

# Dokumentasjon



# Hvorfor bør vi skrive dokumentasjon?

diskuter to og to

# Hvorfor

- Planlegging før koden skrives
- Beskrivelse av programmet
- Hvilke tester skal gjøres og hva er kriteriene
- Lettere for andre, eller deg selv om to år, og sette seg inn i programmet.
- Nyttig når programmet senere skal videreutvikles

# To hoveddeler av dokumentasjon

1. Produktdokumentasjon
2. Teknisk dokumentasjon

# Produktdokumentasjon

- Funksjonelle krav
  - brukerens ønsker og behov til produktet
  - hva skal nettsiden inneholde?
- Ikke-funksjonelle krav
  - backup
  - responstid
  - stabilitet
  - kompatibilitet

# Kravspesifikasjon

- Punktliste som inneholder korte, tydelige og testbare krav.  
For eksempel:
  - Bruker **skal** kunne registrere navn og adresse, og få informasjon om registreringen blir godkjent.
  - Animasjon **skal** vises ved oppstart av program
- Utarbeides sammen med kunden som skal ha nettsiden

# Skisser - wireframes

- enkel oversikt over brukergrensesnittet
- kan tegnes for hånd eller i et tegneprogram
- uttrykke en ide på et overordnet nivå
- godt utgangspunkt for en diskusjon
- ikke presist nok til å avdekke alle brukers behov

# Use case - bruksmønster

- viser gangen i produktet
- hendelser som bruker foretar seg
- lister, tabeller eller diagrammer (UML)
- enkleste variant: nummerert liste med de stegene brukeren går gjennom i programmet



# Testplan

- det finnes tester på mange ulike nivåer.
- i denne sammenheng:
  - teste om alle krav fra kravspesifikasjonen fungerer
  - dokumenteres for eksempel i en tabell

# Teknisk dokumentasjon

- kommentarer i koden
- pseudokode
- flytdiagram
- brukerveiledninger

# Kommentarer i koden

- hvorfor:
  - andre utviklere skal forstå
  - forstå selv hva du har gjort
  - “huskelapper” underveis
  - bortkommentering under testing og feilsøking
- korte kommentarer, en linje eller to: //
- lengre kommentarer: /\* \*/

# Pseudokode

- blanding av tekst og kode
- kan gjøres på mange forskjellige måter (ingen fasit)
- egen planlegging
- presentere en ide

# Eksempel Pseudokode

Funksjon som tar i mot to tall, legger de sammen og sjekker om svaret er positivt eller negativt

1. var tall1 =verdi fra input felt
2. var tall 2= verdi fra input felt
3. var sum = tall1 + tall2
4. hvis sum større en 0  
    skriv ut: Tallet er positivt  
ellers  
    skriv ut: Tallet er negativt

# Eksmepel Pseuokode - minibank

```
function uttak(beløp){  
    hvis saldo er mindre enn beløpet  
        skriv ut:Du har ikke dekning  
    ellers  
        trekk penger fra konto  
        skriv ut: ny saldo er kr...  
}
```

# Flytdiagram

- vise flyten i programmet
- planleggingsfasen
- eksempel: Quiz
- kan bruke PowerPoint -

<https://support.office.com/nb-no/article/video-opprette-et-flytskjema-af4e3f4c-3854-486a-88ff-eb35692663dc>

- finnes også andre spesiallagde verktøy
- vise program

# Brukerveiledning

- før: manualer med skjermdump
- nå: mer selvforklarende løsninger med hjelp underveis
- screen casts: videoopptak av skjermbildet



# Universell utforming

- mest mulig tilgjengelig for alle mennesker
- ikke diskriminerer som følge av funksjonsnedsettelse
- nye krav ble innført 1. juli 2014 - gjelder både for offentlige og private virksomheter
- WCAG 2.0 - <https://uu.difi.no/krav-og-regelverk/wcag-20-standarden>
  - **Web Content Accessibility Guidelines**