

IT2, 2-timersprøve: “Påskerenn”

13. februar 2020. 08.15-10.25

The english version can be found at the end of the document.

Alle hjelpemiddel er tillatt, med unntak av kommunikasjon (inkluderer å samarbeide). Du bør begrense deg til boka og W3Schools.com for å øve til eksamen.

Plagiering: Dersom du hentar innhald som kode og grafikk frå andre kjelder enn boka så skal dette visast til i vedlagt dokument (eller som tydeleg kommentar i koden). Spør meg dersom du er usikker.

Generell informasjon og vurdering

Du skal løyse oppgåvene slik dei er presentert. Dersom du av ulike grunnar vel andre løysningar må du passe på å få vist same kompetanse på annan måte. Du skal utnytte dei teknikkane me har lært, til dømes strategisk bruk av ulike variablar og kolleksjonar (til dømes tekst, tal og arrays), if-setningar, løkker, funksjonar, skjema osv. Der du gjer tilpassingar må du begrunne og forklare dette i kommentarane dine. Har du behov for større mengder tekst skriv du i eit eige tekstdokument.

Krav til kode: Strukturert og ryddig satt opp, med ein gjennomgåande måte å gjere det på. Du skal kommentere koden. Variabelnavn og andre element skal ha sjølvforklarande navngjeving og fylgje ein standard.

Det er eit mål at du skal kunne separere HTML, Javascript og eventuell CSS i eigne filer.

Du skal ha eit fokus på å begrense moglegheitene for at brukaren kan gjere feil, og dermed at programmet ditt feilar. Tydelege feilmeldingar og tilbakemeldingar er viktige punkt.

Løysinga skal i størst mogleg grad vere dynamisk.

Krav til presentasjon: Oversikteleg design, med allerede nemnte “tydelege beskjedar og tilbakemeldingar til brukaren”.

Krav til innlevering: Du skal jobbe i ei mappe med god struktur. Når du skal levere inn komprimerer du denne mappa, med alt innhaldet, og leverer inn zip-fila via Classroom. Dette er òg ein del av vurderinga du får.

Kompetansemål:

- lese og bruke dokumentasjon og kode
- definere variabler og velge hensiktsmessige datatyper
- tilordne uttrykk til variabler
- programmere med enkle og indekserte variabler eller andre kolleksjoner av variabler

- programmere med valg og gjentakelser
- lage egne og bruke egne og andres funksjoner eller metoder med parametere
- programmere funksjoner eller metoder som blir aktivisert av hendelser
- Utvikle og sette sammen delprogrammer
- teste og finne feil i programmer ved å bruke vanlige teknikker
- planlegge og utvikle multimedieapplikasjoner ved å kombinere egne og andres multimedieelementer av typene tekst, bilde, lyd, video og animasjoner
- bruke programmeringsspråk i multimedieapplikasjoner
- vurdere og bruke relevante filformater for tekst, bilde, lyd, video og animasjoner
- vurdere multimedieprodukter med hensyn til brukergrensesnitt og funksjonalitet

Oppgåvetekst

Du skal lage eit program som skal brukast under det årlege påskerennet på stølsområdet Olastøl i Hardanger. Her samlast gode hyttenaboar for å gjennomføre konkurransar i 3 stk. ulike aktivitetar. Prestisje, knekte lårhalsar og forholdsvis godt humør er viktige stikkord.

Lag ein liten introduksjon til påskerennet øverst i dokumentet, gjerne med noko enkel påskegrafikk. Du blir ikkje vurdert på kor bra denne delen ser ut.

Krav 1: Deltakerlista varierer frå år til år, men du skal **forhåndsregistrere** 5 stk. deltakarar (hardkoda inn ved oppstart). Informasjonen som MÅ vere med om desse er:

- Navn
- Alder
- Resultat frå aktivitet 1 (endå ikkje nokre data her)
- Resultat frå aktivitet 2 (endå ikkje nokre data her)
- Resultat frå aktivitet 3 (endå ikkje nokre data her)
- Total poengsum, resultatet frå dei tre aktivitetane

Utover dette kan du legge til meir informasjon etter behov. Kva sjølve aktivitetane består av treng du ikkje ta stilling til, om du ikkje vil.

Krav 2: Deltakerlista skal skrivast ut til brukaren, ved oppstart, slik at ein ser kven som er med. Typisk ser ein for seg at navn og alder må vere med i denne oversikten. Altså, deler av data frå “krav 1” skal visast ved oppstart.

Krav 3: Det skal vere mogleg å registrere nye deltakarar, utover dei forhåndsbestemte. Igjen er navn og alder det som er nødvendig på dette punktet. Etter at ein ny deltaker er registrert skal deltakerlista (frå krav 2) oppdaterast.

Krav 4: Det skal vere mogleg å registrere resultat frå konkurransane. Eksempelvis, deltakar nr. 1, “Jo Bjørnar”, har fått 34 poeng under aktivitet nr. 1, 22 p. for akt. 2 og 31 p. for akt. 3. Basert på dette skal data i kolleksjonen oppdaterast, og total poengsum reknast ut.

Du set sjølv begrensingar på maks poengsum (til dømes maksimum 100 poeng per aktivitet) og andre eventuelle reglar/krav.

Krav 5: Det skal vere mogleg å vise ein oppsummering av resultata frå konkurransen. Typisk vil ein ved å trykke på ein knapp få ei utskrift som viser resultatlista sortert på den totale poengsummen. Viktig: Det skal berre skrivast ut resultat for deltakarane som har fått ein total poengsum.

Når det gjeld dei tre første plassane skal det visast fram pokalar (eller annan unik grafikk, type bilete/ikon eller liknande) foran desse. Resten av plasseringane bør det stå nummer foran.

English version

Information on grading (assessing)

You are to show your skills in variables, if-else, loops (for and/or while), functions and collections. Use them to solve the tasks you are given in the following parts.

The chances of the user doing something wrong should be minimal. Give good information on the part of the input, and try to handle errors in the input.

Your program should provide the user a clean interface, with clear instructions and feedback.

You should use good naming (f.ex. functions, variables), indentations etc. "Good order".

Write comments to help explain your code.

When you are finished with our programs you should add all the necessary files to a compressed file (zip, rar, 7z or similar) and upload the file to Google Classroom. Optionally, send the files to jo_hau@asvg.no if you are experiencing problems.

Do NOT get anyone else to write code for you. No cooperation with others. BUT, you can search for and get help from all available sources. W3Schools is our primary source, but any website is available. No **posting** in forums, messaging friends asking for direct help etc. All sources you are using should be given as a source.

More information in the Norwegian introduction.

Task: Competition

Every year a group of friends get together to host a "friendly" competition in the mountains. There should be three different activities. Your task is to make a program that should solve some tasks, given below.

Task 1: The participants vary from year to year, but you should pre-register 5 participants from the start of the program. You should keep the following information on these:

- Name
- Age
- Results from activity no. 1 (no registered data to start with)
- Results from activity no. 2 (no registered data to start with)
- Results from activity no. 3 (no registered data to start with)
- Total points

Add more information if needed. You don't need to know what the activities are.

Task 2: The list of participants should be output to the user (written to the HTML) at start-up so that you can see who is participating. Typically, name and age should be shown here.

Task 3: It should be possible to register more participants, in addition to the pre-registered ones. Name and age is necessary. After this is added an updated version of the list from "task 2" should be written to the user.

Task 4: It should be possible to register results from the competition. For example, participant no. 1, "Jo Bjørnar", has been given 34 points in activity no. 1, 22 p. for act. 2 and 31 p. for act. 3. Based on this the data in the collection should be updated and the total score should be set.

Task 5: It should be possible to show the results from the competition. Typically the user pushes a button and is given an output with the list including the results, sorted on the total score. Only the participants with a registered score should be shown in the results.

The first three places should have a specific graphic shown, for example a trophy. The others should typically have a number beside them.