Fagligt Indhold for Studiepraktik
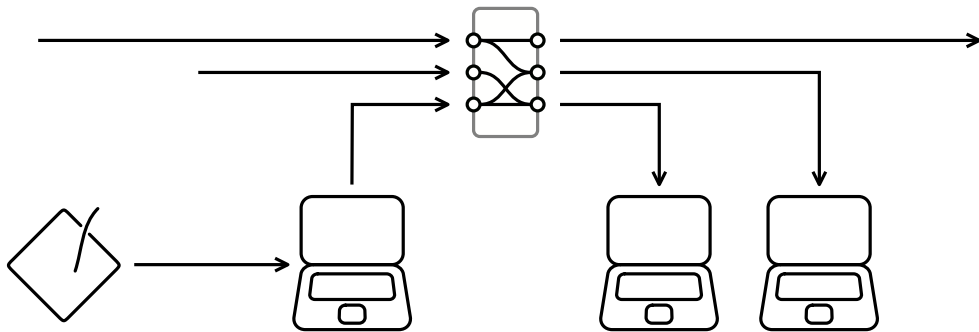## Software Engineering og Software Technology

Aslak Johansen asjo@mmmi.sdu.dk

Oktober 25-26, 2023

SDU✿

Part 0:
Introduktion

Overblik
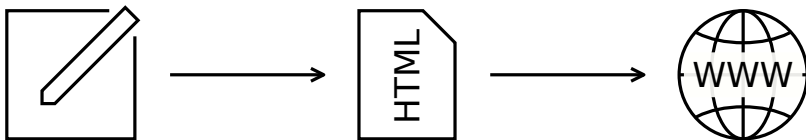

Vi skal lave et *dashboard*!

Repositorie

`https://github.com/aslakjohansen/sdu-sest-student-visit`

Tekst Editor

Vi skal skrive noget kode. Det gøres i en *tekst editor*.

	Windows	**OSX**	**Linux**
Installeret:	Notepad	TextEdit	...
Bedre valg:	Sublime Text	Sublime Text	Gedit/Kate/Sublime Text

Sublime Text download: `https://www.sublimetext.com/download`

Part 1:
HTML Dokumenter

Arbejde med HTML Filer

Hello, World

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Hello, World

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Hello, World - Indhold

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Hello, World - Metadata

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Hello, World - Dokumentets Titel

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Hello, World - Dokumentets Kodning

```html
<html>
  <head>
    <title>Hello, World</title>
    <meta charset="utf-8" />
  </head>
  <body>
    Hello, World
  </body>
</html>
```

Anatomien af et Tag

Overblik:

- ▶ Indkapsling: `<div>` ... `</div>`
- ▶ Navn: `<div>` ... `</div>` → `div`
- ▶ Shorthand hvis indhold er tomt: `<div></div>` → `<div />`
- ▶ Parametre: `<div a="1" b="2" />`
- ▶ Identitet: `<div id="data" />`

Dokumentet organiseres ved at

- ▶ placere tags indeni hinanden.
- ▶ placere tags efter hinancen, eventuelt med tekst imellem.

Eksempel

```html
<html>
  <head>
    <title>Eksempel</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>1 Introduktion</h1>
    <h2>1.1 Problembeskrivelse</h2>
    <p>Hvordan kan man give elever fra en gymnasiel uddannelse en
    ↪   introduktion til Software Teknologi <b>(ST)</b> og Software
    ↪   Engineering <b>(SE)</b> uddannelserne?</p>
    <h2>1.2 Tilgang</h2>
    <p>Vi <i>prøver</i> os frem!</p>
  </body>
</html>
```
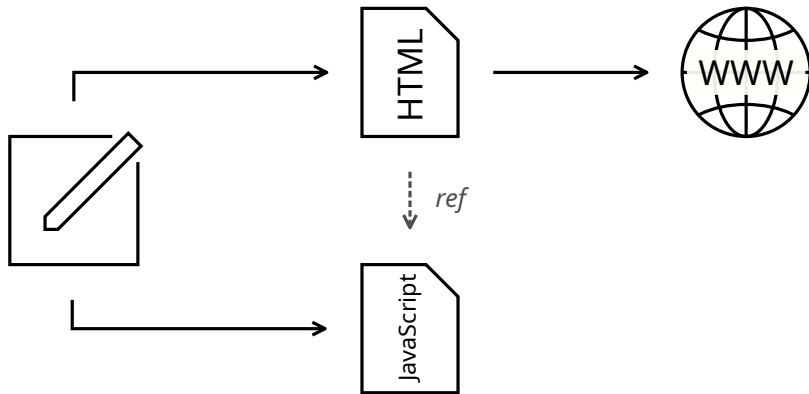
Part 2:
Logik med JavaScript

JavaScript?

Et programmeringssprog der

- ▶ primært er designet til at kunne afvikles i en browser.
- ▶ kan manipulere den HTML datastruktur der vises i browseren.
- ▶ kan reagere på handlinger på hjemmesiden.
- ▶ kan interagere med omverdenen.
- ▶ kan udvides igennem moduler.

Arbejde med Javascript Filer

Dokument

```html
<html>
  <head>
    <title>Test</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Test</h1>

    <div id="data" />

    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

Dokumentet

```html
<html>
  <head>
    <title>Test</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Test</h1>

    <div id="data" />

    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

Dokumentet

```html
<html>
  <head>
    <title>Test</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Test</h1>

    <div id="data" />

    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```
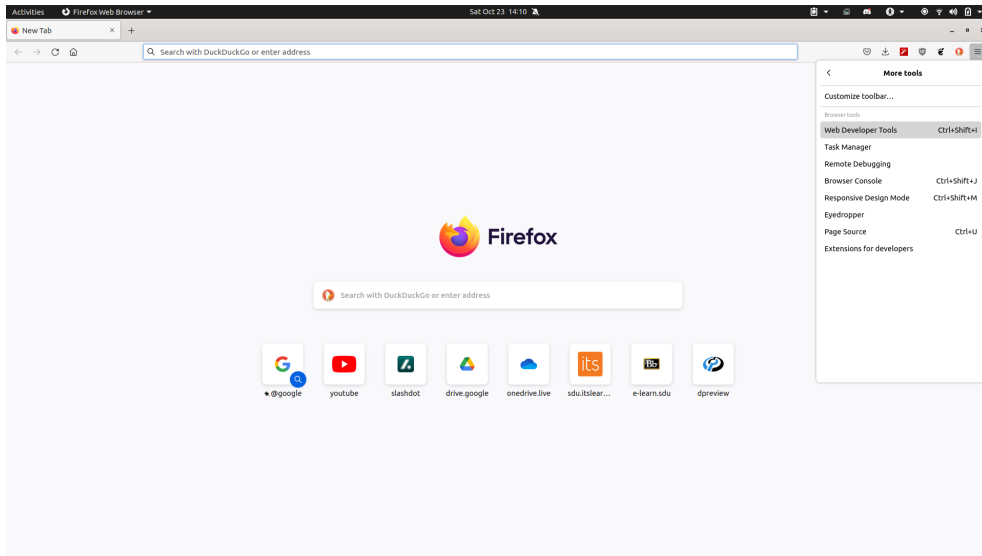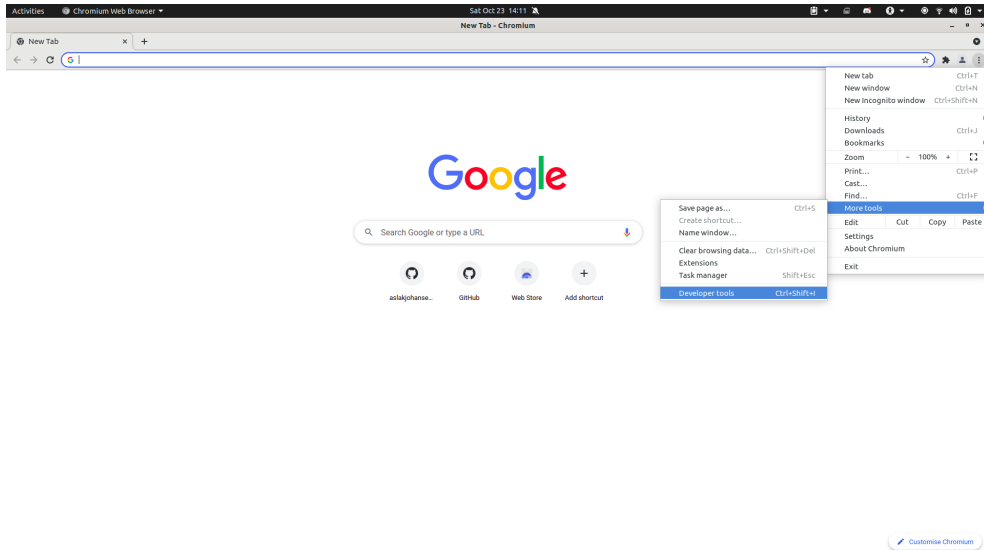
Udvikling - Firefox

Udvikling - Chrome og Chromium

Konsollen

Vores første JavaScript program:

```
console.log("Hello, World");
```

Logik

```
console.log("Hello, World");

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    console.log("Cirkel med radius "+radius+" har arealet "+area);
  } else {
    console.log("Cirkel med radius "+radius+" har for stort et areal");
  }
}
```

Logik

```
console.log("Hello, World");

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    console.log("Cirkel med radius "+radius+" har arealet "+area);
  } else {
    console.log("Cirkel med radius "+radius+" har for stort et areal");
  }
}
```

Logik

```
console.log("Hello, World");

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    console.log("Cirkel med radius "+radius+" har arealet "+area);
  } else {
    console.log("Cirkel med radius "+radius+" har for stort et areal");
  }
}
```

Logik

```
console.log("Hello, World");

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    console.log("Cirkel med radius "+radius+" har arealet "+area);
  } else {
    console.log("Cirkel med radius "+radius+" har for stort et areal");
  }
}
```

Manipulation med HTML

```javascript
document.getElementById("data").innerHTML = "<p>Hello, World</p>";

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    entry = "<p>Cirkel med radius "+radius+" har arealet
    ↪  "+area+"</p>";
  } else {
    entry = "<p>Cirkel med radius "+radius+" har for stort et
    ↪  areal</p>";
  }
  document.getElementById("data").innerHTML += entry
}
```

Manipulation med HTML

```javascript
document.getElementById("data").innerHTML = "<p>Hello, World</p>";

pi = 3.14
rs  = [1, 2, 3, 5, 8, 13]
for (i=0 ; i<rs.length ; i++) {
  radius = rs[i];
  area = pi*radius*radius;
  if (area<100) {
    entry = "<p>Cirkel med radius "+radius+" har arealet
    ↪  "+area+"</p>";
  } else {
    entry = "<p>Cirkel med radius "+radius+" har for stort et
    ↪  areal</p>";
  }
  document.getElementById("data").innerHTML += entry
}
```
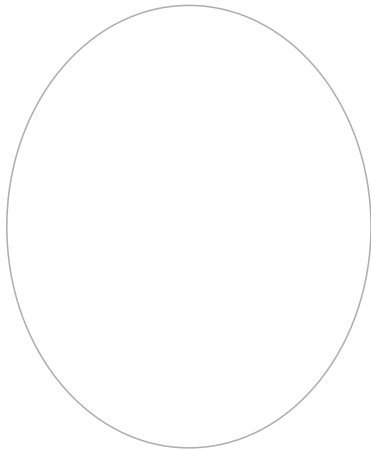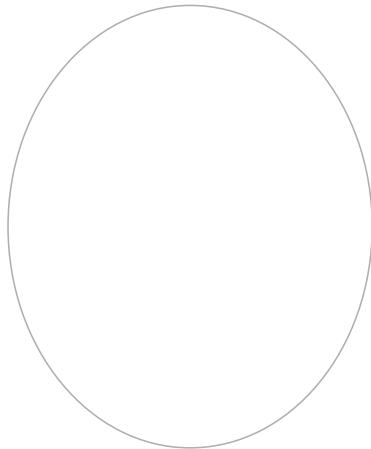
Part 3:
Strukturerede Data

Datastrukturer - JavaScript Objekter

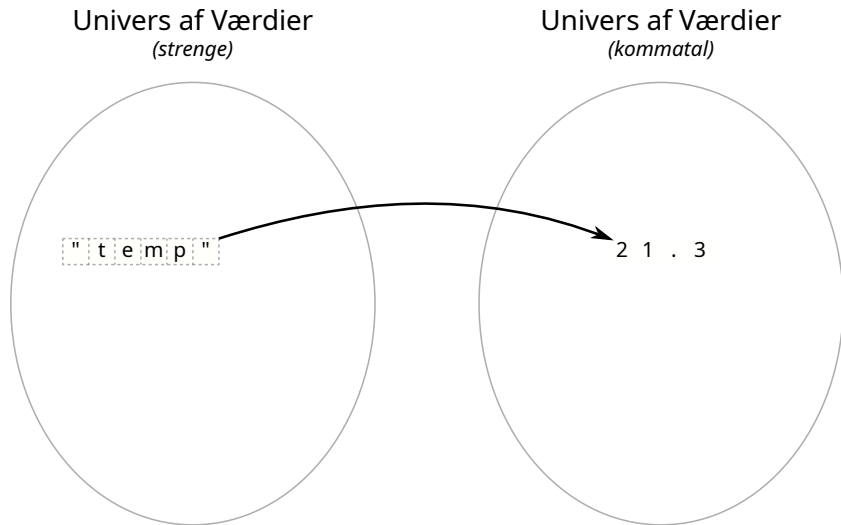### Univers af Værdier
(strenge)

Univers af Værdier
(kommatal)

Datastrukturer - JavaScript Objekter

Univers af Værdier
(strenge)

Univers af Værdier
(kommatal)

" t e m p "

2 1 . 3

Datastrukturer - JavaScript Objekter

Datastrukturer - JavaScript Objekter, Anvendelse af

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

Datastrukturer - JavaScript Objekter, Anvendelse af

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

Datastrukturer - JavaScript Objekter, Anvendelse af

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

Datastrukturer - JavaScript Objekter, Anvendelse af

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

Datastrukturer - JavaScript Objekter, Anvendelse af

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

Tekststrenge

En tekststreng (eller bare *string*) er en sekvens af tegn.

Vi har før set denne:

Hello, World!

Men vi kan også oprette denne:

{"temp":21.3, "hum":61.7}

JSON Formatet

```javascript
data = {"temp": 21.3, "hum": 61.7};
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

JSON Formatet

```javascript
data = JSON.parse('{"temp": 21.3, "hum": 61.7}');
result = "<p>Indhold:</p>";

keys = Object.keys(data);
result += "<ul>";
for (i=0 ; i<keys.length ; i++) {
  key = keys[i];
  value = data[key];
  result += "<li><b>"+key+"</b> peger på <i>værdien</i> "+value+".</li>";
}
result += "</ul>";

if ('temp' in data)
  result += "<p>data inddeholder nøglen 'temp'</p>";
if ('wind' in data)
  result += "<p>data inddeholder nøglen 'wind'</p>";

document.getElementById("data").innerHTML = result;
```

JSON er en måde at konvertere en datastruktur til en streng, og tilbage igen.

Part 4:
Data med Publish Subscribe

Modtagelse af Beskeder - HTML Fil

```html
<html>
  <head>
    <title>Dashboard</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Test</h1>

    <div id="data" />

    <script
    ↪   src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"
    ↪   type="text/javascript"></script>
    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```
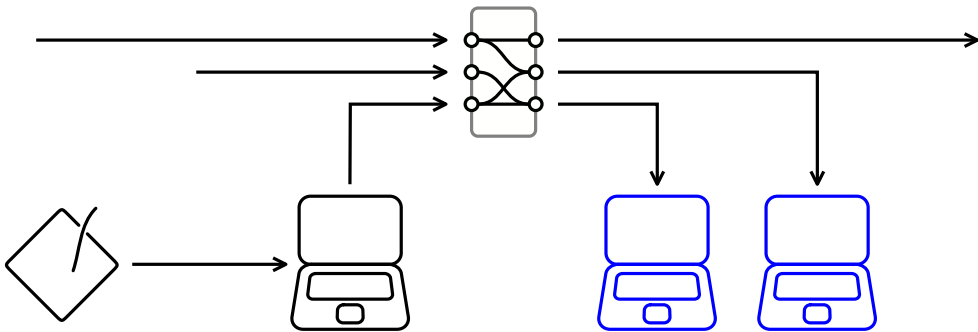
Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  document.getElementById("data").innerHTML = message.payloadString;
}
```

Behandling af Beskeder

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+value+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```

Behandling af Beskeder

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+value+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```

Behandling af Beskeder

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+value+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```

Behandling af Beskeder

```javascript
client_id = "client"+Date.now();
topic = "dk/sdu/sest/test";

// Create a client instance
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe(topic);
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+value+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```
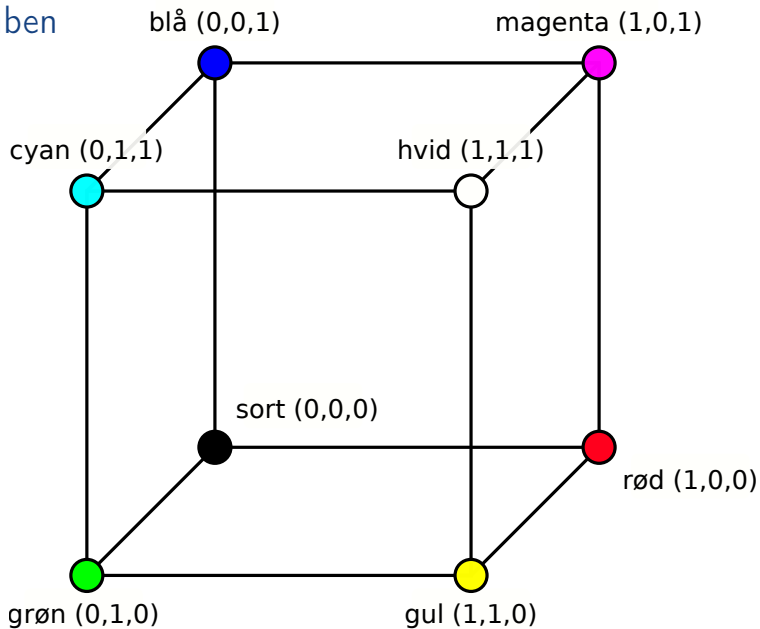
Part 5:
Farver

RGB Farvekuben

blå (0,0,1)   magenta (1,0,1)

cyan (0,1,1)   hvid (1,1,1)

sort (0,0,0)

rød (1,0,0)

grøn (0,1,0)   gul (1,1,0)

Hexadecimal Repræsentation

Altså, en farve i en computer er et punkt i et 3d koordinatsystem med en grundfarve på hver akse, og hvor hver akse spænder mellem 0 og 1.

I praksis arbejder man med 8 bit per *kanal*. Dette betyder at hver akse (aka kanal) inddeles i $2^8 = 256$ niveauer. Dette giver $256^3 = 16777216$ forskellige farver.

I HTML filer beskrives disse værdier typisk hexadecimalt, altså ved hjælp af et 16-tals system.

Der følges formatet "`rrggbb`". I hver farves kodning repræsenterer det højre tegn en værdi mellem 0 og 15 (herunder $a \mapsto 10, b \mapsto 11, c \mapsto 12, d \mapsto 13, e \mapsto 14, f \mapsto 15$), og det venstre tegn er 16 gange så meget værd.

`000000` `ff0000` `00ff00` `0000ff` `ffff00` `00ffff` `ff00ff`

Tærskelværdier

```javascript
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, "client"+Date.now());
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```

Tærskelværdier

```javascript
client = new Paho.MQTT.Client("broker.hivemq.com", 8000, "client"+Date.now());
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}

function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";

  document.getElementById("data").innerHTML = result;
}
```

Part 6:
Plots

HTML Fil

```html
<html>
  <head>
    <title>Dashboard</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Test</h1>

    <div id="data"></div>
    <div id="plot" style="width:600px;height:250px;"></div>

    <script
    ↪   src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"
    ↪   type="text/javascript"></script>
    <script src="https://cdn.plot.ly/plotly-2.4.2.min.js"></script>
    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

JavaScript Fil

```javascript
times = [];
temps = [];
starttime = Date.now()

client = new Paho.MQTT.Client("broker.hivemq.com", 8000,
                             "client"+starttime);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}
```

```javascript
function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";
  document.getElementById("data").innerHTML = result;

  times.push((Date.now()-starttime)/1000);
  temps.push(payload["temperature"]);

  tag = document.getElementById('plot');
  Plotly.newPlot(tag, [{
          x: times,
          y: temps }], {
        margin: { t: 0 } } );
}
```

JavaScript Fil

```javascript
times = [];
temps = [];
starttime = Date.now()

client = new Paho.MQTT.Client("broker.hivemq.com", 8000,
                              "client"+starttime);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}
```

```javascript
function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";
  document.getElementById("data").innerHTML = result;

  times.push((Date.now()-starttime)/1000);
  temps.push(payload["temperature"]);

  tag = document.getElementById('plot');
  Plotly.newPlot(tag, [{
          x: times,
          y: temps }], {
        margin: { t: 0 } } );
}
```

JavaScript Fil

```javascript
times = [];
temps = [];
starttime = Date.now()

client = new Paho.MQTT.Client("broker.hivemq.com", 8000,
                              "client"+starttime);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}
```

```javascript
function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";
  document.getElementById("data").innerHTML = result;

  times.push((Date.now()-starttime)/1000);
  temps.push(payload["temperature"]);

  tag = document.getElementById('plot');
  Plotly.newPlot(tag, [{
          x: times,
          y: temps }], {
        margin: { t: 0 } } );
}
```

JavaScript Fil

```javascript
times = [];
temps = [];
starttime = Date.now()

client = new Paho.MQTT.Client("broker.hivemq.com", 8000,
                             "client"+starttime);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}
```

```javascript
function onMessageArrived(message) {
  payload = JSON.parse(message.payloadString);
  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";
  document.getElementById("data").innerHTML = result;

  times.push((Date.now()-starttime)/1000);
  temps.push(payload["temperature"]);

  tag = document.getElementById('plot');
  Plotly.newPlot(tag, [{
          x: times,
          y: temps }], {
       margin: { t: 0 } } );
}
```

Part 7:
Test med Manuel Publicering

Modtagelse af Beskeder - HTML Fil

```html
<html>
  <head>
    <title>Test Producer</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Publicér til broker.hivemq.com:8000</h1>

    <p>Topic: <input type="text" id="topic" value="dk/sdu/sest/test"/></p>
    <p>Besked:</p>
    <textarea id="payload" rows=10 cols="60">{"temperature": 16,
    "Humidity":53}</textarea>
    <p><button onclick="clicked()">Send</button></p>

    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"
    type="text/javascript"></script>
    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

Modtagelse af Beskeder - HTML Fil

```html
<html>
  <head>
    <title>Test Producer</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Publicér til broker.hivemq.com:8000</h1>

    <p>Topic: <input type="text" id="topic" value="dk/sdu/sest/test"/></p>
    <p>Besked:</p>
    <textarea id="payload" rows=10 cols="60">{"temperature": 16,
    ↪  "Humidity":53}</textarea>
    <p><button onclick="clicked()">Send</button></p>

    <script
    ↪  src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"
    ↪  type="text/javascript"></script>
    <script src="logic.js" type="text/javascript"></script>
  </body>
</html>
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
function clicked() {
  client_id = "mqtt_producer_"+(new Date()).getTime()
  topic   = document.getElementById("topic").value
  payload = document.getElementById("payload").value

  function callback () {
    msg = new Paho.MQTT.Message(payload);
    msg.destinationName = topic;
    client.send(msg);
  }

  // Create a client instance
  client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
  client.connect({onSuccess:callback});
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
function clicked() {
  client_id = "mqtt_producer_"+(new Date()).getTime()
  topic   = document.getElementById("topic").value
  payload = document.getElementById("payload").value

  function callback () {
    msg = new Paho.MQTT.Message(payload);
    msg.destinationName = topic;
    client.send(msg);
  }

  // Create a client instance
  client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
  client.connect({onSuccess:callback});
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
function clicked() {
  client_id = "mqtt_producer_"+(new Date()).getTime()
  topic   = document.getElementById("topic").value
  payload = document.getElementById("payload").value

  function callback () {
    msg = new Paho.MQTT.Message(payload);
    msg.destinationName = topic;
    client.send(msg);
  }

  // Create a client instance
  client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
  client.connect({onSuccess:callback});
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
function clicked() {
  client_id = "mqtt_producer_"+(new Date()).getTime()
  topic   = document.getElementById("topic").value
  payload = document.getElementById("payload").value

  function callback () {
    msg = new Paho.MQTT.Message(payload);
    msg.destinationName = topic;
    client.send(msg);
  }

  // Create a client instance
  client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
  client.connect({onSuccess:callback});
}
```

Modtagelse af Beskeder - JavaScript Fil

```javascript
function clicked() {
  client_id = "mqtt_producer_"+(new Date()).getTime()
  topic   = document.getElementById("topic").value
  payload = document.getElementById("payload").value

  function callback () {
    msg = new Paho.MQTT.Message(payload);
    msg.destinationName = topic;
    client.send(msg);
  }

  // Create a client instance
  client = new Paho.MQTT.Client("broker.hivemq.com", 8000, client_id);
  client.connect({onSuccess:callback});
}
```

Robusthed

Hvad sker der hvis vi sender strengen "`{"temperature":`" i stedet for et gyldigt JSON objekt?

Vores modtager holder op med at reagere!

Et program siges at være *robust* hvis det reagerer hensigtsmæssigt (e.g., det ikke crasher) når det gives fejlbehæftede input.

JavaScript Fil

```javascript
times = [];
temps = [];
starttime = Date.now()

client = new Paho.MQTT.Client("broker.hivemq.com", 8000,
                             "client"+starttime);
client.onMessageArrived = onMessageArrived;
client.connect({onSuccess:onConnect});

function onConnect() {
  client.subscribe("dk/sdu/sest/test");
}

function colorize(type, value) {
  if (type==="temperature" && value>25) {
    return '<span style="color:#bb2200;">'+value+'</span>';
  } else {
    return value;
  }
}
```

```javascript
function onMessageArrived(message) {
  try {
    payload = JSON.parse(message.payloadString);
  } catch(_) {
    return;
  }

  keys = Object.keys(payload);

  result = "<p>Status:</p>";
  result += "<ul>";
  for (i=0 ; i<keys.length ; i++) {
    key = keys[i];
    value = payload[key];
    result += "<li><b>"+key+":</b> "+colorize(key, value)+"</li>";
  }
  result += "</ul>";
  document.getElementById("data").innerHTML = result;

  times.push((Date.now()-starttime)/1000);
  temps.push(payload["temperature"]);

  tag = document.getElementById('plot');
  Plotly.newPlot(tag, [{
        x: times,
        y: temps }], {
        margin: { t: 0 } } );
}
```

Part 8:
Chat System

Øvelse

- ▶ Vælg et nyt topic navn.
- ▶ Lav en HTML side med en knap og et tekstfelt. Når knappen aktiveres skal indholdet af tekstfeltet publiceres til det valgte topic.
- ▶ Opret en subscription til det valgte topic. Når data ankommer på dette topic vises det et sted på siden.

Spørgsmål?