2. semester introduktion

# Introduktion 26.08.2019

Eksamen er en gruppeeksamen. For at kunne gå til eksamen, skal man have godkendt 3 obligatoriske opgaver (bundne forudsætninger) på Wiseflow. 3-5 pr. gruppe. Vi får udleveret en opgave, hvor vi skal skrive en gruppe rapport samt en projektopgave. Vi får tildelt en procesvejleder og en fagvejleder.

Se vigtige datoer i slide for bl.a. valgfag næste semester og eksamensdatoer.

Se semesteroversigt i slide.

Tema 7 – Dynamisk webdesign

# Introduktion til tema 7: Dynamisk webdesign

Det modsatte at statisk webdesign, som vi tidligere har arbejdet med. I stedet for selv at taste alt ind i HTML, skal vi ”hente” indhold *dynamisk indhold* til hjemmesiden. (Eksempel konorostorv.dk – alt er ikke skrevet ind, der er lavet en skabelon som henter indhold/data der bliver struktureret i skabelonen)

Cases som kan ses i slide.

Fag i temaet: Webudvikling (individuelt), UI/design og UX (gruppearbejde).

## Git, GITHUB og Brackets git – versionsstyring

Samarbejdsværktøj, redigeringshistorik. Repository (Repo).

### Programmeringssprog

Se slide.

Html og css er IKKE programmeringssprog. Men JavaScript er.

Variabler og konstanter – se slide.

## Betingelser (conditions) if-statements

*Logiske operatorer* (se slide) fx == (når man tildeler en værdi bruger man = fx ved let. == betyder er lig med), !=, < osv.

Se eksempler på brug af operatorer i slide.

*Forgrenede* betingelser: if – else if – else if – else

*Indlejerede* betingelser (nested if-statements): Hvis én betingelse er opfyldt, kan man stille andre betingelser.

## Funktioner

En måde at begrænse kode på og gøre det mere overskueligt.

Man kan bruge *parameter* i funktioner fx function findSum(a, b) hvor a og b er parametrene, som man så kan kalde ved at skrive *argumenterne* i funktionen fx findSum(8, 90); Man kan kalde funktionen mange gange med forskellige argumenter, så det er fleksibelt. – Se andre eksempler i slide. 1B Javascript.

# JavaScript og DOM

DOM = Document Object Model – Dokumentet er delt op i forskellige objekter. Struktur som JavaScript skal bruge. DOM træ.

Window console log (Browser Object Model) står udenfor DOM

Se slides.

Husk semantikken, brug ikke bare div altid.

Med JavaScript kan man vælge og ændre elementer samt tilføje og fjerne nye elementer. DOM manipulationer i script:

* Vælg et (eller flere) DOM-element: brug document.querySelector(””)
* Gem querySelector i en variabel: brug const
* Nyt indhold i element: der kan bruges textContent (til ren tekst) eller inner HTML (kan hackes – ”java politiet” vil ikke bruge det her :b)
* Nyt indhold med createElement og appendChild
  + Overblik over hvornår man skal bruge hvilke af 2 overstående i slide.

Flere DOM-manipulationer i slide.

## EventListener & EventHandler

Begreberne:

* DOMContentLoaded kan være et *event*.
* Hele sætningen document.addEventListener(”DOMContentLoaded”, start); er *EventListener*
* *Eventhandler* er når man skriver funktionen: function start()

Direkte manipulation af elementer – JS kan også overskrive CSS’en. Syntaks: bindestregen i CSS erstattes af camleCase x hedder font-family, font Family.1

Flere ting som jeg ikke kunne koncentrerer mig om…………………………………………..!!!!!!!!!!!!!!

# JavaScript-objekter og arrays af objekter

## Grit

Laver en skabelon for indholdet, skal styles i fx antal koloner og deres bredde. Hvis man definerer bredden i fraktion *fr****,*** så fylder den automatisk siden ud. Der skal være en container (hvor man definerer grid på) og noget indhold.

*Grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));*

## Objekter

Mere fleksible end variabler, de kan have flere egenskaber.

## Templates – skabeloner

Det er vigtigt at man ikke bruger ID men at man bruger class, da det er en skabebon som bliver klonet – gentaget mange gange – og det må man ikke med ID’er.