅ TAVLA (html --- $scope --- ctrl)

Filters

1. Endre til lowercase
2. Notice "ng-model=search", så "| filter:search"
3. Legg til "reverse"-filter.
   KODE: return input.split('').reverse().join('');

DI

1. Angular was created by Java developer
2. So they needed dependency injection
3. One of the most important features of Angular
4. Simplifies splitting the code base into smaller parts and showing what some piece of code depends on

DI

1. Her har vi laget en konstant
2. Som vi så bruker i en controller til å sette start verdi
3. Bytt rekkefølge på "$scope" og "greet"
4. Legg til $timeout i tillegg

```
$timeout(function() {
  $scope.name = "Oppdatert"
}, 4000)
```

Slutt: http://jsbin.com/tehete/10/edit

Talking to servers

1. One of the most important tasks in a JavaScript web application
2. We now create single-page applications
3. They need to send and receive data

Ajax

1. In addition to $scope and $timeout, angular has a lot of other things. For Ajax it has $http
2. Tegn på tavla
3. The server can fail for many reasons
4. Data is missing, validation failed, nothing exists, ...

Saving in Angular

1. We send a regular JavaScript object
2. Angular transforms this into a "JSON string" that the server can understand
3. When the request is finished, either success or error is called

Repeated URLs

1. The problem with $http is repetition.
2. When we work a lot with the same url, it is repeated all over the place
3. There is a better way

Including Resource

1. However, resource is not available yet!
2. This is how we depend on other libraries
3. They are now available for dependency injection

Refreshable URLs

1. An important part of single-page apps
2. What if every time you refreshed you went back to the first page?

Routes

1. We configure a routeprovider
2. html is here in the current directory
3. This is the name of a controller that must be injectable (so it must live in the myApp module)
4. BUT where does the template end up?

Forms

1. We have a user object on scope
2. We reuse our User service that we created earlier using $resource

Validation

1. novalidate is used to disable browser's native form validation.
2. `ng-dirty`: active if user has interacted with the form
3. `ng-invalid`: active if input is invalid

Validation

1. Lek litt med å endre css på ng-invalid, ng-dirty, nn-pristine