

Diskusjonsspørsmål

IDE (Visual Studio Code): What are some of the key features of VS Code that users find most helpful for coding? Research how VS Code is compared to other text editors or IDEs?

- Noen key features er «search»; det gir globalt søk og erstatt på tvers av den åpne mappen.
- «source control» inkluderer Git-kildekontroll som standard. «Run» - VS Code's Run and Debug View viser variabler, «call stacks», og «breakpoints». «Exstensions» – Installer og administrer utvidelsene i VS Code.
- Command Line
- Command Palette
- Git Integration
- Change language mode
- Customization
- Zen Mode
- Split view
- Status Bar
- Debugging
- Default Keyboard shortcuts

VS Code støtter et bredt spekter av språk, noe som gjør den egnet for flerspråklige prosjekter. Mens f.eks. programmet PyCharm fokuserer først og fremst på Python-utvikling og støtter dermed færre språk. PyCharm, med sin Python-sentriske design, er skreddersydd for Python-utviklere, mens VSCode er mer som en lommekniv – lett og tilpasningsdyktig med sitt enorme plugin-økosystem.

2. GitHub: In your own words, explain the concept of version control and why it's important in software development.

Versjonskontroll/kildekontroll handler om å spore og lage endringer i programvarekode. GitHub er et program der man kan administrere og holde styr på kildekodehistorikken sin. Git er designet for å hjelpe til å bedre administrere åpne kildekode-prosjekter. Det er viktig i software fordi man kan holde styr på endringer i kode, filer og andre digitale eiendeler.

How can GitHub facilitate collaboration and teamwork?

Man kan dele og samarbeide på koder, men også lagre endringer lokalt, slik at man også har en egen versjon. Programmet har også skylagring og egne profiler, slik at man kan lagre, dele og se andres koder.

3. Folder Structure: Why is a well-organised folder structure crucial for managing programming projects? What potential issues could arise from a poorly structured project?

Det er viktig med god mappe struktur for at man enkelt kan finne fram og lagre prosjektene/filene sine på riktig sted. Det sparer deg mye tid å ha et bra system i filmappene. Problemer som kan oppstå er at man ikke finner fram til prosjektene sine, og kan evt. Miste noen av filene hvis de ikke blir lagret korrekt. En dårlig strukturert oppgave kan skape problemer for kodingen, ved at oppgaven blir

vanskelig å vedlikeholde, utvide eller endre det. Det kan bli utfordrende å tilpasse seg endringsbehov og innlemme nye funksjoner.

4. Naming Conventions: Why are consistent naming conventions for files and folders important?

What are some common best practices for naming files and folders in a programming context?

Det er viktig for å ha god oversikt, det er lurt å ha faste over-mapper for f.eks. ulike fag, og deretter systematisere det etter prosjekter osv. I den mest spesifikke og relevante mappa for filen kan man legge inn denne. Det er også viktig med et godt system på navngivning av filene.

Gode fremgangsmåter for å navngi filer:

- Hold filnavn korte, men meningsfulle.
- Benytt deg av konsistente, relevante elementer.
- Unngå spesialtegn og mellomrom.
- Dokumenter og del filnavnekonvensjonen.
- Logisk filorganisering

5. Collaboration Challenges: What are some of the potential challenges or misunderstandings that can arise when working on code projects collaboratively? How can these challenges be addressed or mitigated?

En utfordring kan være at konflikter oppstår når to eller flere utviklere endrer den samme delen av koden samtidig. Å løse dette kan være tidkrevende og komplisert, spesielt hvis endringene ikke er kompatible med hverandre. Da kreves en god forståelse av koden og endringene som gjøres av hver utvikler. Hvis det ikke håndteres riktig, kan det føre til feil og andre problemer i programvaren.

Et annet problem kan være versjonskontroll. Det jobbes ofte med flere versjoner av koden samtidig gjennom et samarbeidsprosjekt. Det kan være vanskelig å holde styr på disse versjonene og sikre at den riktige blir brukt. Det gjelder spesielt når prosjektet er stort og involverer mange utviklere.

I tillegg kan andre utfordring være feilkommunikasjon og dårlig informasjonsflyt, mangel på samhold rundt mål, dårlig engasjement blant teammedlemmer, mangel på godt samarbeid, ubalansert arbeidsfordeling og samarbeidende treghet.

Problemet omhandlende versjonskontroll kan løses ved hjelp av egne versjonskontrollsystemer. Ellers så hjelper det å ha en tydelig kommunikasjon mellom alle parter. Man bør være flink og åpen for å gi og motta konstruktiv kritikk på arbeid og arbeidsvaner.

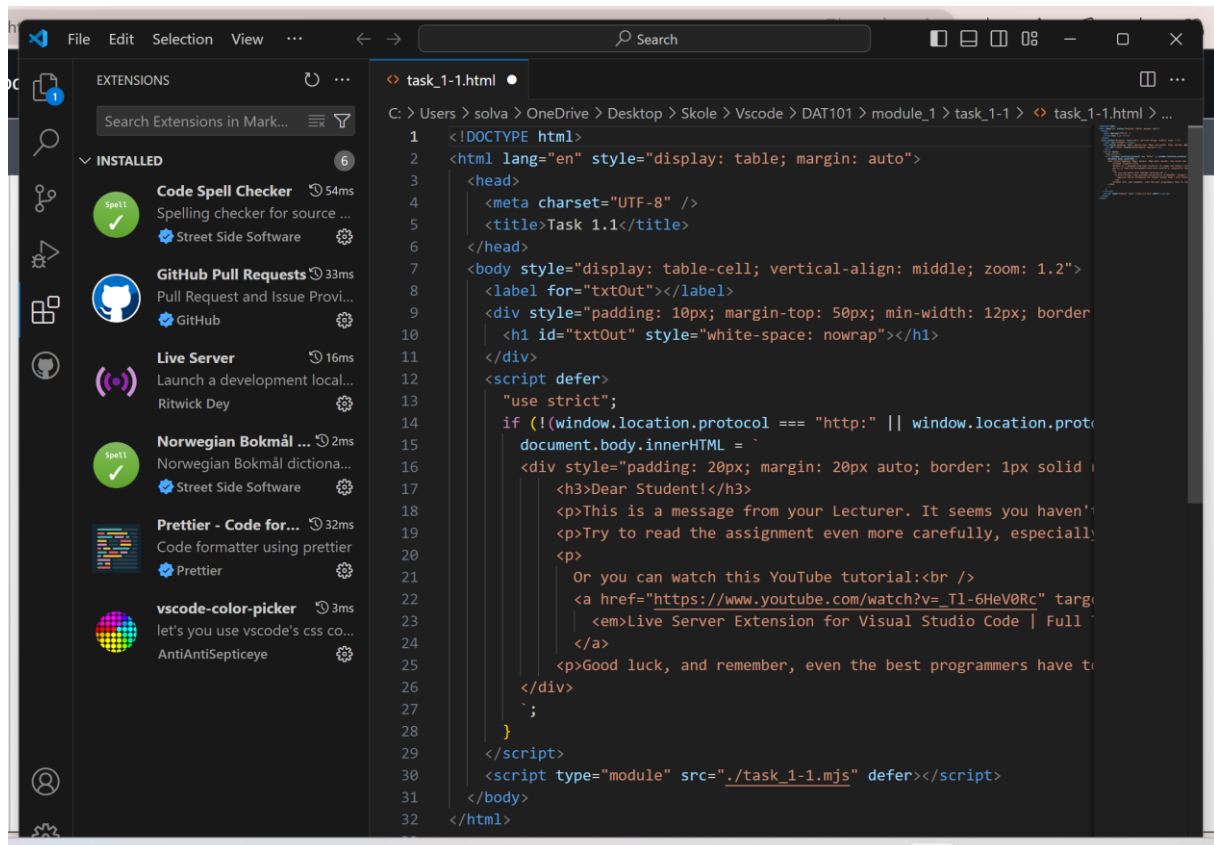
6. Best Practices: Based on your group discussion, create a list of 3-5 best practices for using VS Code, GitHub, and organising code projects.'

1. holde et systematisk og oversiktlig system
2. bruk versjonskontroll
3. orientere seg med programmene så man er klar over alle funksjoner man kan ta i bruk

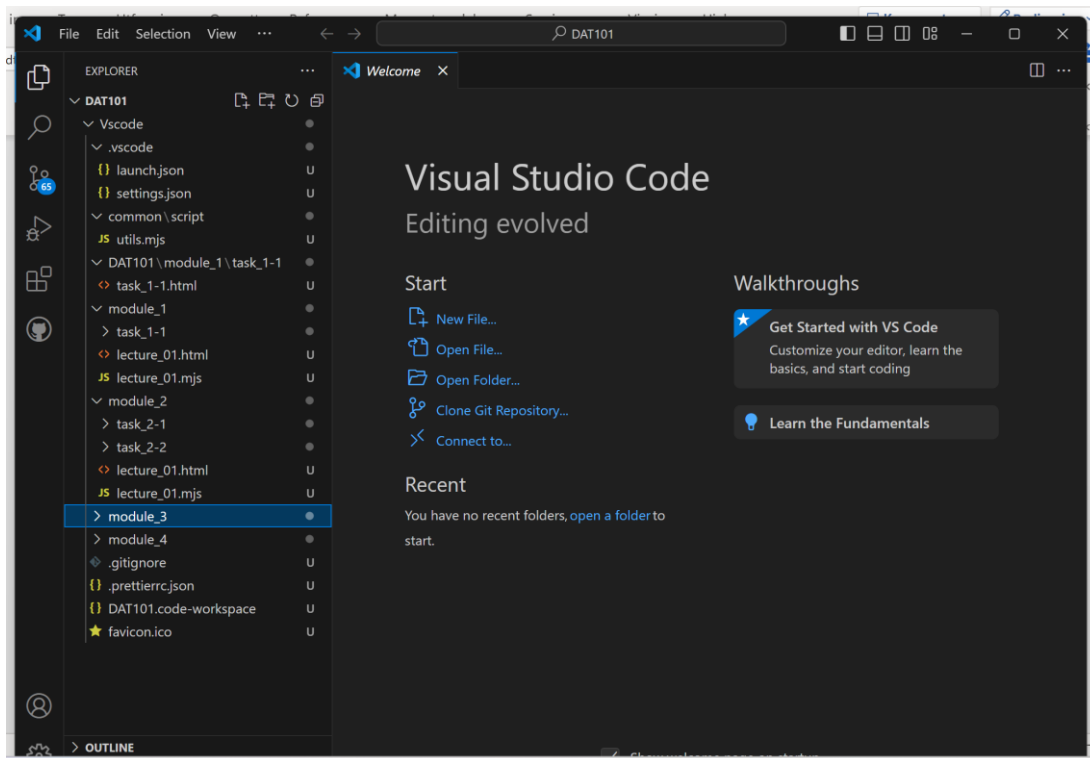
7. Real-World Application: Can you think of any real-world scenarios where the skills you've learned in Module 1 (using VS Code, GitHub, folder structure, naming conventions) would be directly applicable?

På jobb og i skolearbeid er det veldig hjelpsomt med en god fil- og mappe struktur. I fremtidige gruppeoppgaver så hjelper det at alle på gruppa er klar over og har lagt seg gode rutiner, og er enige om et bestemt system, slik at alle kan finne fram til riktig prosjekt der man kan kode sammen.

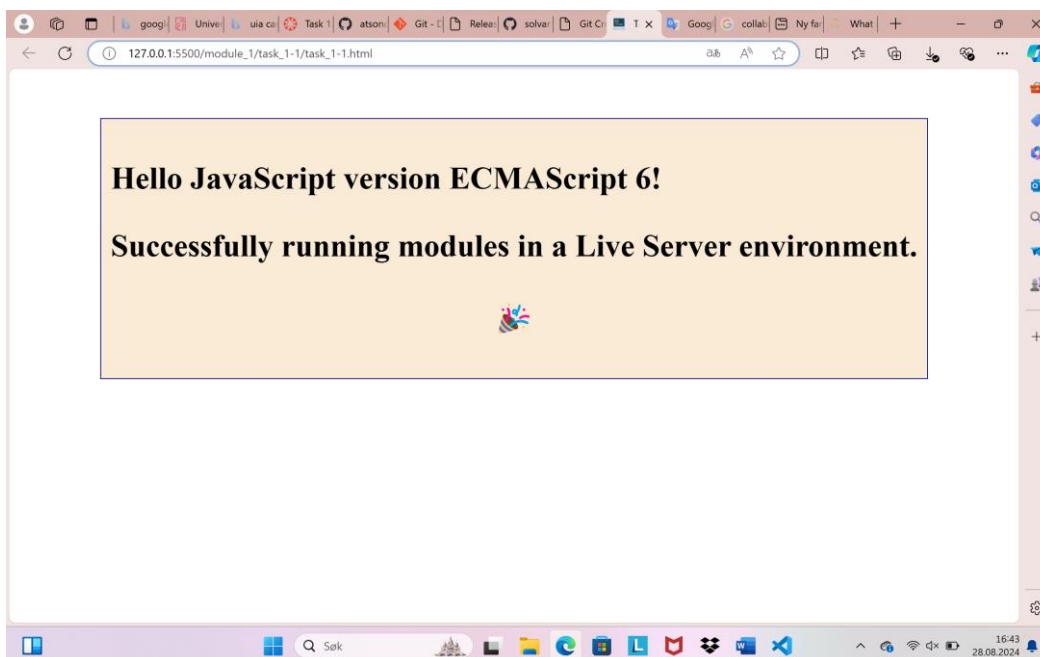
Screenshot 1



Screenshot 2



Screenshot 3



Commit URL:

[initial commit · solvar16/DAT101@7ec308b \(github.com\)](https://github.com/solvar16/DAT101@7ec308b)