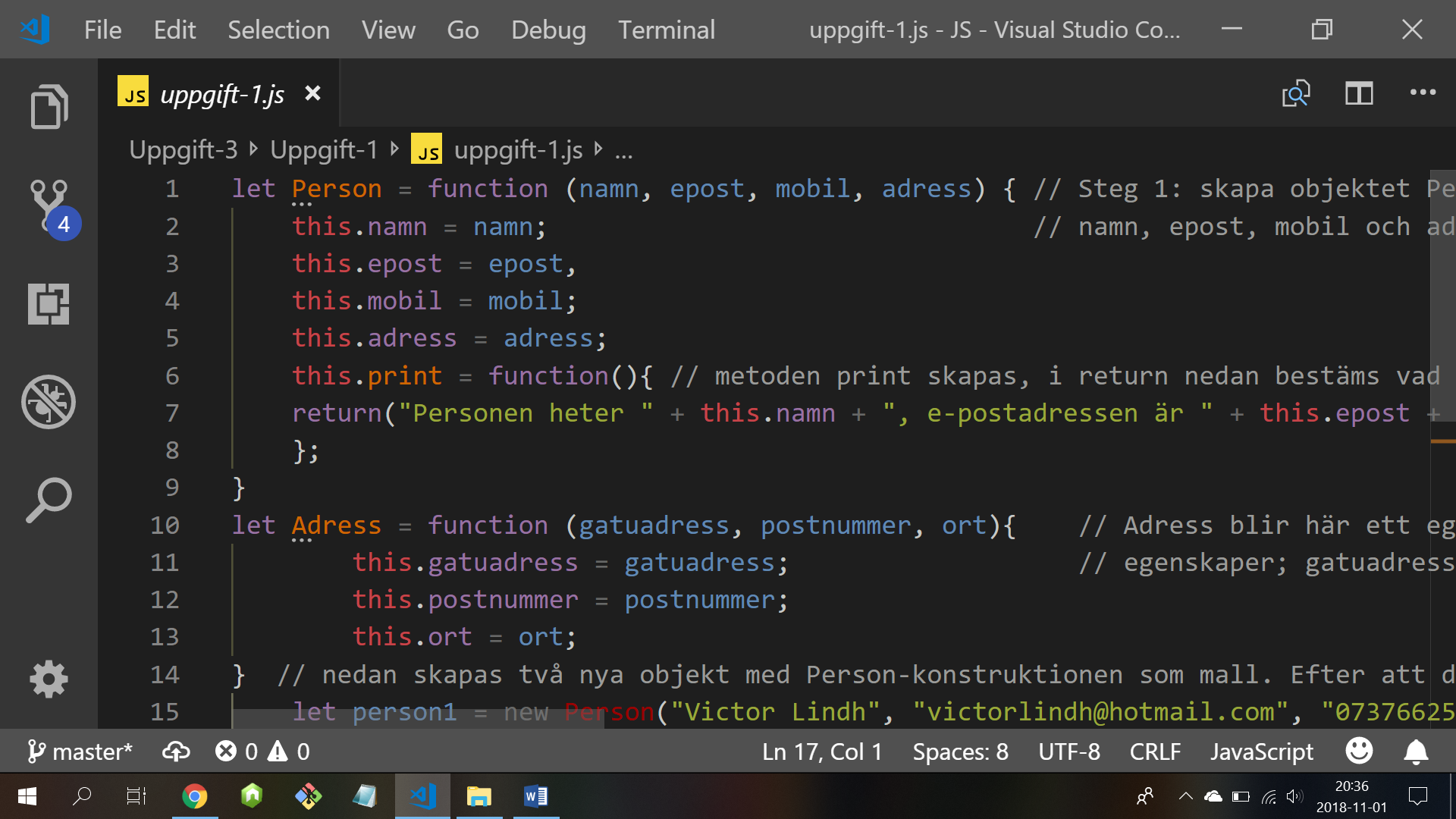
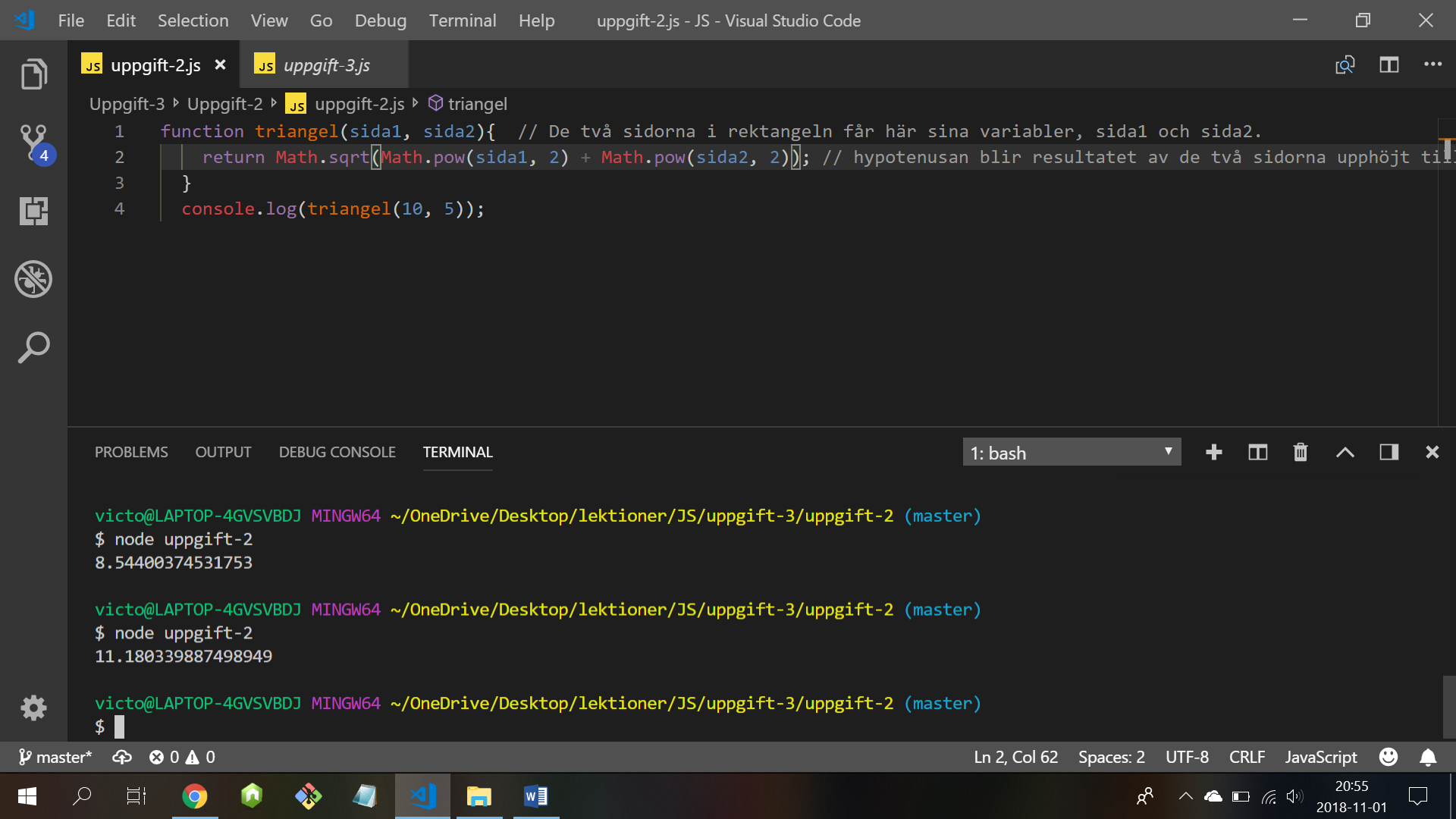
Uppgift 3 – Victor Lindh, FEND18

Deluppgift 1



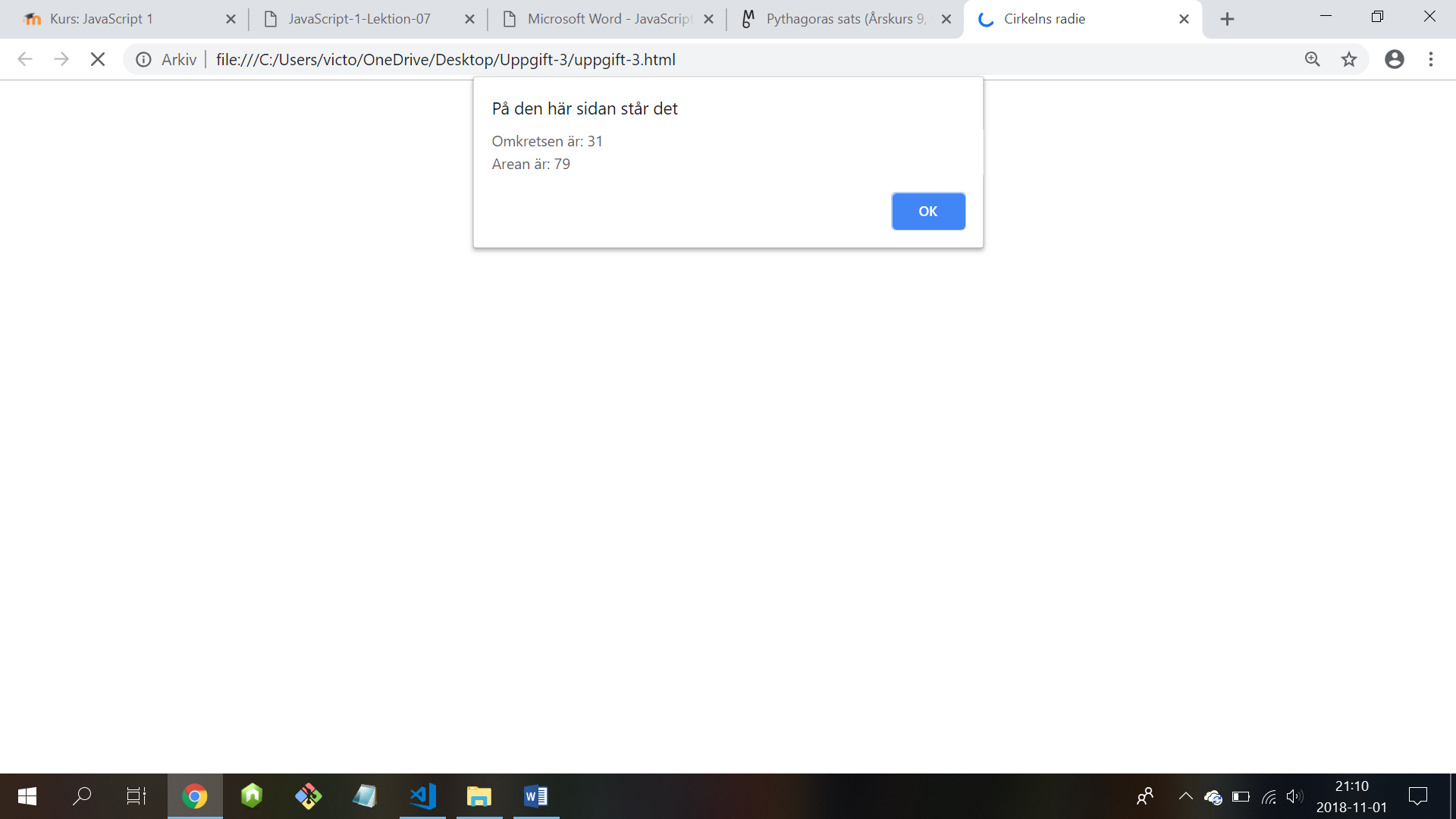
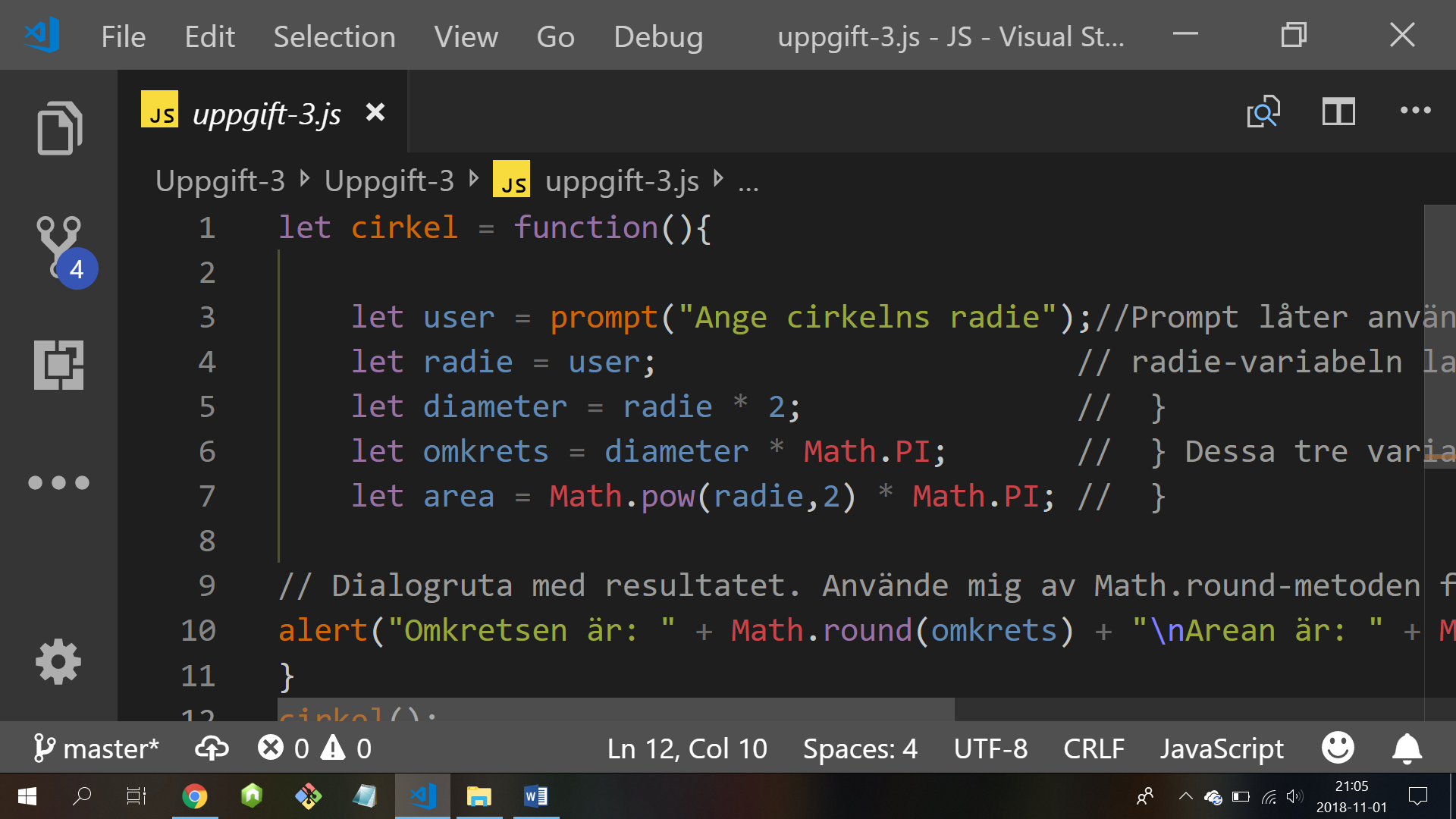
Adress blev i min lösning även det en konstruktor, precis som Person. Adress får sedan sina argument inuti argumenten för person-objekten, inbäddade med egna parenteser.

Deluppgift 2



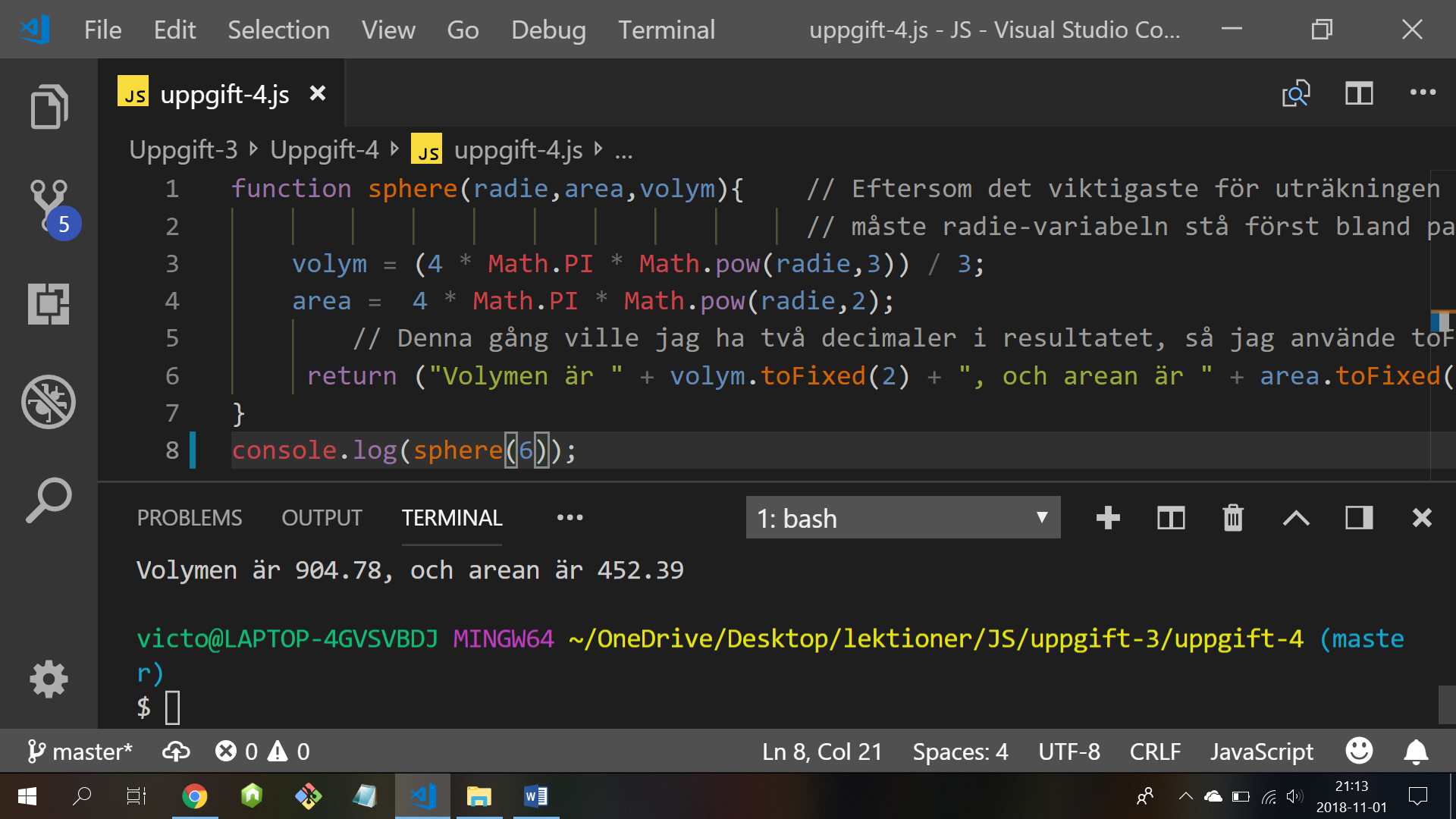
I den här uppgiften fick i alla fall jag möjlighet att slipa mina mattekunskaper ordentligt. Jag var faktiskt tvungen att kolla upp hur Pythagoras sats ska ställas upp, för det hade jag helt glömt bort. Men så fort den var på plats var det inga större svårigheter. Stommen i Pythagoras formel är att kunna beräkna potenser, vilket Math.pow-metoden hjälper till med. Denna tillsammans med möjligheten att kunna beräkna kvadratrötter (Math.sqrt-metoden), samt att ”läsa från höger till vänster” löste uppgiften.

Deluppgift 3

­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­

Precis som uppgift 2 handlade denna uteslutande om matematiskt tänkande. Användaren får ange radien av den cirkel denne vill beräkna area och omkrets av. Radien \* 2 hamnar i diameter-variabeln, diameter \* pi (Math.PI-metoden) resulterar i omkrets. Radien upphöjt i 2 \* PI blir arean. För snyggare utförande lade jag här till Math.round-metoden för att avrunda utfallet.

Deluppgift 4



Potensräkning återigen. Sfären har tre parametrar: radie, area och volym. Volym och area får sina värden genom att beräkna radien upphöjt i 3 respektive upphöjt i 2, dessa värden \* 4 \* PI (Math.PI-metoden). Radiens värde bestämmer i sin tur användaren, via console.log.

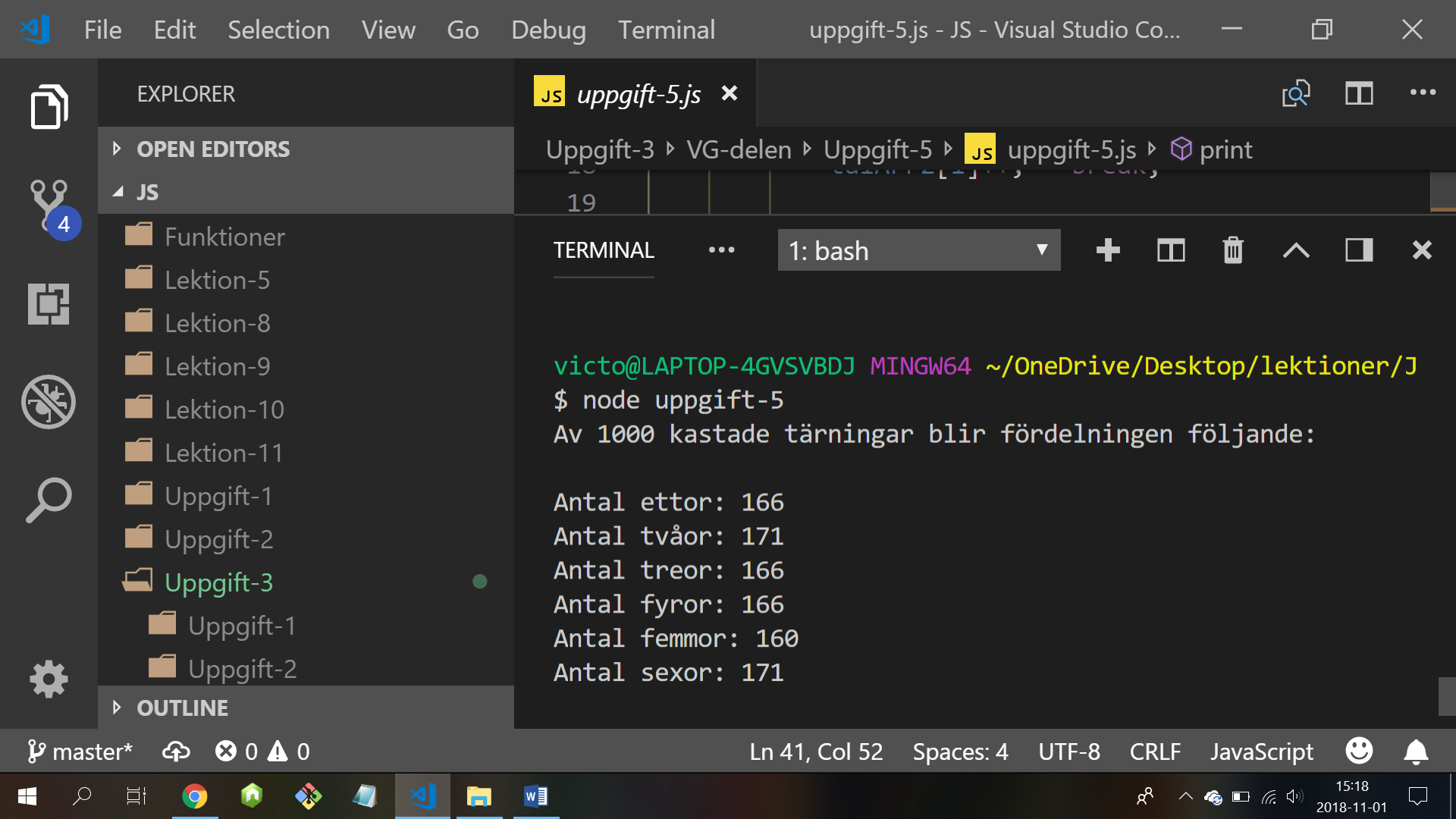
Deluppgift 5

Jag hade en hel del problem med den här uppgiften. Att skriva själva randomDice()-funktionen var inte det svåraste, där var min tanke redan från början klar om hur jag skulle gå tillväga – jag ville ha en tom array dit resultatet från varje tärningskast skulle hamna, beroende på vilken siffra som kom upp skulle resultatet indexeras i arrayen – samt en for-loop som håller räkningen – inte så knepigt.

Men sedan började jag göra för många fel – jag blandade ihop variabler, satte return innanför for-loopen istället för utanför (vilket tog ett tag innan jag upptäckte), bland annat. Sedan, när jag fått ordning på räknaren, började jag med print()-funktionen, vilket skulle visa sig vara ännu krångligare. Ett fel med randomDice-funktionen i början var också att loopen var felinställd, vilket gjorde räknaren oändlig. Det blev sammanlagt många omstarter av terminalen.

Jag började med att göra if-else-satser. Dock höll jag på länge utan att riktigt lyckas få in värdena från randomDice()-funktionen i print (vid det här laget har jag nästan glömt bort hur jag lyckades lösa problemet, har haft så många problem), innan en klasskamrat upptäckte att jag behövde ytterligare en array för att lagra värdena i print(), genom att kalla på randomDice()-funktionen i print()-funktionen (det hjälper tydligen om man läser instruktionerna till uppgiften ordentligt).

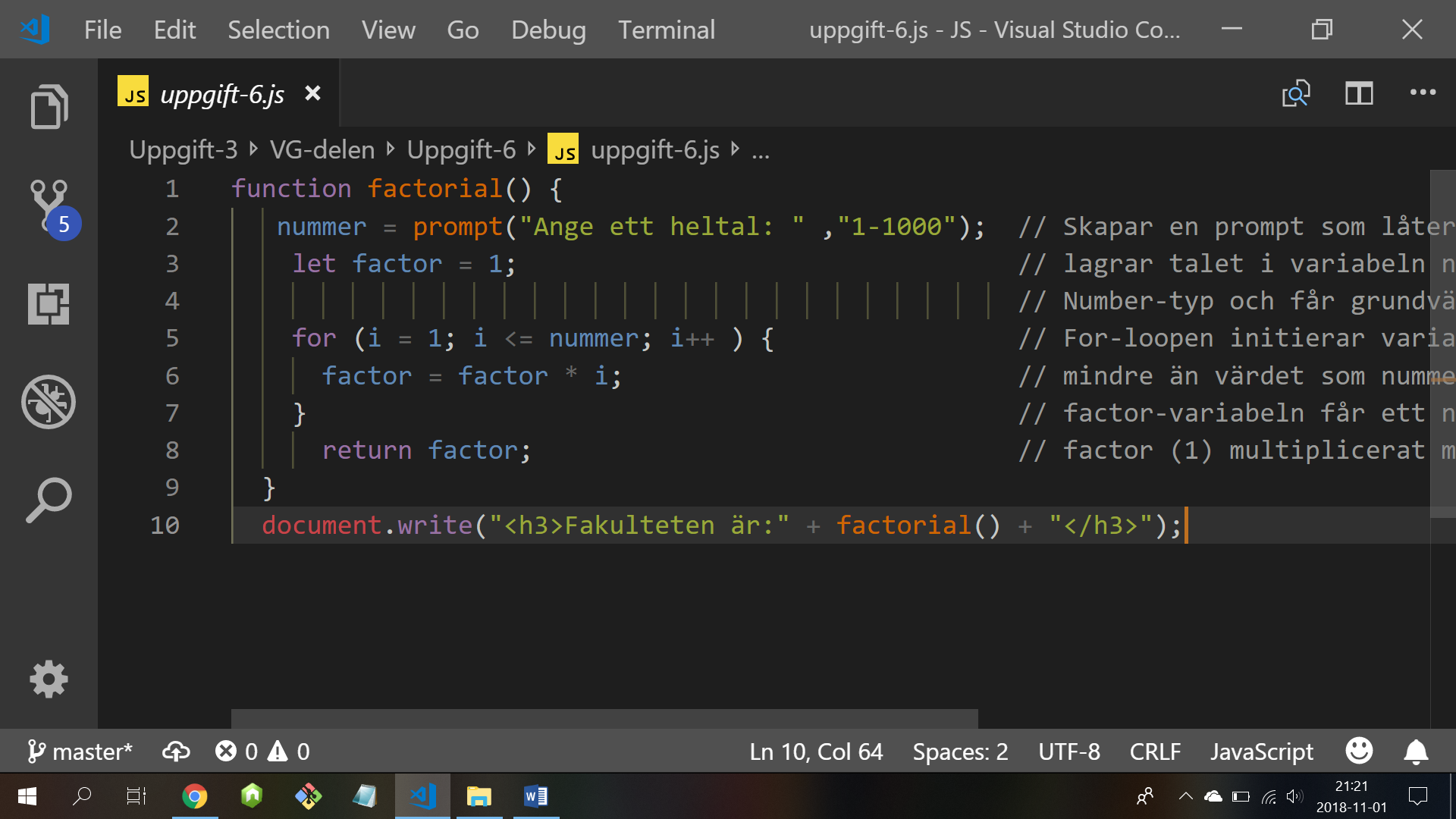
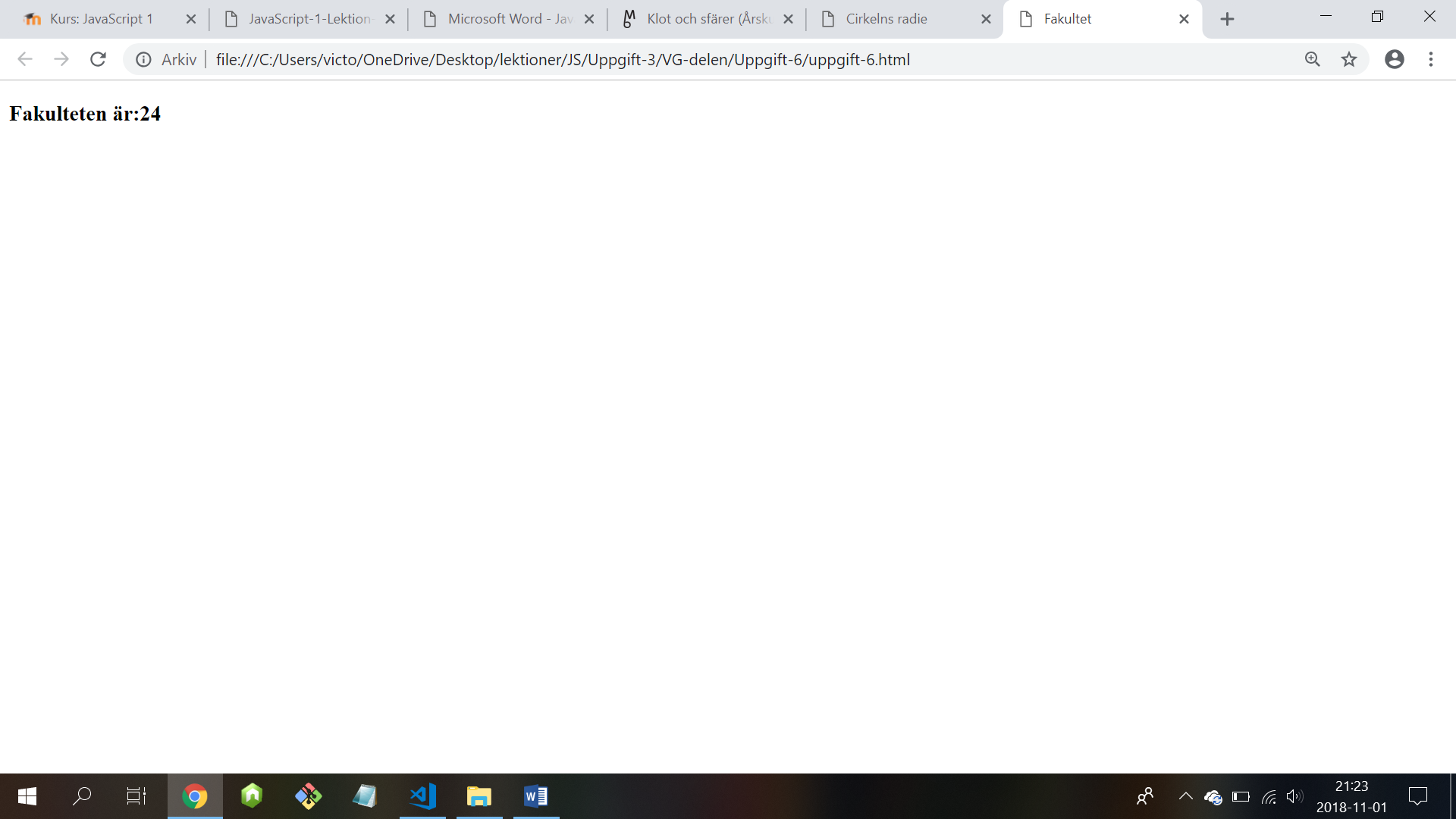
Sedan ändrade jag till att skriva print() med switch-satser istället. Jag gillar strukturen på switch bättre än if/else; det ser renare och snyggare ut, inte lika mycket kod.

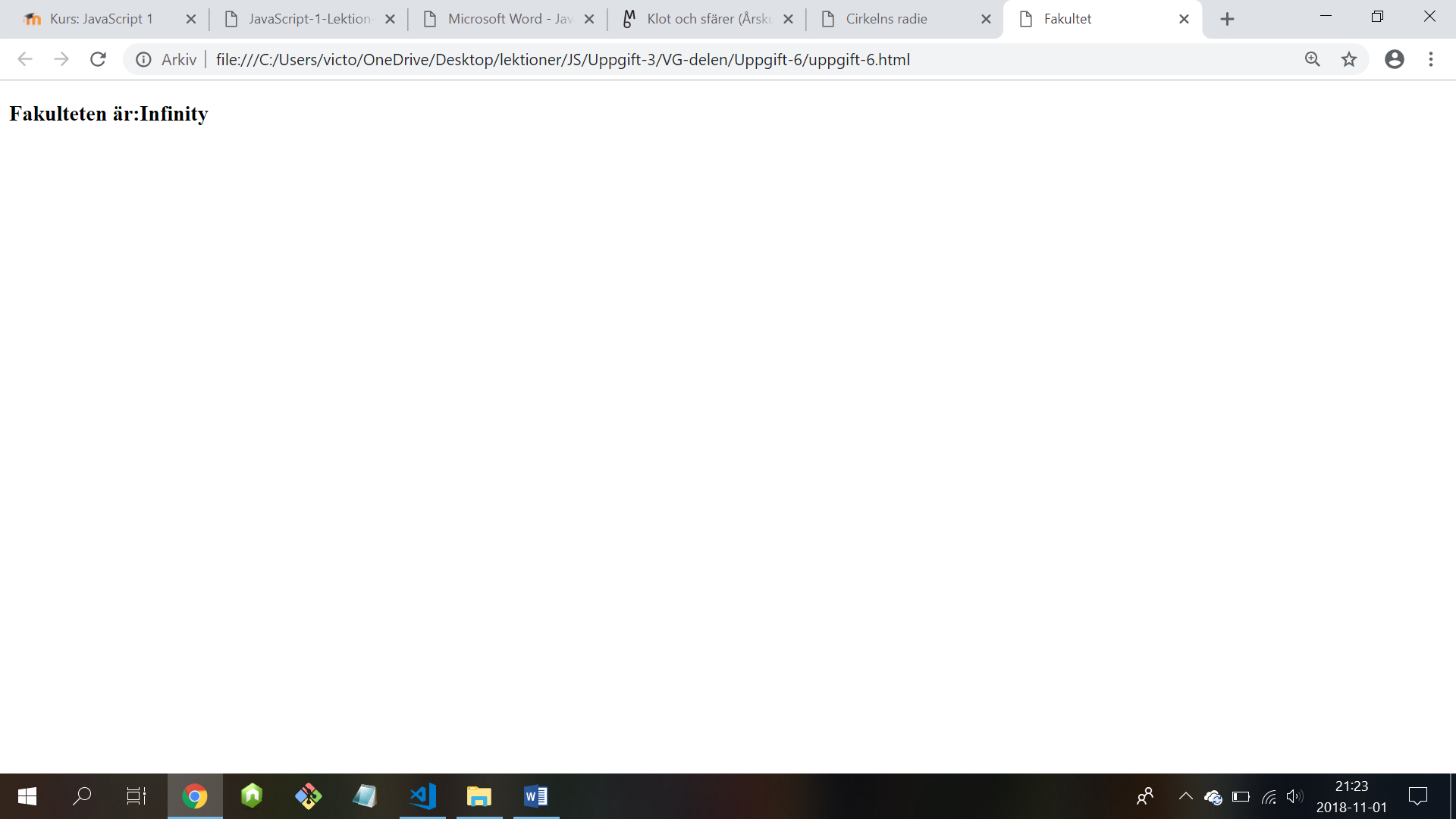


Vad jag har lärt mig:

Att console.logga mycket. Och ofta. Jag hade nog inte insett den faktiska nyttan med den metoden innan den här uppgiften, men nu vet jag bättre.

Uppgift 3 – deluppgift 6

 Fakultet var så gott som okänt för mig innan den här uppgiften. Alltid kul med nya lärdomar. Även här fick användaren mata in ett värde via prompt. Variabeln factor får värdet 1. En for-loop räknar (i) och tar värdet av prompten och inkrementerar i så länge som användaren bestämt. Det som sedan händer är att factor får ett nytt värde – vilket startar med 1 och multiplicerar en gång per loop, ända upp tills den siffra som angivits via prompt.

Problemet var ju bara att komma upp till siffran 1000, vilket jag inte lyckats lösa. Efter 170 visar räknaren bara Infinity.