

# Projektuppgift

Täckta kursplansmål:

Efter genomförd kurs ska den studerande ha färdigheter i att:

6. Utveckla webbsidor som hanterar dynamiskt innehåll och användarhändelser.
7. Tillämpa teknik för att hämta och skicka data till servern
8. Använda teknik för att hantera data i webbläsaren

Efter genomförd kurs ska den studerande ha kompetens för att:

9. Producera interaktiva webbapplikationer med JavaScript med användande av versionshantering och på ett säkert sätt och enligt god programmeringssed.

## Övergripande

Detta projektet blir som ett litet hackathon där ni i par ska komma på, snabbt skissa upp och sen implementera en MVP(minimum viable product). Sidan ska hostas på [GitHub Pages](#) och I slutet så lägger läraren in alla pages länkar i ett galleri som vi sen kollar igenom efter deadline.

Det finns ingen begränsning på vad appen ska göra eller lösa för problem, men det måste finnas ett motiv och den måste använda sig av en eller flera web api:er.

Varje vecka kommer att ha en liten mini-deadline varje torsdag kväll. Bara skriv till läraren och länka med ert GitHub repository. Om ni möter den deadlinen så kan ni få feedback från läraren på fredagen dagen efter.

De tre veckorna kommer att delas in i tre faser:

v1: Första utkast i HTML/CSS. Idén med sidan ska finnas i en README och sidan ska vara hostad på GitHub pages.

v2: JS kod finns på plats för att kalla på web api:er och uppdatera DOM, det finns säkert en hel del buggar och mycket feedback att ge.

v3: Det är dags att finpolera, städa upp i CSS, se till att sidan inte kraschar.

**Avgränsning:** jQuery är tillåtet. Även andra frameworks med såsom TypeScript

## G krav

- Alla medlemmar i gruppen ska bidra och committa ungefär lika. **Om en person bara har en två tre commits efter inlämning kommer den personen att behöva komplettera på egen hand senare.**
- Ta in input från och reagera på användaren
- Utnyttja en eller flera web api:er
- Manipulera DOM för att presentera resultatet
- Skriv stilren kod (ES6, "use strict"; auto-indentering, enhetliga namn, endast let&const)
- Enskild rapport på vad du gjort i projektet

## VG krav

- Utöka din rapport med ett stycke där du reflekterar över, och analyserar de lösningar som valts i gruppprojektet gällande händelsehantering, dynamiskt innehåll samt kommunikation med server

## Deadline

**Den 4/3, Tors kl 23:55**

## Vecka 1

### Bestäm Web API

Leta reda på en web-api som inspirerar till en app idé. Det finns en bra samlingssida [här](#) men det är helt ok att bara söka runt på nätet med. Speciellt om ni har en idé redan och bara behöver hitta rätt teknologi för att möjliggöra den.

#### Exempel:

Med hjälp av [pokeapi.co](#) tänker jag bygga en sida som är en [pokédex](#) precis som i animen. Man kan titta på alla Pokémon och välja de man har samlat på sig. Alla man väljer, sparas ner i en lista som finns kvar efter man har stängt ned sidan med hjälp av [localStorage](#). Om det finns tid över i slutet av projektet vill jag göra så att man kan se sina Pokémon springa runt i bakgrunden omkring pokédex:en. (Alla Pokémon sprites i B&W uppsättningen finns som animerande gif:ar)

### Skapa utkast på GitHub Pages

Så fort ni har idén, börja med en snabb skiss eller en wireframe. Det borde max ta en timme. Starta sen projektet och börja pusha upp HTML/CSS kod för att bygga grundstrukturen. Undvik att få CSS:n att fungera perfekt, hemsidan kan vara lite stökig i början. Se redan nu till att båda får in några commits var och att GitHub Pages sidan ser lika dan ut som när ni debuggar lokalt. När allt funkar så skriv ner eran kod i en README och länka till eran GitHub Pages så att man enkelt kan nå siten från GitHub. Lägg gärna en länk till githubben någonstans inne på själva sidan med, till exempel i footern.

## Vecka 2

### JavaScript filstruktur

Det kan lätt bli mycket kod att hålla koll på så bryt ut JS koden i flera filer så långt som det är rimligt. Till exempel så kan all eventhantering för en .html fil ligga i en egen .js fil. Så om vi har två sidor så har var sida sin .js fil. Om det finns något vi gör på båda sidorna, kanske en speciell animation eller likande så kan vi flytta ut bara den biten till en util.js fil som båda sidor använder.

På samma sätt så kan varje web-api ha sin egna fil med, som ansvarar för att kalla på och plocka ut den viktiga informationen från just det stället. Oftast är det bästa sättet att packa ner allt till en eller ett par funktioner som är lätta att använda i resten av programmet.

Det finns en [ES6 funktionalitet som hjälper till med detta kallad modules](#).

### Kalla på Web API

Ta en web-api i taget och koda er defensivt fram. Börja med en blank testhemsida som bara har en enda knapp och bara länkar till web-api JS kodfilen. Då finns det inget annat som kan distrahera när ni sen debuggar koden. Först gör ni ett anrop som har så få variabler som möjligt så att ni bara kan kolla att ni förstått dokumentationen ordentligt. Stepthrough debugging, alert, prompt och console.log är alla jätteanvändbara verktyg.

När ni väl lyckat få ett svar är nästa steg att tänka på hur resten av koden ska använda denna api:n.

Exempel:

I pokédex sidan så hade det varit bra om det fanns en funktion `getPokemon()` som tar in en pokemons id som in-parameter och skickar tillbaka ett objekt som har lite info om pokemonen, och gärna lite länkar till bilder med. Då hade jag enkelt kunnat fylla ut en beskrivning och visat visuellt vilken pokemon man tittar på just nu. Men eftersom det tar tid att få den informationen får jag göra det till en async funktion.

```
async function getPokemon(id) {  
  let pokeObj = {};  
  // ...  
  pokeObj.name = ...;  
  pokeObj.desc = ...;  
  pokeObj.imageUrl = ...;  
  // ...  
  return pokeObj;  
}
```

## Vecka 3

### Uppstädning

Börja med att skapa en TODO lista med allt som inte fungerar och rangordna punkterna från viktigast till minst viktigast. Målet är att göra de 20% mest viktiga sakerna denna vecka och släppa resten.

Tanken med en MVP är inte att man avslutar ett projekt på ett fint sätt, utan att man bygger så pass mycket att man kan se hur en framtida större produkt hade kunnat bli. Den minsta möjliga versionen som ändå gör något och förmedlar tanken bakom.

### Dubbelkoll

Gå igenom G- och VG-kraven i detta dokumentet en gång till och kolla att ni inte har missat något. Kolla att allt som ska vara uppladdat finns uppladdat på PingPong, att den senaste koden finns upppushat till GitHub och zippat, att GitHub Pages fungerar som det ska, etc.