

Clientsideprogramming

Fagnr.: 16476
Underviser Rasmus Dal

RADA - Februar 2020



SYDDANSK
ERHVERVSSKOLE

General information

- Faget **Clientsideprogrammering** har en varighed på 15 skoledage og afsluttes med en standpunktskarakter.
Faget er spredt ud over to skoleperioder: H1 (5 dage) & H2 (10 dage).
- Arbejdsformer på denne skoleperiode:
 - **Teoretisk gennemgang:** Hvor læreren giver oplæg eller kommer med eksempler på projektor eller tavle.
 - **Praktisk arbejde:** hvor eleven arbejder selvstændigt eller i grupper med den stillede opgaver.
- På dette hovedforløb afsluttes faget med en **individuel evalueringssamtale** med din lærer.

Denne evaluering er tidssat til ca. **10 min** inkl. spørgsmål.

Her skal du demonstrere din færdige opgave og fortælle om din viden og forståelse af faget og fagets læringsmål.

Din lærer vil stille uddybende spørgsmål og du vil på modtage en **delkarakter**.

Den endelige standpunktskarakter gives på H2, hvor faget afsluttes.

Niveauer og evaluering

- Faget **Clientsideprogrammering**, bedømmes på 3 niveauer: **Rutineret**, **Avanceret** og **Ekspert**.
- Det er valgfrit, hvilket niveau man vil evalueres på, men alle starter på **Rutineret**.
- Jo højere niveau, jo højere selvstændighedsgrad forventes, samt redegørelse for valg og beslutninger skal kunne uddybes og understøttes.
- Opgaven er den samme for alle, men ønsker man at tage dette fag på Avanceret og Ekspertniveau, forventes der at opgaven laves med et stort engagement, og med et løsningsforslag som ligger ud over målene.
- Desuden forventes der, at man kan begrunde sin løsning fagligt i alle relevante sammenhænge.
- Der lægges vægt på at du kan planlægge, tilrettelægge og udføre opgaven således at kvaliteten i processen og resultatet sikres.

Niveauer

Rutineret niveau: Eleven kan planlægge og gennemføre en opgave eller aktivitet eller løse et problem i en rutinemæssig eller kendt situation og omgivelse, alene og i samarbejde med andre. På dette niveau lægges vægt på den personlige kompetence til selvstændigt at sætte sig ind i mere komplicerede problemstillinger og til at kommunikere med andre om løsningen heraf. Yderligere lægges vægt på fleksibilitet og omstillingsevne.

Avanceret niveau: Eleven kan vurdere et problem, kan planlægge, løse og gennemføre en opgave eller aktivitet eller løse et problem også i ikke-rutinesituationer – alene eller i samarbejde med andre – under hensyn til opgavens art. På dette niveau lægges vægt på den personlige kompetence til at tage selvstændigt ansvar og vise initiativ samt kompetence til selv at formulere og løse faglige og sociale opgaver og problemer.

Yderligere lægges vægt på kvalitetssans og kreativitet.

Stk. 2. Fag, der er beskrevet på avanceret niveau, kan beskrives med følgende yderligere præstationsstandard, som er valgfri for eleverne:

Ekspertniveau: Eleven kan løse komplekse arbejdsopgaver og kan argumentere for valgte løsninger af opståede problemer. Der lægges vægt på evnen til at kunne bruge allerede opnåede kompetencer i en ny kontekst, på evnen til at arbejde med overblik og deltage i arbejdspladsens innovative processer. På dette niveau lægges vægt på den personlige kompetence til målrettet at kunne planlægge, tilrettelægge, udføre og evaluere arbejdsprocesserne således, at kvaliteten i processen og resultatet sikres. Endvidere lægges vægt på, at eleven kan vurdere og begrunde behovet for forbedringer af arbejdsprocesser, samt på at eleven kan kommunikere om sin faglighed i alle relevante sammenhænge.

Forventninger

- Jeg forventer af dig, at du er arbejdsom, møder til tiden og **deltager aktivt** i timerne.
- Jeg forventer at du **overholder pausetiderne**.
- Jeg forventer at du lytter **opmærksomt** til oplæg og **stiller spørgsmål**, når du kommer i tvivl.
- Når du arbejder med de praktiske opgaver, er du altid velkommen til at **stille spørgsmål**.
- Ræk bare hånden på 😊
- **HUSK: Man kan ikke spørge for meget, men man kan spørge for lidt – det er din uddannelse og dit ansvar at lære noget!**

Læringsmål:

– hvad lærer du I faget?

I faget clientsideprogrammering på H1, skal du:

- Lære at lave en responsiv hjemmeside med HTML, CSS og et CSS/JavaScript frameworks som Bootstrap
- Lærer om Client-Server modellen, herunder HTTP-protokollen
- Lærer grundlæggende HTML struktur og semantik
- Lærer at anvende HTML formularer
- Lærer grundlæggende CSS, herunder CSS selectors

Bedømmelseskriterier:

– hvordan bedømmes du I faget?

I faget clientsideprogrammering på H1, bedømmes du på flg.:

- Din forståelse af og evne til at forklare Client-Server modellen, herunder HTTP-protokollen
- Din forståelse af og evne til at forklare HTML:
Herunder: struktur, semantik og formularer
- Din forståelse af og evne til at forklare CSS:
Herunder selectors og struktur
- Din evne til at lave en responsiv hjemmeside med HTML, CSS og evt. et CSS/JavaScript framework (eks. Bootstrap)

SPØRGSMÅL ?

- **Hvad er Internettet ?**
- **Hvad er WWW ?**
- **Hvad er en server / webserver ?**
- **Hvad er en klient ?**

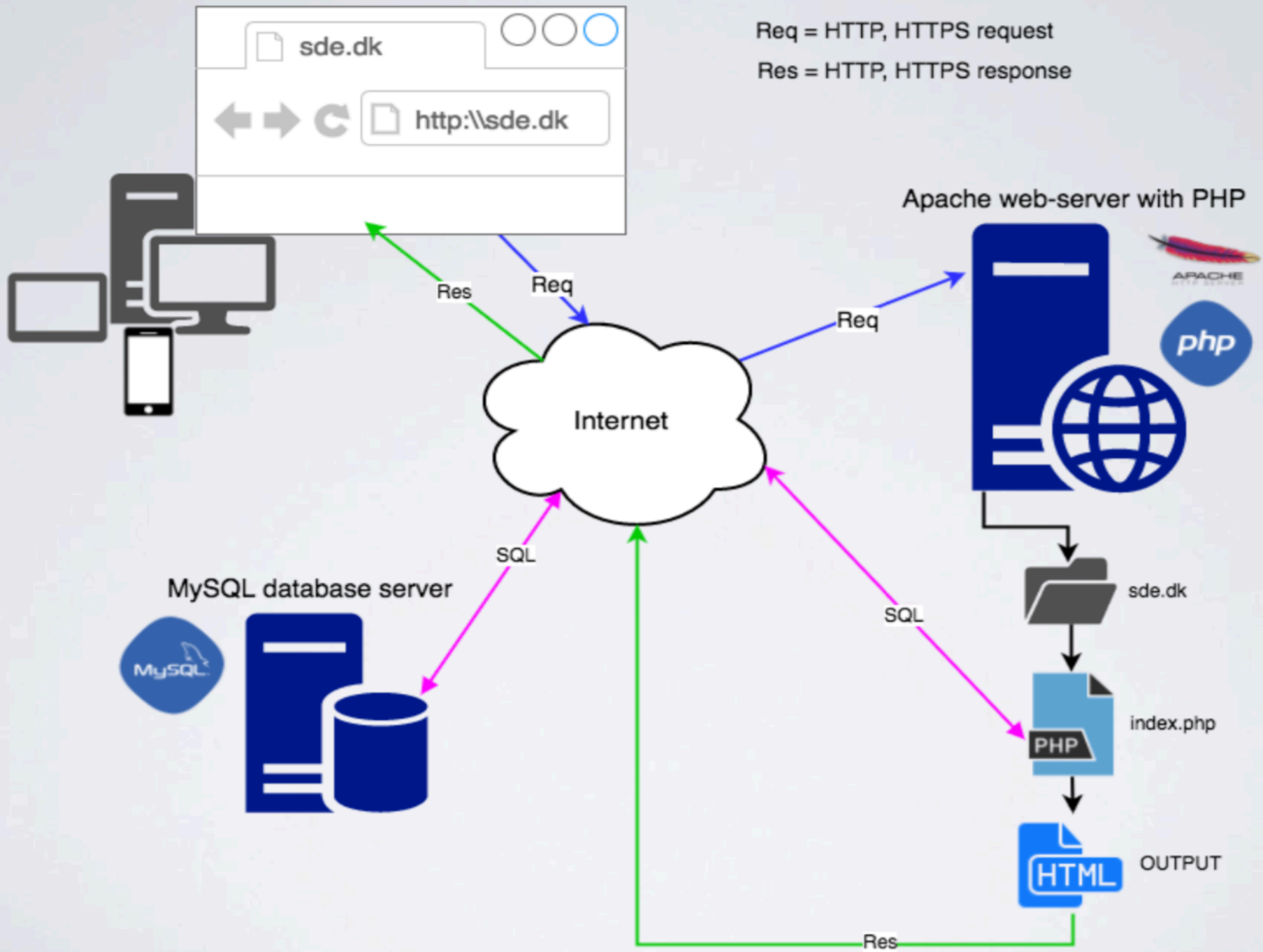
HTTP og Client Server



SYDDANSK
ERHVERVSSKOLE

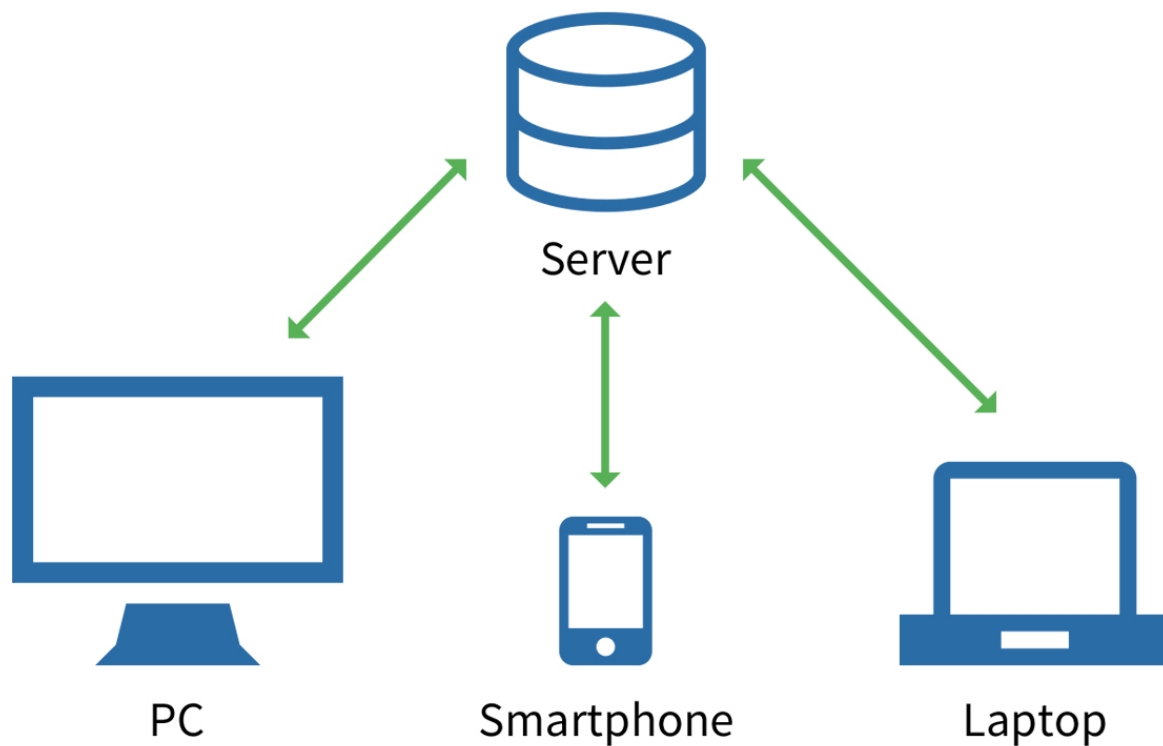
HTTP-protokollen

- **HTTP** eller **H**yper **T**ext **T**ransfer **P**rotocol er en **kommunikationsprotokol**, som primært bruges til kommunikation på **World Wide Web** (WWW) over Internettet.
- **HTTP ressource**: Alle ressourcer på www (eks. hjemmesider) adresseres med en URL og et. en URI.
URL (**U**niform **R**essource **L**ocator) sørger for at lokalisere, hvor ressourcen findes eks.: *http://sde.dk* (DNS for IP adressen på den webserver, der hoster sde's hjemmeside)
URI (**U**niform **R**essource **I**dentifier) er den del af en adresse, der udpeger en bestemt ressource på eks. en hjemmeside: *http://sde.dk/**kurser**/*
- **HTTP Request**: En klient sender et **HTTP request** efter en ressource på en webserver via en adressering med en URL og evt. en URI.
- **HTTP Response**: En webserver svarer på et HTTP request ved at finde og forberede den ressource, der efterspørges, hvorefter den sender det retur via et **HTTP Response**.
- **Stateless** (Tilstandsløs): HTTP er **stateless**, hvilket vil sige, at den ikke bibeholder en dataforbindelse mellem klient og server efter udførelse af et HTTP request og HTTP response.
- **HTTP Status koder**: Serveren svarer med HTTP-protokolkode, for at angive det overordnede resultat af et HTTP request, og derefter typisk selve ressourcen (fx et HTML-dokument eller et billede) eller en fejlbesked.
Eks.: *200 OK* eller *404 Not Found*.



Client-Server modellen

Client-Server Model



Client-Server modellen

I en klient-server-arkitektur deler man et it-system op i en klientdel og i en serverdel.

Klienten er ansvarlig for

- præsentation af grafik, tekst, og information til brugeren
- interaktion med brugeren (museklik, tastatur, osv.)
- kommunikation af information til og fra serveren

Serveren er ansvarlig for

- opbevare og redigere information/data som deles af klienterne
- foretage simple procedurer og beregninger på informationen
- sende og modtage information til og fra klienterne

Client-Server modellen

- Et klient-server it-system har én server og mange klienter.
- Klienterne er de "aktive" forstået således, at det altid er dem, der starter en forespørgsel eller ændring af data.
- Serveren er "reaktiv" forstået således, at den ikke selv tager initiativ til noget, men venter på forespørgsler fra klienter. En forespørgsel vil bevirke, at serveren udfører procedurer, ændrer data, og til sidst sender et svar tilbage til klienten.
- Information flyder mellem klienten og serveren via en netværksforbindelse, typisk internettet.
- Klienter kan ikke sende og modtage information fra andre klienter: alting sker via serveren.

HTML



SYDDANSK
ERHVERVSSKOLE

HTML

- **H**yper **T**ext **M**arkup **L**anguage (**HTML**)
- Markup sprog: At opmærke, definere indhold på websider
- Beskriver strukturen på websider ved hjælp af markering
- HTML elementer er byggestenene på HTML sider
- Man gør det ved hjælp af <tags>

HTML opbygningen

Se opbygningen af et HTML dokument som en "sandwich":

```
<!DOCTYPE html> Fortæller browseren hvilken version af HTML (5) du
bruger

<html> Brødet i sandwichen, øverste bolle

  <head> Tilbehør i sandwichen, Mayo, Bacon osv.
    <meta charset="UTF-8"> Globalt Karaktersæt inkludere Æ, Ø og Å
    <title>16476 - Clientsideprogrammering</title>
  </head>

  <body> Kødet i sandwichen / Alt det du ser
    <div>
      <h1>Overskrift 1</h1>
      <p>Hello world!</p>
    </div>
  </body>

</html> Brødet i sandwichen, nederste bolle
```

HTML attributter

- https://www.w3schools.com/html/html_attributes.asp
- Der findes flere attributter som kan bruges på alle HTML tags
- Link elementet `<a>` er en af dem, hvor man sætter en `href` på som angiver den URL browseren styrer hen til
`Syddansk Erhvervsskole`

HTML Semantik

- https://www.w3schools.com/html/html5_semantic_elements.asp

Strukturering

- Oprettelse af ordentligt strukturerede sider er grundlaget for enhver god hjemmeside.
- En solid semantisk struktur giver dig mulighed for at formidle meningen og indholdet på siden.
Dette gør din indhold mere tilgængelig, lettere at indeksere og søge, og gør det nemmere at genbruge.
- Husk, at der er mange forskellige enheder, robotter og andre brugere, som vil læse din kode.
- Uden en konsekvent semantisk struktur bliver dit websted mindre effektivt.



CSS



SYDDANSK
ERHVERVSSKOLE

CSS

- **Cascading Style Sheets**
- Manipulerer standard udseendet af html elementer
- In-document CSS kan skrives som en del af et HTML dokument ved brug af HTML tagget *style*:
`<style>font-size: 12px;</style>`
- In-line CSS Kan skrives som en attribut inde i HTML koden:
`<p style="font-size: 12px;"></p>`
- Kan også skrives i særskilte filer, som man blot linker til i HTML:

```
<!DOCTYPE html>
<html lang="da">
  <head>
    <meta charset="UTF-8" />
    <meta name="description" content="En side til at vise faget
Clientsideprogrammering">
    <title>16476 - Clientsideprogrammering</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

CSS selectors attributter

- Der findes flere attributter som kan bruges på alle HTML tags
- Link elementet `<a>` er en af dem, hvor man sætter en `href` på som angiver den URL browseren styrer hen til

```
<a href="http://www.sde.dk">Syddansk Erhvervsskole</a>
```

- Ligeledes kan man sætte attributter på som `id` og `class`, og som giver os en mulighed for at skrive CSS regler, disse bruges som *selector*
- *id* selectoren skrives i CSS med en havelåge `#`

```
#kasse1 {
```

```
}
```

- *class* selectoren skrives med et punktum `.`

```
.kasse {
```

```
}
```

- *id* selectoren gælder kun ved et specielt HTML element, hvor *class* kan genbruges til andre elementer

CSS/JavaScript framework



SYDDANSK
ERHVERVSSKOLE

Responsive Web Design



SYDDANSK
ERHVERVSSKOLE