Dokumentasjon

##### IT2 Eksamen

#### Kandidatnummer:

#### Fagkode:

#### Dato:

Innholdsfortegnelse

### Forside 1

### Innholdsfortegnelse 2

### Innledning 3

### Litteraturliste 3

### Programvareoversikt 3

### Utviklingsprosessen 4

### Mappestruktur 5

### Struktur for Besvarelse 5

### Besvarelse 6

### Oppgave 1 7

### Oppgave 2 8

### Oppgave 3 9

# Oversikt

## Litteraturliste:

* Kode 2 - Gløer O. Langslet, Yngve V. Eriksen, Anders Haugeto – Cappelen Damm.
* Annen litteratur (Ikke Hefter, Prøveark eller andre ark som vi fikk under prøven).

## Programvareoversikt

##### Operativssystem: macOS Sierra 10.12.6

##### Kodeeditor: Visual Studio Code Version 1.20.1 (1.20.1)

##### Kjøremiljø/Browser: Chrome Version 62.0.3202.94 (Official Build) (64-bit)

##### Andre Program:

##### Bildeeditor: Adobe Photoshop Version 19.0 Release, Apple Preview Version 10.0 (944.5)

##### Video og lyd editor: iMovie Version 10.1.9

## Biblioteker

##### Selvlaget javascript open source bibliotek: Openlib. Lenke til Github Repo: https://github.com/WilliamMRS/IT2\_Open\_Source\_Bibliotek

## Utviklingsprosessen

I IT2 har vi lært og brukt en trinnvis metode for programvareutvikling. Det finnes andre metoder å utvikle programvare på men denne prosessen inkluderer alle elementene. Vi har 5 Steg:

### Krav –> Design –> Kode –> Test –> Produksjon

**Krav:** Her spesifiserer man hva man ønsker å lage. Den som ønsker programmet laget, evt. Bestiller eller Programmerer må ha klarhet i alle spesifikasjonene og da gjerne laget et dokument med spesifikasjonene som både evt. Bestiller og programmerer er enige om og fullt forstår hva som står der.

**Design:** Her finner man ut hvordan løsningen skal bygges. Gjerne både teknisk og visuelt/brukerorientert.

Teknisk design beskriver den tekniske delen av hvordan programmet skal bygges. Hvordan filsystemet skal organiseres, filene skal snakke sammen og brukerinput. Det skal også bestemmes hvordan programmet skal organiseres i form av datastrukturer, funksjoner, objekter og variabler. Da bruker man også gjerne pseudokode for å lage en sann ca representasjon av hvordan programmet skal fungere før man skriver det inn som kode.

Brukerorientert design beskriver den brukerorienterte tekniske løsningsprosessen. Her beskriver man hvordan brukeren skal bruke programmet og hvordan brukere kan bruke programmet. Her lager man gjerne Wirelines og designer grove skisser over hvordan ting skal se ut og hva skal ligge hvor. I HTML er det vanlig å også inkludere piksel avstander og andre markører som <div> representert som bokser el.

**Kode:** Ut i fra designet og kravspesifikasjonen kodes det en løsning i form av kildekode. I IT2 bruker vi JavaScript, HTML og CSS. Her gjør man gjerne del-tester igjennom utviklingen for å forsikre seg om at enkelte deler av programmet fungerer, f.eks en funksjon. Man gjør gjerne de tre første fasene i flere iterasjoner for å finpusse og få så nært det man ønsker seg som mulig. Det følger også med en brukerdokumentasjon og en teknisk dokumentasjon av programmet. Brukerdokumentasjonen for de som bruker programmet og teknisk dokumentasjon for utviklere som skal vedlikeholde eller videreutvikle programmet.

**Test:** Testfasen er en arbeidsprosess hvor flere tester blir utført for å forsikre seg om at programvaren er opp til gitte standarder og oppfyller alle kravene uten problemer. Her letes det etter problemer med programmet slik de kan løses. Alle avvik og feil skal listes opp sammen med deres status slik at teamet kan komme i gang og fikse de og ha en oversikt over hva som funker og hva som ikke gjør det.

**Produksjon:** Her Slippes produktet til sluttbrukerne gjerne sammen med et versjonsnotat. Det er viktig at alle uferdige ressurser og ting som ikke er en del av den utgitte versjonen kun er tilgjengelig for utviklerne. Om det oppstår fatale feil eller andre større problemer i den utgitte versjonen kan man gjøre en ”rollback” og gå tilbake til forrige versjon for å fikse feil uten at sluttbrukerne blir mer påvirket.

## Mappestruktur

**Bibliotek:** Her ligger eventuelle Kodebibliotek jeg har brukt for å løse eksamensoppgaven.

**Dokumentasjon:** Dokumentasjon av oppgaven.

**Ressurser:** Bilder, Videoer og andre ressurser. Ofte kopiert og redigert fra vedlagte filer fra egen mappe.

**Oppgave 1:** Besvarelsen til Oppgave 1

**Oppgave 2:** Besvarelsen til Oppgave 2

**Oppgave 3:** Besvarelsen til Oppgave 3

Velger å strukturere oppgavene i mapper hver for seg for å unngå eventuelle konflikter. I en produksjons-situasjon ville det ferdige produktet vert samlet i en løsning.

## Struktur for besvarelse

### Index.html

Index.html er hjemmesiden til programmet. Her ligger all html kode og koblinger til forskjellige filer med css og javascript kode, som biblioteker eller programspesifikk kode.

### index.js

Her ligger programspesifikk javascript kode. Denne heter også index, men har en annen filending (js) for å spesifisere at her ligger javascript og skille filene. Det jeg oppnår med å ha samme navn er å ha orden i filene om det blir flere html og script filer i programmet i framtiden.

### index.css

Her ligger html filens statiske CSS kildekode. Denne filen gir programmet/nettsiden sitt pene utseende og bidrar til økt brukervennlighet.

### Mappe\_Ressurser

I mappen ressurser ligger bilder, videoer, data ol som programmet trenger for å fungere. Legger redigerte bilder, lyd og videoer i denne mappen istedenfor å tukle med vedleggs-mappen til oppgaven.

# Besvarelse

## Oppgave 1

### Krav

### Design

### Kode

### Produksjon

## Oppgave 2

### Krav

### Design

### Kode

### Produksjon

## Oppgave 3

### Krav

### Design

### Kode

### Produksjon