

# Tentamen 1 - Test (HT2022)

Tentamen består av frågorna nedan, som du ser ger olika frågor olika antal poäng. Du kan fritt välja vilka frågor som du vill lösa.

För G behöver du lösa problem som räcker till 4 poäng.

För VG behöver du lösa problem som räcker till 6 poäng.

I princip får man endast poäng om man löser hela problemet. Men det är inte omöjligt att ni får delar av poängen för ett problem om ni har kommit ganska långt men inte löst det helt. Så lämna in all kod som du upplever visar att du kan saker.

Inlämningen till Ladok ska bestå av ett flertal mappar, alla zippade till en fil. Varje mapp ska innehålla koden som är lösningen till ett problem.

Det är INTE tillåtet att prata med någon annan under tentamen. Snälla, fråga inte någon annan för om man kopierar från någon annan så åker båda dit, den som kopierade och den som lät kopiera. Det är inte så roligt att hamna hos disciplinnämnden.

Notera att du kommer att behöva redovisa och förklara koden som du har lämnat in. Du måste helt behärska den.

OBS: Videon som är längre eller väger mer än det som anges under varje problem accepteras inte. Använd gärna Zoom som har bra video-optimering. Du behöver inte synas på videon (men du behöver höras).

## 1. 1 poäng

Vi har skapat en enkel sida i mappen "p1". När sidan laddas upp förväntar vi oss se en siffra mitt på den. Men så blir det inte. Spela in en kort video (max 2 minuter, max 15Mb) som förklarar:

- Varför koden inte fungerar som förväntat.
- Hur koden ska se ut för att sidan ska fungera som förväntat.

## 2. 1 poäng

Resursen <https://api.dictionaryapi.dev/api/v2/entries/en/river> är en array som inkluderar information om det engelska ordet "river". Studera arrayen och notera att den har två element (två vitt skilda meningar av ordet "river"). Varje element är ett objekt som inkluderar nyckeln "meanings", som är en array av objekt. Ett av dess nycklar är "definitions", som också är en array av objekt. Detta objekt inkluderar nyckeln "definition", som vi är intresserade av.

Vi är alltså intresserad av första definition av första meningen av första "ord-användningen". För "river" är det strängen "A large and often winding stream which drains a land mass, carrying water down from higher areas to a lower point, oftentimes ending in another body of water, such as an ocean or in an inland sea."

Problemet går ut på att koda en sida (all HTML och CSS som behövs finns på plats och får ej ändras) som, efter att tangenten Enter tryckts, läser användarens input och visar definitionen av ordet som står där. Du behöver inte ta hänsyn till vad som händer om användaren skriver ett ord som inte finns på ordlistan i dictionaryapi. Om ordet finns ska första definitionen av första meningen av första "ord-användningen" visas på sidan (se video). Om ordet inte finns så behöver det inte hända något.

I lösningen får du inte använda metoden .then()

## 3. 1 poäng

Resursen <https://teaching.maumt.se/apis/musiker/> innehåller information om en musiker. Den tar emot följande GET-parametrar:

musician: Namnet på musikern du söker information om. Det finns bara information om dessa fyra musiker: Nina\_Simone, Madonna, Jimi\_Hendrix och Stevie\_Wonder. Notera underscore i namnen.

instruments: Om parametrern finns med (den behöver inte anges något värde) så kommer det att i resursen finnas en array av några instrument som musikern spelar.

songs: Om parametrern finns med (den behöver inte anges något värde) så kommer det att i resursen finnas en array av ett fåtal låtar som musikern har publicerat.

Parametern "musician" måste vara med. En (endast en) av de andra två parametrar, antingen instruments eller songs, måste vara med (men behöver som sagt inte ha något värde). Studera vilken data-struktur resursen har beroende på vilka GET-parametrar som anges.

I mappen finns det HTML- och CSS-filer redo att använda. De får ej ändras. Se videon för att förstå vad problemet går ut på. Vi bryr oss inte om vad som händer om namnet som skrivs på input-fältet (musikern) inte finns med i databasen.

I lösningen får du inte använda `async` eller `await`.

## 4. 1 poäng

Detta problem är en fortsättning på antingen P2 eller P3. Du får välja. Du kan endast välja ett.

Problemet går ut på att utveckla koden så att den tar hand om fallet när ordet eller musiker-namnet inte finns i databasen. Studera vilket svar du får från servern och bygg en sida (baserad på den i P2 eller P4) som ger användaren rätt feedback. Se videon.

Om du bygger på P2 får inte `async` eller `await` användas.

Om du bygger på P3 får inte `.then()` användas.

## 5. 2 poäng

Resursen [https://teaching.maumt.se/apis/random\\_number/](https://teaching.maumt.se/apis/random_number/) tillhandahåller en random siffra och accepterar två GET-parametrar:

min: random siffran kommer att vara lika med min eller högre (default value = 0).

max: random siffran kommer att vara lika med max eller lägre (default value = 10).

Resursen [https://teaching.maumt.se/apis/random\\_color/](https://teaching.maumt.se/apis/random_color/) tillhandahåller ett random färg-namn och accepterar en GET-parameter:

colors: ett JSON-kodad array av färgnamn från vilka API:n väljer (default = en array av alla CSS-färgnamn).

Använd resurserna ovan för att skapa en sida som den som syns på videon. Storleken (bredden och höjden, likadana) på de tre fyrkanterna (samma höjd, samma bredd) ska vara mellan 10vw och 20vw (inklusive). Färgen ska vara en av coral, skyblue, gold eller hotpink.

Notera att det redan finns HTML- och CSS-kod som kan hjälpa dig på traven. Det finns dock inget krav på att använda den men det rekommenderas varmt.

Notera också att allting ska starta med funktionsanropet `go(X)`; där X är antalet divar. Om X är större än fyra så kommer sidan inte att se så bra ut men det är inget du behöver tänka på.

## 6. 3 poäng

Koda en sida som skapar AxA divs, alltså en grid av A kolumner och A rader till 100% (100vh och 100vw) fylld med divar. Som ni ser i CSS-filen har alla dessa divar transition-property "background-color", bakgrundsfärgen "transparent" och en initial transition-duration av 1s.

En efter en (serie) ska var och en av dessa div ska få en random bakgrundsfärg (se resursen `random_color` nedan) och en random transition-duration (se resursen `random_number` nedan). När den nya bakgrunds-färgen sätts så startar CSS-transitionen.

Efter det att CSS-transitionen är avslutad ska en ny färg och en ny transition-duration hämtas (via resurserna) FÖRUTOM om diven hade bakgrundsfärgen "skyblue". Då ska inget mer hämtas och diven ska inte ändra bakgrundsfärg mer. Till slut kommer alla divar att ha den bakgrundsfärgen. Se videon.

Resursen [https://teaching.maumt.se/apis/random\\_number/](https://teaching.maumt.se/apis/random_number/) tillhandahåller en random siffra och accepterar tre GET-parametrar:

min: random siffran kommer att vara lika med min eller högre (default value = 0).

max: random siffran kommer att vara lika med max eller lägre (default value = 10).

bullet: svaret ges med minsta möjliga väntan. Parametern behöver inget värde.

Resursen [https://teaching.maumt.se/apis/random\\_color/](https://teaching.maumt.se/apis/random_color/) tillhandahåller ett random färg-namn och accepterar en GET-parameter:

colors: ett JSON-kodad array av färgnamn från vilka API:n väljer (default = en array av alla CSS-färgnamn).

bullet: svaret ges med minsta möjliga väntan. Parametern behöver inget värde.

Du måste använda bullet-parametern för båda resurserna så att svaren kommer så snabbt som möjligt. Färgen för divarnas bakgrunde måste vara en av dessa: skyblue, lightblue, cadetblue eller seagreen.

Koden får inte ha några globala variabler och måste starta med funktionsanropet `go(A)`; där A är antalet kolumner och rader i griden.