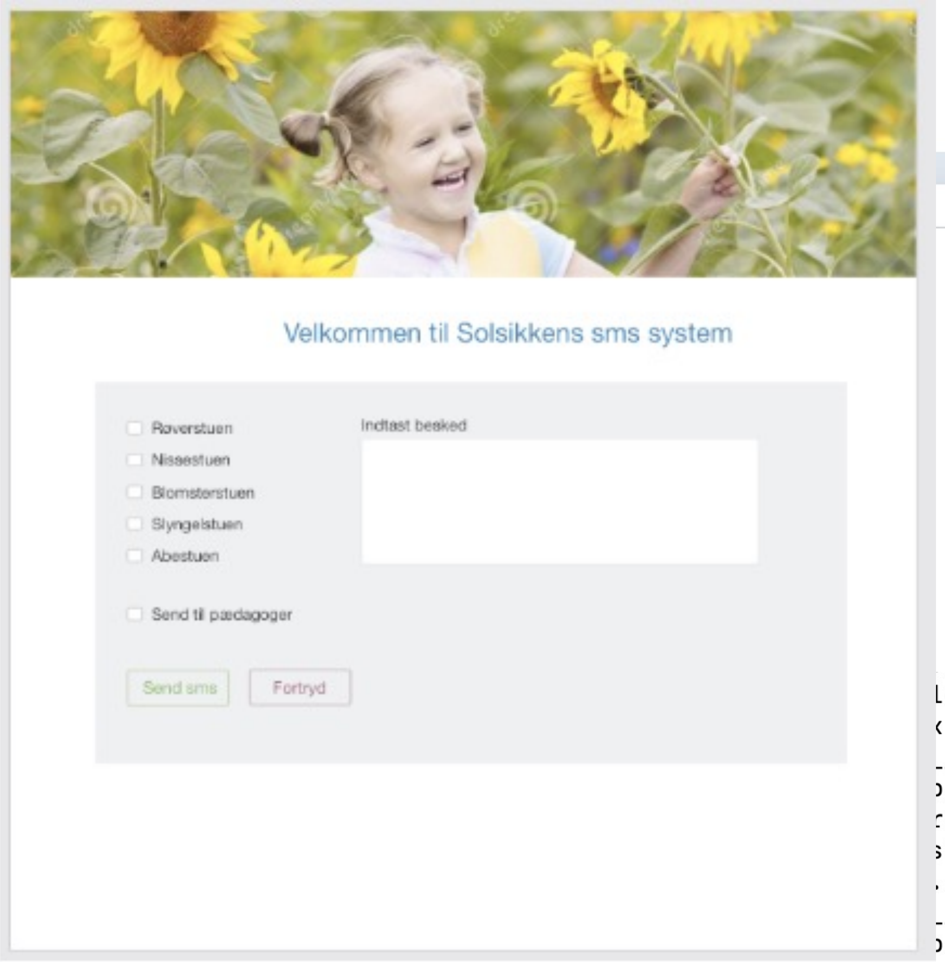
Fra Design til Kode – Form & indhold

- Modul 4 - Webstack
- Recap om web-sider, web-servere (php), html, css og datakilder
- Arbejde i breakout med html/css ud fra design + Jons fly-booking

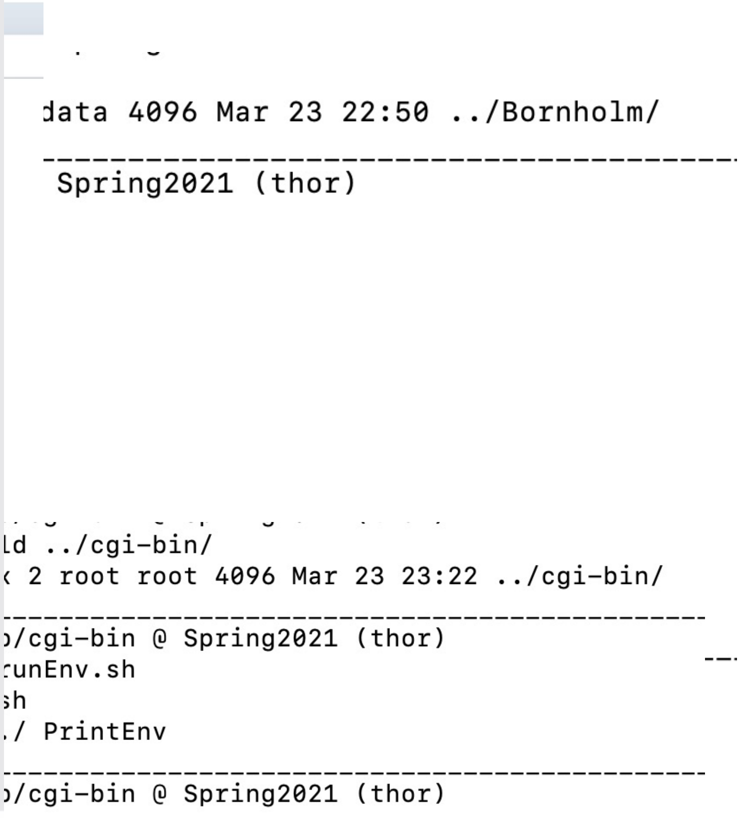


Velkommen til Solsikkens sms system

□ Røverstuen
□ Nissestuen
□ Blomsterstuen
□ Slyngelstuen
□ Abestuen

□ Send til pædagoger

Indtast besked



```
data 4096 Mar 23 22:50 ../Bornholm/
-------------------------------------------
 Spring2021 (thor)
```

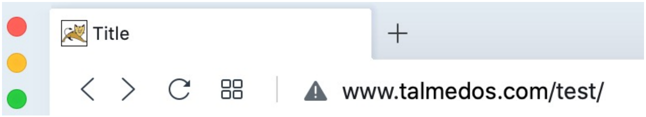
```
ld ../cgi-bin/
( 2 root root 4096 Mar 23 23:22 ../cgi-bin/
------------------------------------------
)/cgi-bin @ Spring2021 (thor)
runEnv.sh
sh
./ PrintEnv
------------------------------------------
)/cgi-bin @ Spring2021 (thor)
```

Take aways …

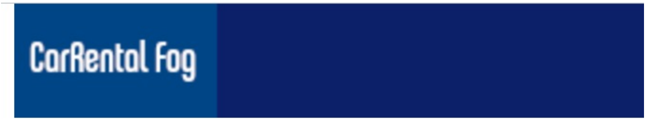
- Need to
  - Bygge en form med alle elementer
    - Inputtypes (husk id og name)
    - Action-attribut der peger på en htmlside
  - Analysér urlens parametre
  - Kende til <script> og <link>
  - Kan ”tænde” udviklerværktøjet
  - Kunne transportere en html-fil til din droplet med f.eks FileZilla
  - åbne og redigere i en texteditor

- Nice to
  - Bygge en form med alle elementer
    - Action-attribut der peger på en cgi
    - Få svar tilbage fra serveren med input
  - Style med css og/eller bootstrap
  - Basal DOM-manipulation med JS
  - Kan identificere fejl i konsollen
  - Kunne transportere en tar-fil til din droplet i gitbash, evt script
  - Kunne redigere en fil i vim
  - Kan ”beskrive” en simpel http-server

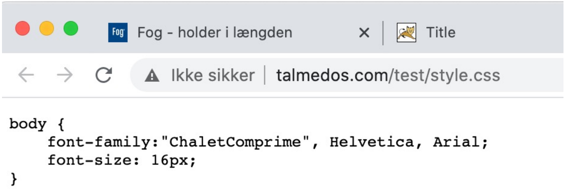
Elementer fra siden



Domæne-navn

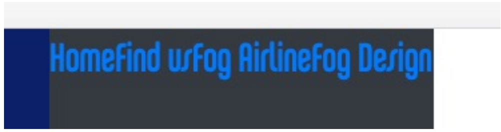


Banner

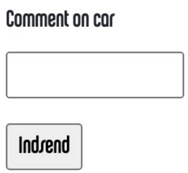


```
body {
    font-family:"ChaletComprime", Helvetica, Arial;
    font-size: 16px;
}
```

Font



Navigation



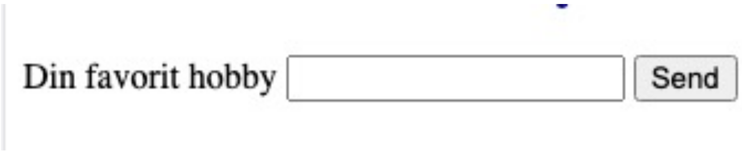
HTML elementer

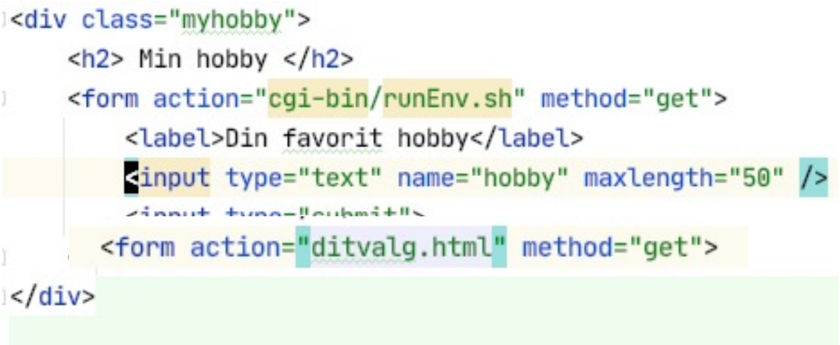
-Forms

-dropdowns

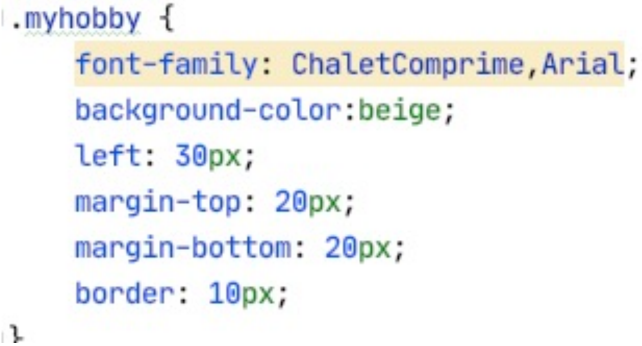
- fritekst søgning

Elementer fra siden II

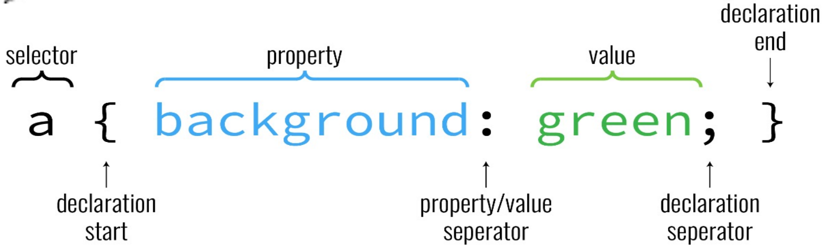




```
<div class="myhobby">
    <h2> Min hobby </h2>
    <form action="cgi-bin/runEnv.sh" method="get">
        <label>Din favorit hobby</label>
        <input type="text" name="hobby" maxlength="50" />
        <input type="submit">
     <form action="ditvalg.html" method="get">
</div>
```



```
.myhobby {
    font-family: ChaletComprime,Arial;
    background-color:beige;
    left: 30px;
    margin-top: 20px;
    margin-bottom: 20px;
    border: 10px;
}
```



Statisk HTML elementer

-Forms (input type="…")

- type, name, id,

Dynamisk HTML

-action=cgi-script or servlet

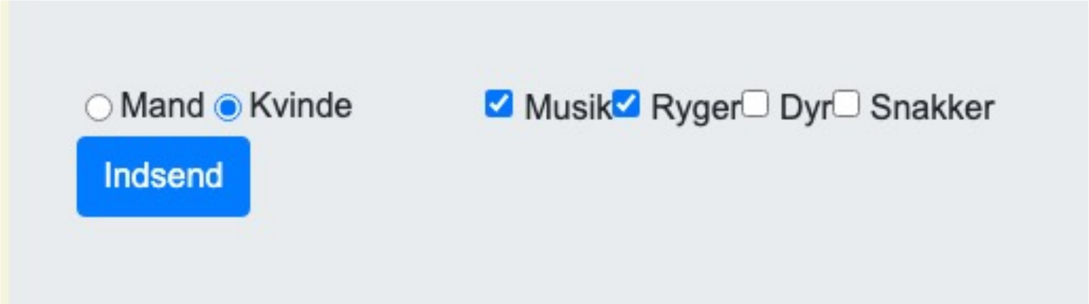
CSS – Formatering

- Selectors
- Text
- Borders
- Positioning
- Box Model

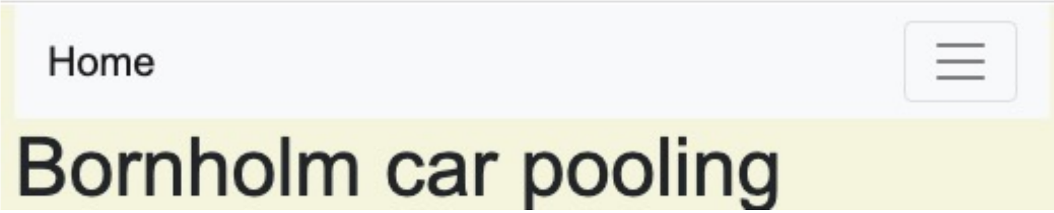
CSS – Implementering

CSS – Regler

Elementer fra siden III – Bootstrap & params



○ Mand ● Kvinde ☑ Musik ☑ Ryger □ Dyr □ Snakker



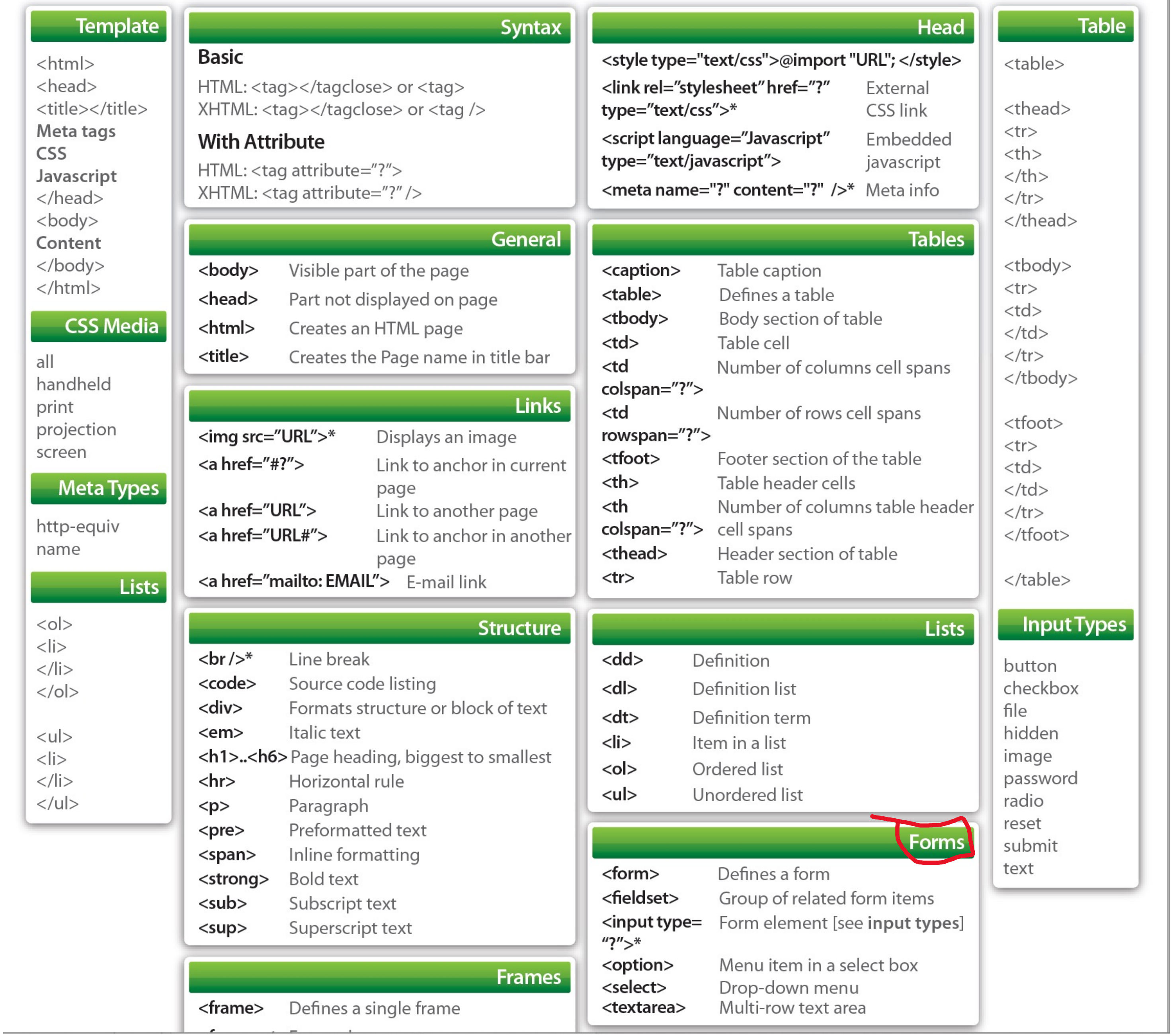
Bornholm car pooling

- Input
- Navbar
- Jumbotron
- Knapper
- Grid

http://localhost:63342/MyBornholm/Bornholm/cgi-bin/runEnv.sh?start=ronne&destsel=ronne&date=&afg=Kurt&dest=Verner&hobby=Ridning&gender=F&mus=mus&r%C3%B8g=smoke&dyr=dyr

TOOLING

- Cheat Sheets
- Chrome
- IDE - IntelliJ



Template

<html>
<head>
<title></title>
Meta tags
CSS
Javascript
</head>
<body>
Content
</body>
</html>

CSS Media

all
handheld
print
projection
screen

Meta Types

http-equiv
name

Lists

<ol>
<li>
</li>
</ol>

<ul>
<li>
</li>
</ul>

Syntax

Basic

HTML: <tag></tagclose> or <tag>
XHTML: <tag></tagclose> or <tag />

With Attribute

HTML: <tag attribute=”?”>
XHTML: <tag attribute=”?” />

General

<body>	Visible part of the page
<head>	Part not displayed on page
<html>	Creates an HTML page
<title>	Creates the Page name in title bar

Links

<img src=”URL”>*	Displays an image
<a href=”#?”>	Link to anchor in current page
<a href=”URL”>	Link to another page
<a href=”URL#”>	Link to anchor in another page
<a href=”mailto: EMAIL”>	E-mail link

Structure

<br />*	Line break
<code>	Source code listing
<div>	Formats structure or block of text
<em>	Italic text
<h1>..<h6>	Page heading, biggest to smallest
<hr>	Horizontal rule
<p>	Paragraph
<pre>	Preformatted text
<span>	Inline formatting
<strong>	Bold text
<sub>	Subscript text
<sup>	Superscript text

Frames

<frame>	Defines a single frame

Head

<style type=”text/css”>@import "URL"; </style>	
<link rel=”stylesheet” href=”?” type=”text/css”>*	External CSS link
<script language=”Javascript” type=”text/javascript”>	Embedded javascript
<meta name="?" content="?" />*	Meta info

Tables

<caption>	Table caption
<table>	Defines a table
<tbody>	Body section of table
<td>	Table cell
<td colspan=”?”>	Number of columns cell spans
<td rowspan=”?”>	Number of rows cell spans
<tfoot>	Footer section of the table
<th>	Table header cells
<th colspan=”?”>	Number of columns table header cell spans
<thead>	Header section of table
<tr>	Table row

Lists

<dd>	Definition
<dl>	Definition list
<dt>	Definition term
<li>	Item in a list
<ol>	Ordered list
<ul>	Unordered list

Forms

<form>	Defines a form
<fieldset>	Group of related form items
<input type= ”?”>*	Form element [see **input types**]
<option>	Menu item in a select box
<select>	Drop-down menu
<textarea>	Multi-row text area

Table

<table>

<thead>
<tr>
<th>
</th>
</tr>
</thead>

<tbody>
<tr>
<td>
</td>
</tr>
</tbody>

<tfoot>
<tr>
<td>
</td>
</tr>
</tfoot>

</table>

Input Types

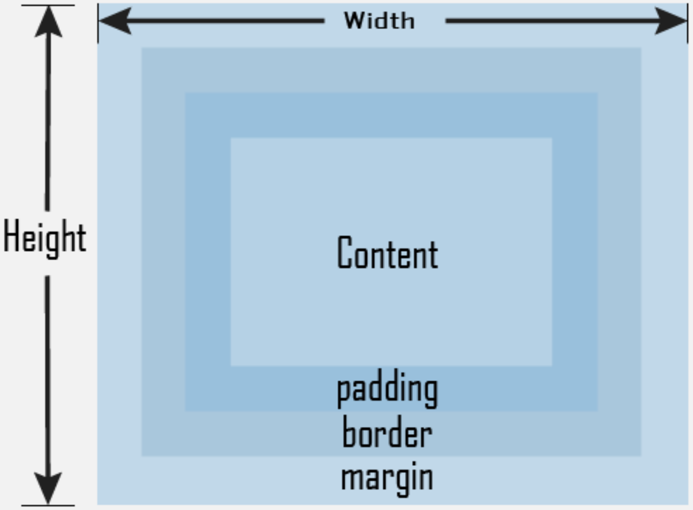
button
checkbox
file
hidden
image
password
radio
reset
submit
text

CSS Cheat Sheet

Selectors

div	all DIV tags
div, spam	all DIV tags and all SPAN tags
div span	all SPAN tags inside DIVs
#content	element with ID “content”
.box	all elements whith CLASS “box”
ul#box	UL tag with ID “box”
span.box	all SPAN tags with CLASS “box”
*	all elements
#box*	all elements inside #box
a:link, a:active, a:visited	links in normal state, in clicked state, and in visited state
a:hover	link with mouse over it
div > span	all SPANs one-level deep in a DIV
div + span	all SPANs preceded by DIV

Box Model



Positioning

Position	place elements on screen, e.g. absolute, fixed, relative
float	stacks elements horizontally in a particular direction, e.g. left
top, left, right, bottom	specifies the offsets used in absolute, fixed, and relative positions, e.g. top:10px;left:10px
display	sets how the elements is placed in the doc flow, e.g. block, inline, none
z-index	sets the stracking order of elements, e.g. z-index of 1 is below z-index of 2
overflow	sets what happens to content outside of container, e.g. auto, hidden

Text

font-family	font used, e.g. Helvetiva, Arial
font-size	text size, e.g. 60px, 3em
color	text color, e.g. #000, #abcdef
font-weight	how bold the text is, e.g. bold
font-style	what style the text is e.g. italic
text-decoration	sets a variety of effects on text, e.g. underline, overline, none
text-align	how text is aligned, e.g. center
line-height	spacing between lines, e.g. 2em
letter-spacing	spacing between letters, e.g. 5px
text-indent	indent of the first line, e.g. 2em
text-transform	applies formatting to text, e.g. uppercase, lowercase, capitalize
vertical-align	align relative to baseline, e.g. text-top

Borders and Dimensions

border	sets border style for all borders, in the format: border: (solid, dashed, dotted, double) (width) (color), e.g. border: solid 1px #000
border-top border-bottom border-left border-right	sets border style for a specific border (same property syntax used for padding and margin, e.g. margin-left)
height	sets height of an element
max-height	sets maximum height of an element
min-height	sets minimum height of an element
width	sets width of an element
max-width	sets maximum width of an element
min-width	sets minimum width of an element

Everything Else

background	sets background of an element, in the format: background: (color) (image) (repeat) (position), e.g. background: #000 url(bg.png) repeat-x top left
cursor	sets shape of cursor, e.g. pointer
outline	a border drawn around an elements that doesn't affect the box model
clear	sets on what side a new line starts in relation to nearby floated elements, e.g. left, right, both
box-shadow	attaches one or more shadows to an element
box-sizing	defines how the width and height of an element are calculated: should they include padding and borders, or not.

The most common DOM methods at a glance

Reaching Elements in a Document

document.getElementById('id'): Retrieves the element with the given **id** as an object

document.getElementsByTagName('tagname'): Retrieves all elements with the tag name **tagname** and stores them in an array-like list

Reading Element Attributes, Node Values and Other Data

node.getAttribute('attribute'): Retrieves the value of the attribute with the name **attribute**

node.setAttribute('attribute', 'value'): Sets the value of the attribute with the name **attribute** to **value**

node.nodeType: Reads the type of the **node** (1 = element, 3 = text node)

node.nodeName: Reads the name of the **node** (either element name or #textNode)

node.nodeValue: Reads or sets the value of the **node** (the text content in the case of text nodes)

Navigating Between Nodes

node.previousSibling: Retrieves the previous sibling node and stores it as an object.

node.nextSibling: Retrieves the next sibling node and stores it as an object.

node.childNodes: Retrieves all child nodes of the object and stores them in an list. here are shortcuts for the first and last child node, named **node.firstChild** and **node.lastChild**.

node.parentNode: Retrieves the node containing **node**.

Creating New Nodes

document.createElement(element): Creates a new element node with the name **element**. You provide the name as a string.

document.createTextNode(string): Creates a new text node with the node value of **string**.

newNode = node.cloneNode(bool): Creates **newNode** as a copy (clone) of **node**. If **bool** is **true**, the clone includes clones of all the child nodes of the original.

node.appendChild(newNode): Adds **newNode** as a new (last) child node to node.

node.insertBefore(newNode,oldNode): Inserts **newNode** as a new child node of node before **oldNode**.

node.removeChild(oldNode): Removes the child **oldNode** from node.

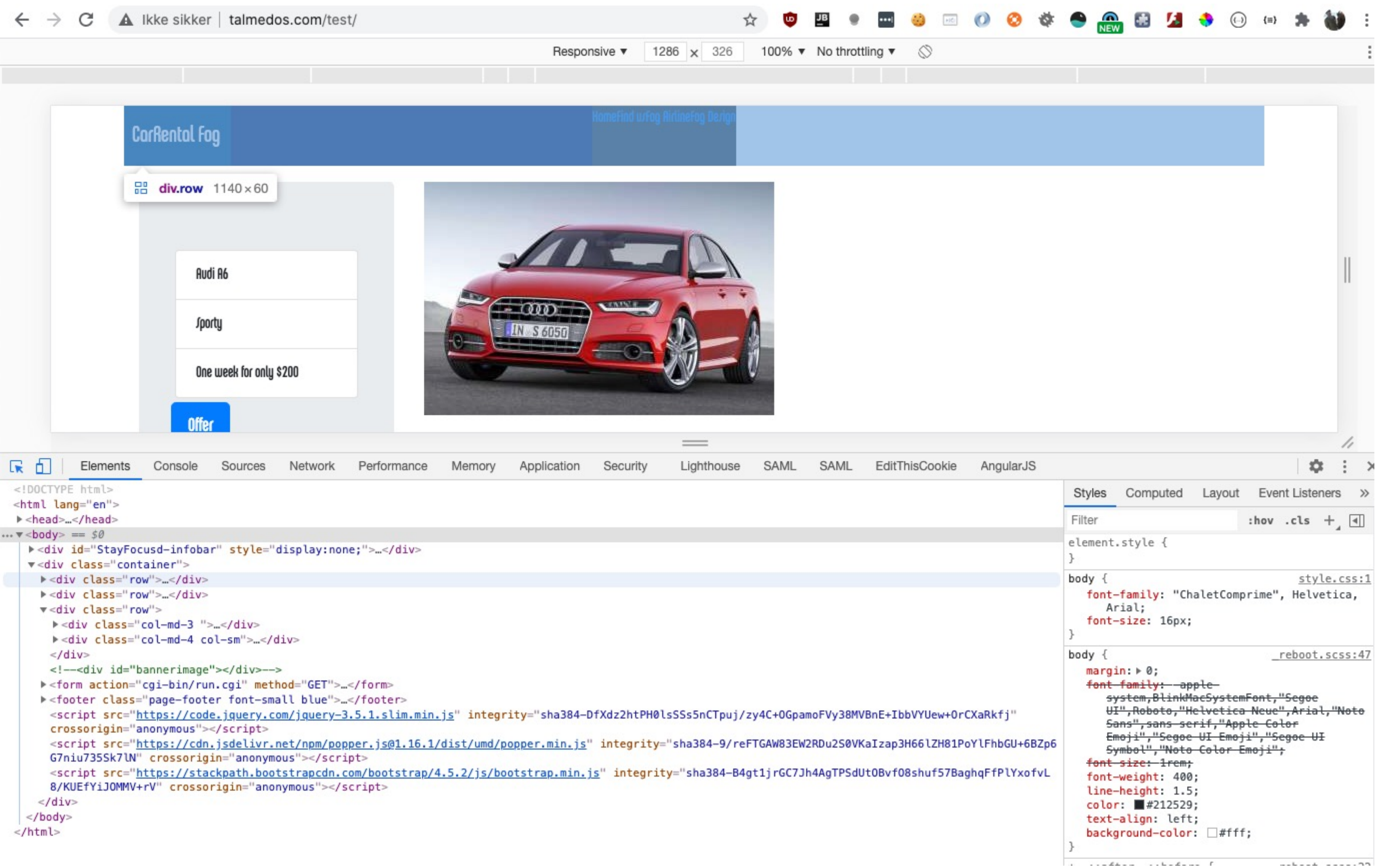
node.replaceChild(newNode, oldNode): Replaces the child node **oldNode** of node with **newNode**.

element.innerHTML: Reads or writes the HTML content of the given element as a string—including all child nodes with their attributes and text content.

Known browser quirks:

getAttribute and **setAttribute** are not reliable. Instead, assign the property of the element object directly: **obj.property = value**. Furthermore, some attributes are actually reserved words, so instead of **class** use **className** and instead of **for** use **HTMLfor**.

Real DOM compliant browsers will return linebreaks as text nodes in the **childNodes** collection, make sure to either remove them or test for the **nodeType**.

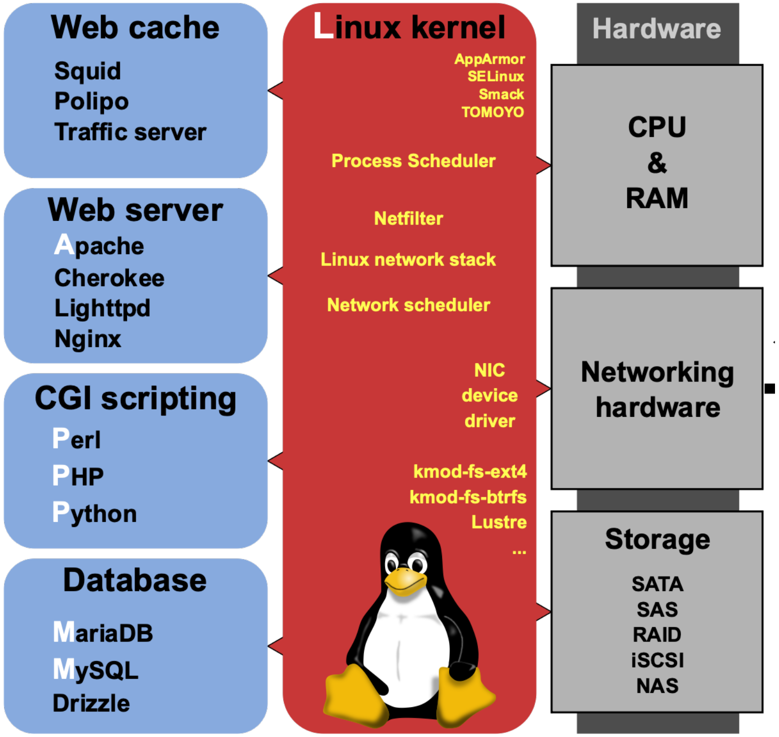


```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
…▼<body> == $0
  ▶<div id="StayFocusd-infobar" style="display:none;">…</div>
  ▼<div class="container">
    ▶<div class="row">…</div>
    ▶<div class="row">…</div>
    ▼<div class="row">
      ▶<div class="col-md-3 ">…</div>
      ▶<div class="col-md-4 col-sm">…</div>
    </div>
    <!--<div id="bannerimage"></div>-->
    ▶<form action="cgi-bin/run.cgi" method="GET">…</form>
    ▶<footer class="page-footer font-small blue">…</footer>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
    crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6
    G7niu735Sk7lN" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL
    8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
  </div>
</body>
</html>
```

```
element.style {
}
body {                                          style.css:1
    font-family: "ChaletComprime", Helvetica,
        Arial;
    font-size: 16px;
}
body {                                      _reboot.scss:47
    margin: ▶ 0;
    ~~font-family: -apple-
        system,BlinkMacSystemFont,"Segoe
        UI",Roboto,"Helvetica Neue",Arial,"Noto
        Sans",sans-serif,"Apple Color
        Emoji","Segoe UI Emoji","Segoe UI
        Symbol","Noto Color Emoji";~~
    ~~font-size: 1rem;~~
    font-weight: 400;
    line-height: 1.5;
    color: ■#212529;
    text-align: left;
    background-color: □#fff;
}
```

Apache HTTP-server - LAMP

LAMP stack



Web cache

Squid
Polipo
Traffic server

Web server

Apache
Cherokee
Lighttpd
Nginx

CGI scripting

Perl
PHP
Python

Database

MariaDB
MySQL
Drizzle

Linux kernel

AppArmor
SELinux
Smack
TOMOYO

Process Scheduler

Netfilter

Linux network stack

Network scheduler

NIC
device
driver

kmod-fs-ext4
kmod-fs-btrfs
Lustre
...

Hardware

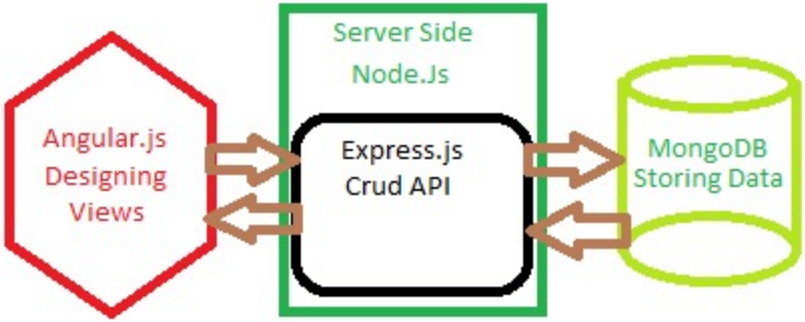
CPU
&
RAM

Networking
hardware

Storage

SATA
SAS
RAID
iSCSI
NAS

MEAN stack



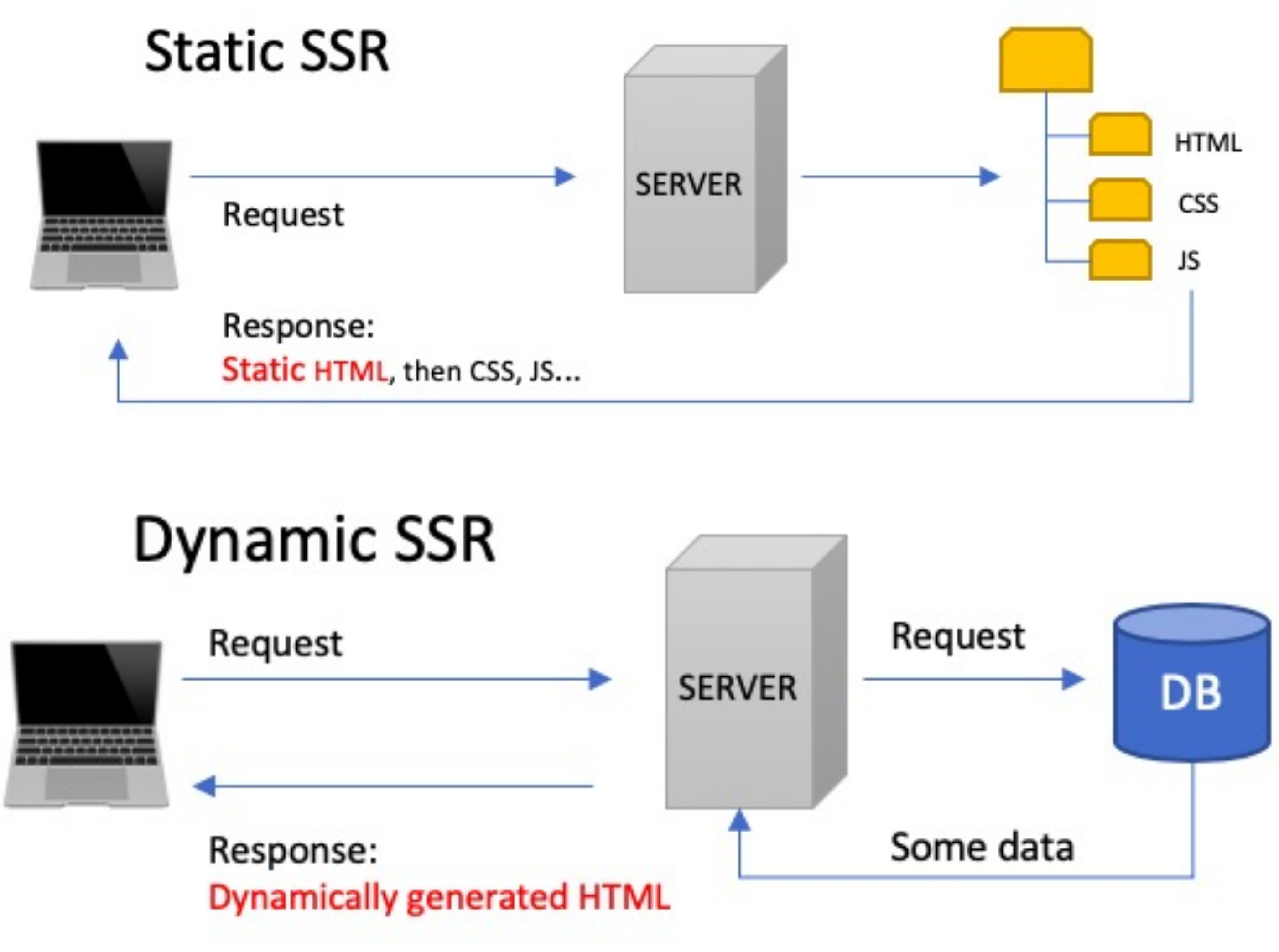
Angular.js
Designing
Views

Server Side
Node.Js

Express.js
Crud API

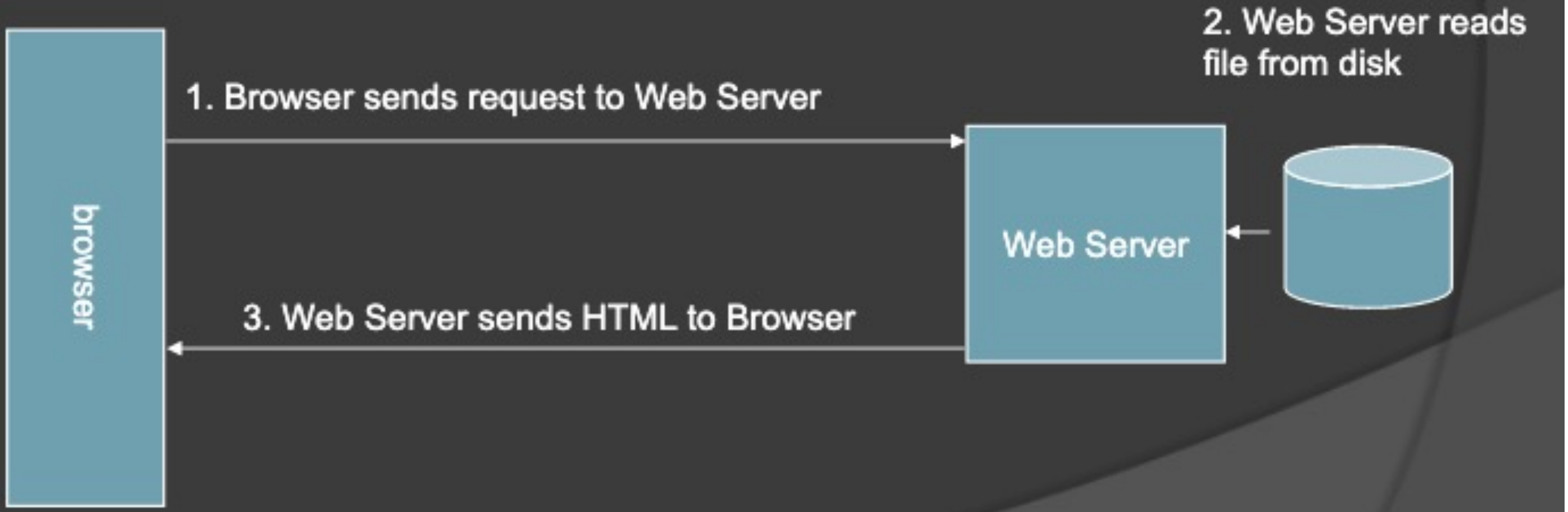
MongoDB
Storing Data

Web Server – Static HTML vs Dynamic



Static content

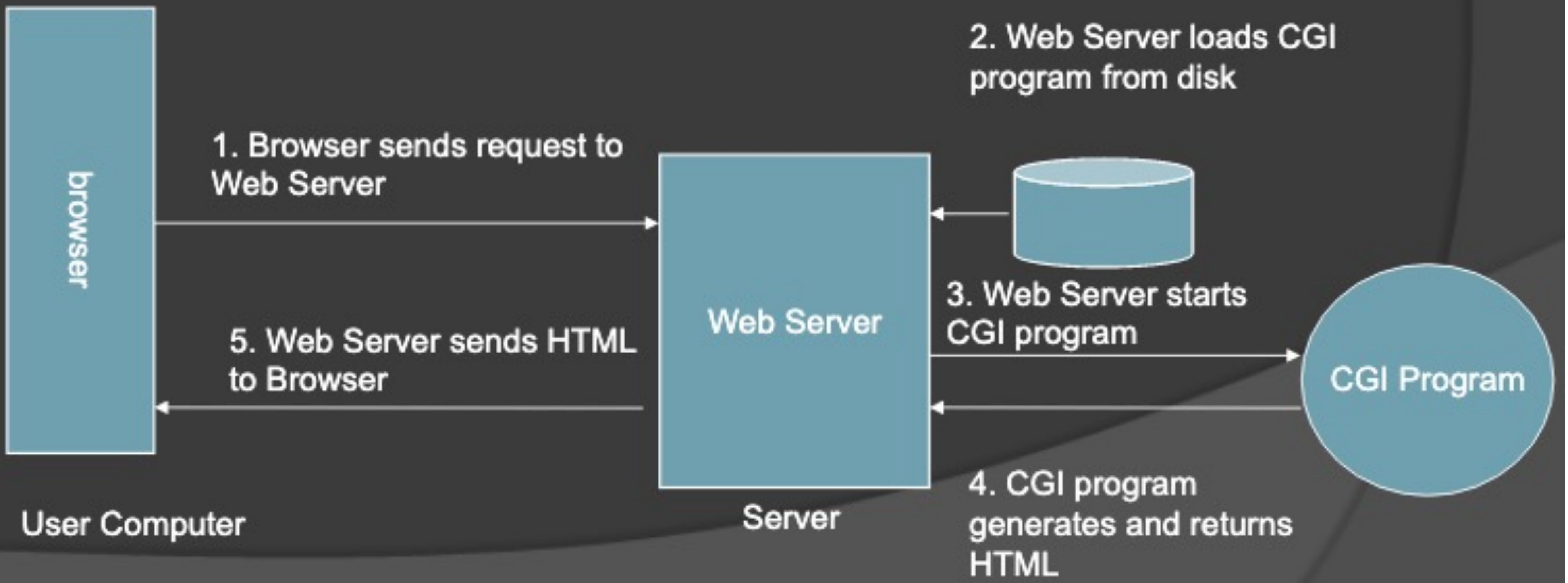
- Web Server delivers contents of a file (html)



2. Web Server reads file from disk
1. Browser sends request to Web Server
browser
Web Server
3. Web Server sends HTML to Browser

Dynamic Content

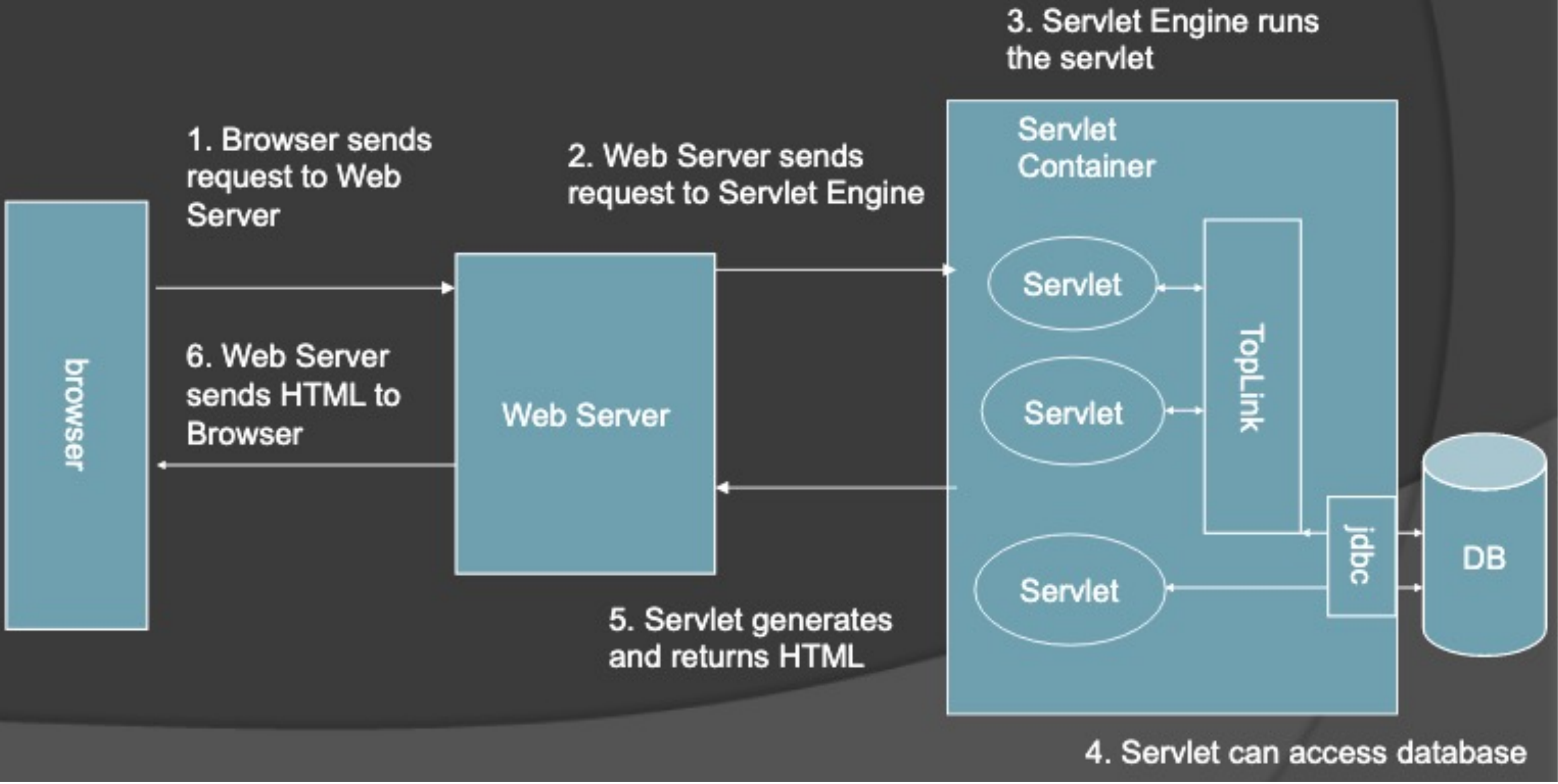
- CGI(Common Gateway Interface)program generates HTML that is returned to Browser



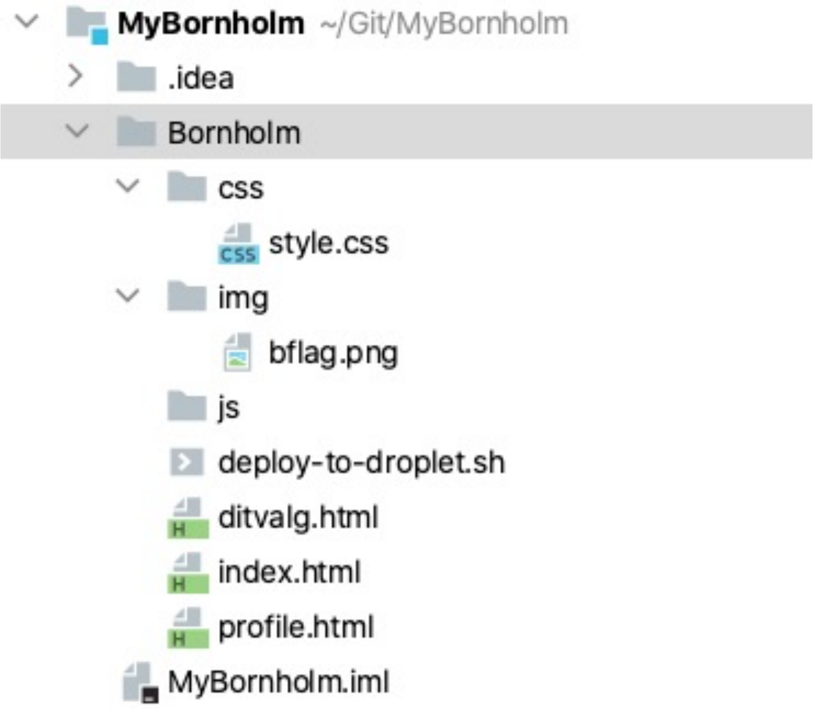
browser
User Computer
1. Browser sends request to Web Server
Web Server
Server
2. Web Server loads CGI program from disk
3. Web Server starts CGI program
CGI Program
4. CGI program generates and returns HTML
5. Web Server sends HTML to Browser

Java Servlet Diagram

- Extends Web Server


3. Servlet Engine runs the servlet
1. Browser sends request to Web Server
2. Web Server sends request to Servlet Engine
Servlet Container
browser
Web Server
Servlet
Servlet
TopLink
6. Web Server sends HTML to Browser
jdbc
DB
Servlet
5. Servlet generates and returns HTML
4. Servlet can access database

I gang med IntelliJ og Slyngelstuens SMS



```
MyBornholm ~/Git/MyBornholm
  .idea
  Bornholm
    css
      style.css
    img
      bflag.png
    js
    deploy-to-droplet.sh
    ditvalg.html
    index.html
    profile.html
  MyBornholm.iml
```