

Individuell uppgift - OOP

- Due Sunday by 23:59
- Points 0

Om uppgiften

Denna uppgift prövar färdigheter hur objekt-orienterad programmering kan användas i Javascript.

- I del 1 ska du med en liten uppgift visa att du förstår grundsättet för objekt-orienterad programmering i Javascript på det prototypbaserade sättet
- I del 2 ska de används det moderna sättet med klasser som kom i ES6 genom att skapa grundlogiken för ett enklare pokerspel

Vad du ska göra

Del 1 - En klass Color (prototypbaserad)

Gör en konstruktorfunktion för en färg, Color, som initieras med argumenten r, g och b (d.v.s varje värde tar in värdet 0-255).

Skapa metoder som konverterar mellan olika färger genom att lägga till var och en av dessa på Colors "prototype". Dessa är:

- rgb() - returnerar färgen "rgb(x,y,z)"
- hex() - returnerar färgens hexkod "#XXYYZZ" - googla efter lämplig omvandling, rgb till hex
- rgba() - användaren anger även alpha-värdet som argument och returnerar "rgba(x,y,z,a)"
- frivilligt med fler omvandlingar ex hsl(), etc

Testa genom att skapa en färg (Color) och anropa respektive metod, exempelvis genom att byta bakgrundsfärgen på en sida.

Del 2 - OOP Poker (ES6 klasser)

I denna del ska du använda ES6 klasser för grundlogiken i ett pokerspel. Gå till <https://github.com/chasacademy-sandra-larsson/js--oop-poker> (<https://github.com/chasacademy-sandra-larsson/js--oop-poker>) och följ instruktionerna.

Pokerspelet är endast tänkt att köras i konsolen, det är alltså frivilligt att bygga vidare med ett UI.

VG-krav

- Även steg 5 - 8 i OOP Poker

- VG-fråga:

Denna del behöver du bara göra om du satsar på VG.

I din readme-fil på github ska du ha med ett resonemang kring din kod. I denna ska du nyanserat resonera kring styrkor och brister i ditt genomförandet, alltså i den kod du utvecklat.


VG-nivån bedöms genom kvalitén på koden i kombination med din förmåga att se just styrkor och brister i den. Detta betyder att om din kod har allt för låg kvalitet räcker det inte med resonemang kring det för att rädda upp, men det betyder också att ingen kod behöver vara helt perfekt men det är bra att du själv kan peka på de brister du då ser.

Din inlämning

En länk till ditt repository som ska vara publikt, innehållandes **all kod** + en **README.me**

Betygsättning

Betygsättningen på denna uppgift utgår från kursmålen och betygskriterierna i matrisen nedan.

En matris (5)			
Criteria	Ratings		
Beskrivning av kriterium	Full poäng	Inga poäng	
 (Färdighet) Bygga frontend för webbapplikationer med Javascript utan ramverk/bibliotek (VanillaJS)	Väl godkänd - Istället utan briser eller missar, eller bara i sådan omfattning att de knappt märks eller påverkar slutresultatet. Samt dessutom nyanserat resonera kring styrkor och brister i genomförandet.	Godkänd - Kan utan allvarliga brister eller missar på egen hand bygga frontend för webbapplikationer med Javascript utan ramverk/bibliotek (VanillaJS)	Icke Godkänd