



**UNDERVISNINGS
MINISTERIET**
STYRELSEN FOR
UNDERVISNING OG KVALITET

Læseplan for valgfaget teknologiforståelse

(forsøg)

Indhold

Indledning	3
Trinforløb for 7.-9. klassetrin	4
Design	4
Programmering	5

Indledning

Valgfaget teknologiforståelse er etårigt og kan vælges i 7./8./9. klasse.

Hvis eleven vælger valgfaget flere gange, skal undervisningen tilrettelægges således, at eleven udfordres i et passende omfang. Såfremt valgfaget læses i 9. klasse, skal faget indgå i den obligatoriske projektopgave.

I valgfaget teknologiforståelse skal eleverne lære at producere og analysere digitale produkter.

Elevernes arbejde med at remixe, videreudvikle og nyudvikle digitale produkter er i centrum. I en løbende vekselvirkning mellem analytisk arbejde, design og programmering skal eleverne opnå forståelse af centrale design- og programmeringsfaglige begreber og kunne vurdere samfundsmæssige implikationer af egne og andres digitale produkter.

Valgfaget har fokus på at fremme elevernes evne til at designe og realisere digitale produkter, herunder at teste og raffinere disse, og undervisningen har fokus på eleverne som kreative og innovative producenter. Eleverne skal som en del af undervisningen kunne diskutere den samfundsmæssige betydning af digitale produkter.

Fælles Mål omfatter to kompetenceområder: Design og Programmering.

Indenfor begge kompetenceområder skal eleverne arbejde med digitale produkter, hvori programmer indgår som en central komponent (fx apps, spil, simuleringer, wearables eller robotter). I arbejdet med digitale produkter skal eleverne selv udarbejde programmer eller dele heraf. Det betyder, at det ikke er tilstrækkeligt, at eleverne bruger en robot, de skal programmere (dele af) robotens adfærd.

Undervisningen tilrettelægges med udgangspunkt i kompetenceområderne. Læseplanen beskriver undervisningens progression i fagets trinforløb og danner grundlag for en helhedsorienteret undervisning. Det tilstræbes, at undervisningen gennem valg af genstandsfelter knytter sammenhænge mellem valgfaget og matematik, natur-, kultur- og/eller praktiske/musiske fag.

Det er væsentligt, at der i det enkelte undervisningsforløb arbejdes med flere færdigheds- og vidensmål på tværs af kompetenceområderne. Det skal endvidere tilstræbes, at undervisningen tilrettelægges, så den vekselvirker mellem den enkeltfaglige fordybelse og det tværfaglige arbejde.

Undervisningen skal tilrettelægges, så den imødekommer målsætningerne om en længere og mere varieret skoledag, jf. lov nr. 1640, heriblandt varieret og anvendelsesorienteret undervis-

ning, bevægelse, åben skole og understøttende undervisning.

Trinforløb for 7. - 9. klassetrin

Elevernes oplevelser og erfaringer med digitale produkter fra andre fag samt fra fritiden medtænkes i forbindelse med planlægning af undervisningen.

Undervisningen sigter mod, at eleverne kan anvende grundlæggende elementer af programmering i forbindelse med remix, videreudvikling samt egen udvikling af digitale produkter.

Undervisningen skal lede frem mod realisering af digitale produkter af forskellig art. Dette kan fx være spil, simuleringer, apps, (dynamiske) hjemmesider og robotter. Eleverne skal gennem arbejdet udvikle kendskab til de udvalgte genstandsfelters særlige virkemidler og anvendelsesmuligheder.

Det er centralt, at eleverne kan arbejde målrettet med design- og realiseringsprocesser i forhold til intenderet brug og funktionalitet. Derudover er det væsentligt, at eleverne kan analysere digitale teknologier i en samfundsmæssig, sikkerhedsmæssig og etisk kontekst.

Design

Kompetenceområdet design omfatter tre færdigheds- og vidensområder:

Ide og specifikation fokuserer på modeller til design af digitale produkter.

Realisering fokuserer på at realisere en idé til et digitalt produkt samt evaluere og forbedre produktet i en iterativ proces.

Samfundsmæssig betydning fokuserer på digitale produkters betydning for og påvirkning af menneskelig aktivitet.

Ide og specifikation

Undervisningen skal fokusere på elevernes viden om arbejdsgange til idegenerering og problemnedbrydning for digitale produkter. Eleverne skal arbejde med at idéudvikle og dekomponere fra en problemstilling. Elevernes aktiviteter skal understøttes ved brug af enkle proces- og datamodeller, fx skitser, flowdiagrammer, storyboard og begrebskort. Eleverne skal bygge videre på den viden og de færdigheder i relation til designprocesser, som de allerede har opnået i forbindelse med undervisning i obligatoriske fag, som håndværk og design, dansk m.fl.

Realisering

Undervisningen skal fokusere på elevernes evne til at gen-

nemføre en trinvis proces til realisering af digitale produkter med afsæt i en specifikation, udtrykt fx i idéskitser, proces- og datamodeller. Eleverne skal i flere omgange arbejde med evaluering og forbedring af deres digitale produkter med udgangspunkt i den intenderede brug, blandt andet ved brug af respons.

Samfundsmæssig betydning

Eleverne skal introduceres til digitale produkter, som udfordrer og radikalt forandrer vilkår i såvel fritids- som arbejdsliv. Undervisningen skal give eleverne indsigt i, at den digitale teknologi giver et kvalitativt løft til kognitiv formåen ved automatisering af og støtte til mentale processer. Eleverne skal kunne vurdere digi-

tale produkters samfundsmæssige potentiale og betydning, herunder brugsmæssige, sikkerhedsmæssige og etiske aspekter.



Programmering

Kompetenceområdet programmering omfatter seks færdigheds- og vidensområder:

Mønstre fokuserer på, hvordan standardløsninger kan anvendes i programmering.

Algoritmer fokuserer på sammenhæng mellem algoritmer og programmer.

Data og programstruktur fokuserer på at anvende lister, procedurer og funktioner i forbindelse med programmering.

Kodning fokuserer på modifikation af andres og konstruktion af egne enkle programmer.

Programmeringssprog fokuserer på konstruktioner i programmeringssprog med henblik på at kunne forstå og udvikle programmer.

Test og fejlretning fokuserer på systematik ved test og fejlretning af programmer.

Mønstre

Undervisningen skal lede frem mod, at eleverne gennem iagttagelse af mønstre kan håndtere enkle programmer på semantisk niveau. Eleverne introduceres til elementære mønstre i programmer med det formål at lette læsning af programmer og skabe et fundament for, at de senere kan arbejde mere abstrakt med programmering. Eleverne skal kunne genkende enkle mønstre og anvende disse i kodning.

Algoritmer

Eleverne skal opnå indsigt i sammenhængen mellem algoritmer og programmer. Eleverne skal kunne læse og skrive algoritmer som udtrykkes grafisk eller som tekst. Det kan fx være gennem brug af pseudokode, symboler og flowdiagrammer. Eleverne arbejder med, hvordan forskellige algoritmer kan løse samme problem. Tilsvarende arbejder eleverne med, hvordan en algoritme med få justeringer kan løse forskellige problemer.

Data- og programstruktur

Eleverne skal gennem arbejdet med lister og mængder blive fortrolige med dataabstraktion. Undervisningen skal desuden omfatte arbejde med funktioner og procedurer som et middel til organisering og forenkling af programmers struktur samt implementering af algoritmer.

Kodning

Eleverne skal arbejde med kodning i enkle, selvstændige øvelser og i forbindelse med egne

digitale produkter. Eleverne skal herunder anvende kodning til at tilføje ny funktionalitet til allerede eksisterende programmer gennem ændringer i enkelte kodelinjer og konstruktion af udvidelser, som implementeres i programmet. Arbejdet med modifikationer skal lede frem mod, at eleverne kan konstruere egne produkter. I valgfaget benyttes kodning som synonym for programmering, men begrænset til enkle problemstillinger med en beskeden grad af problemløsning og algoritmisk design.

Programmeringssprog

Eleverne skal arbejde med en række konstruktioner knyttet til programmering som variabler og

tilhørende datatyper (fx tekst, billede, symbol og tal) og kontrolstrukturerne sekvens, forgrening og løkker. Eleverne skal lære at anvende de forskellige konstruktioner i programudviklingen. Undervisningen tager udgangspunkt i blokprogrammering og kan udvides med tekstprogrammering.

Test og fejlretning

Eleverne skal arbejde systematisk med at finde og rette fejl i simple programmer. Eleverne skal kunne identificere og rette fejl i egne og andres programmer samt arbejde med løbende evaluering og rafinering. Undervisningen skal lede frem mod, at eleverne kender metoder til at gøre programmer enkle og læsbare.