

## Uppgift 5 – Frekvenstabell över tärningskast

### randomDice()

Först skapades funktionen `randomDice()`. Funktionen fick en parameter som kallas för `rolls`, detta för att man skulle kunna välja antal tärningskast funktionen ska utföra.

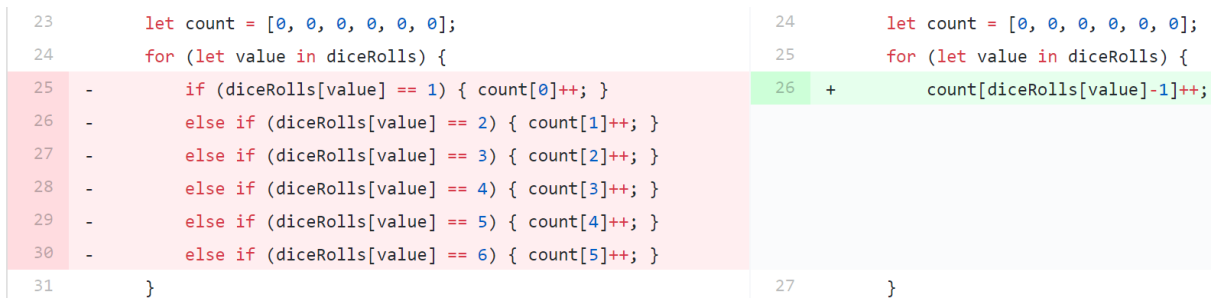
I funktionen initieras en array, som ska innehålla alla tärningskast. En `for`-loop används för att generera `x` antal tärningskast. För varje varv loopen pågår generereras ett slumpmässigt tal mellan 1-6, med hjälp av `Math.ceil` och `Math.random` multiplicerat med 6, värdena pushas till arrayen.

Funktionen returnerar sedan en array med `x` antal slumpmässiga tal mellan 1 och 6.

### print()

I steg 2 så skapades funktionen `print()`, som kallar på funktionen `randomDice()` med argumentet 1000. `randomDice()` returnerar sedan 1000 nummer i en array som sparas i `diceRolls`.

Därefter skapas arrayen `count` som ska hålla reda på antal förekomster av varje nummer i `diceRolls`. En `for-in` loop går igenom `diceRolls` och räknar varje förekomst av 1, 2, 3, 4, 5 och 6 och lagrar antalet i arrayen `count`. Här användes från början `if` och `if else` för att räkna förekomsten av av varje siffra, men efter tips från en gruppkamrats vän, kortades koden ner med några rader. Den nya lösningen blev då `count[diceRolls[value]-1]++`, här ökar värdet (antal förekomster) på det index `i` i arrayen `count` som representerar en siffra från `countDice`, antal 1:or på index 0, 2:or på index 1 osv.



```

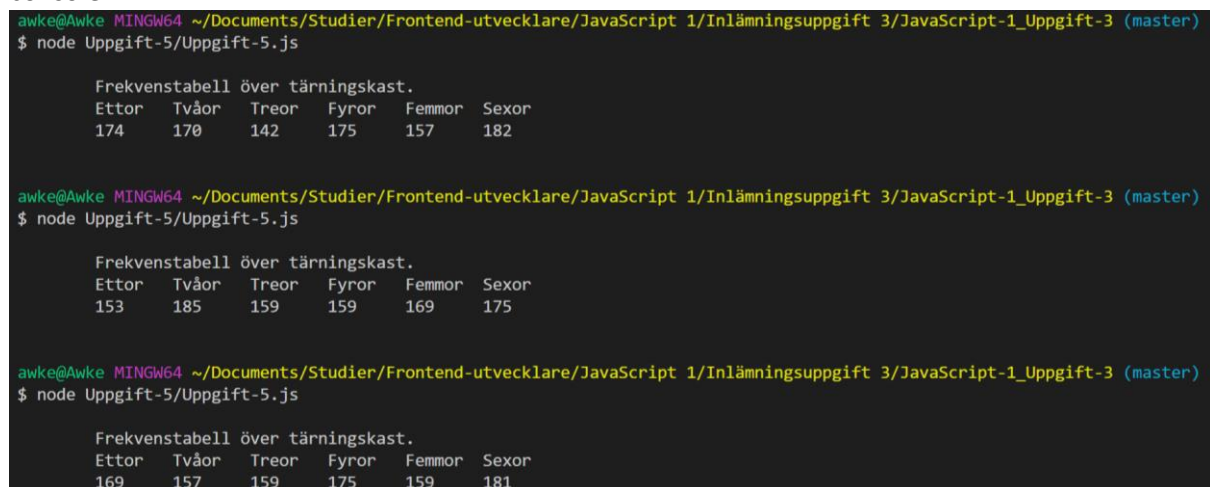
23     let count = [0, 0, 0, 0, 0, 0];
24     for (let value in diceRolls) {
25 -       if (diceRolls[value] == 1) { count[0]++; }
26 -       else if (diceRolls[value] == 2) { count[1]++; }
27 -       else if (diceRolls[value] == 3) { count[2]++; }
28 -       else if (diceRolls[value] == 4) { count[3]++; }
29 -       else if (diceRolls[value] == 5) { count[4]++; }
30 -       else if (diceRolls[value] == 6) { count[5]++; }
31     }

24     let count = [0, 0, 0, 0, 0, 0];
25     for (let value in diceRolls) {
26 +       count[diceRolls[value]-1]++;
27     }

```

Figur 1: Skillnaden mellan första och andra koden.

När funktionen räknat ut antal förekomster av samtliga siffror så skrivs resultatet ut i en tabell i console.



```

awke@Awke MINGW64 ~/Documents/Studier/Frontend-utvecklare/JavaScript 1/Inlämningsuppgift 3/JavaScript-1_Uppgift-3 (master)
$ node Uppgift-5/Uppgift-5.js

Frekvenstabell över tärningskast.
Ettor  Tvåor  Treor  Fyror  Femmor  Sexor
174    170    142    175    157    182

awke@Awke MINGW64 ~/Documents/Studier/Frontend-utvecklare/JavaScript 1/Inlämningsuppgift 3/JavaScript-1_Uppgift-3 (master)
$ node Uppgift-5/Uppgift-5.js

Frekvenstabell över tärningskast.
Ettor  Tvåor  Treor  Fyror  Femmor  Sexor
153    185    159    159    169    175

awke@Awke MINGW64 ~/Documents/Studier/Frontend-utvecklare/JavaScript 1/Inlämningsuppgift 3/JavaScript-1_Uppgift-3 (master)
$ node Uppgift-5/Uppgift-5.js

Frekvenstabell över tärningskast.
Ettor  Tvåor  Treor  Fyror  Femmor  Sexor
169    157    159    175    159    181

```

Figur 2: Körning av programmet.

## Reflektioner och slutsatser

Uppgiften var ganska enkel att lösa och sättet jag först löste den på kom ganska naturligt. En sak jag kom på i efterhand är dock att jag borde ha använt mig av en switch-sats istället för if och else if, då detta hade passat ändamålet bättre. Detta byttes ju dock ut mot en mer avancerad lösning med mindre kod, så i slutändan gör det inte så mycket.

När man kör programmet kan man tydligt se att det ger olika resultat varje gång, och räknar man ihop antal förekomster så resulterar det i 1000 kast.

## Uppgift 6 – Beräkna fakultet (factorial)

### fakultet()

Först skapades funktionen fakultet() med en parameter (tal) som har till uppgift att räkna ut fakultet på det argument som skickas via tal. Efter en hel del testande i console kom jag fram till en lösning för att räkna ut det.

Lösningen blev att ge variabeln f värde 1 och sedan med hjälp av en for-loop som upprepades så många gånger som parametern tal angav, köra satsen `f *= i;` där i representerar vilket varv det är. På så vis ökas värdet av i varje varv tills värdet av tal uppnås, varv 1:  $1*1=1$ , varv 2:  $1*2=2$ , varv 3:  $2*3=6$  osv, dvs ( $1*2*3$  osv).

Därefter returneras resultatet.

### Inmatning

För att testa funktionen skapades ett html-dokument och scriptfilen länkades.

I scriptfilen skapades först variabeln tal (som ska innehålla inmatat värde) och en while-loop som med hjälp av prompt frågade efter ett tal mellan 1-1000 ända tills korrekt inmatning utfördes. För att kontrollera att korrekt inmatning utfördes låg följande villkor inom paranteserna på while-loopen `(!(tal >= 1 && tal <= 1000) || isNaN(tal))`, vilket gjorde att loopen fortsatte så länge tal inte är mellan 1 och 1000 eller så länge talet inte är ett nummer. Utskrift på sidan skedde efter att loopen hade slutat. Detta fungerade riktigt bra, den enda nackdelen var att det inte gick att trycka på avbryt för att avbryta programmet.

För att lägga till möjligheten att avbryta fick koden ändras lite, tyvärr blev det fler rader men funktionsmässigt blev programmet bättre. I nya koden användes istället while (true), och villkor kontrollerades med if och if else för att antingen avbryta programmet eller skriva ut resultatet. Om inte korrekt inmatning sker, efterfrågas en ny. Här flyttades även utskriftskoden upp i loopen istället för att ligga under loopen som tidigare.

```
7 let tal;
8 // Fortsätter fråga efter ett tal tills kravet att talet är
  mellan 1 och 1000 uppfylls.
9 - while (!(tal >= 1 && tal <= 1000) || isNaN(tal)) {
10 -   tal = +prompt("Mata in ett tal mellan 1 och 1000, för att
    beräkna fakulteten.");
11 }

7 let tal;
8 // Fortsätter fråga efter ett tal tills kravet att talet är
  mellan 1 och 1000 uppfylls.
9 + while (true) {
10 +   tal = prompt("Mata in ett tal mellan 1 och 1000, för att
    beräkna fakulteten.");
11 +   if (tal === null) {
12 +     // Trycker man avbryt, skrivs meddelande ut
13 +     document.getElementById('fakultet').innerText =
        'Program avbrutet'; // och programmet avbryts.
14 +     break;
15 +   }
16 +   else if (Number(tal) >= 1 && Number(tal) <= 1000) {
17 +     // Om tal är mellan 1-1000 skrivs resultatet
18 +     document.getElementById('fakultet').innerText = tal +
        '! = ' + fakultet(tal); // ut med hjälp av funktionen
        fakultet().
19 +   }
20 }
```

Figur 3: Skillnad mellan första koden och den andra koden.

På den här sidan står det

Mata in ett tal mellan 1 och 1000, för att beräkna fakulteten.

Figur 4: Körning av programmet, inmatning.

## Uppgift 6 - Beräkna fakultet

$$5! = 120$$

Figur 5: Körning av programmet, resultatvisning.

### Reflektioner och slutsatser

Denna uppgift hade jag lite mer problem med. Det tog lite tid innan jag kom fram till hur jag skulle lösa uträkningen av fakultet. Inmatningen var dock lite enklare att lösa.

För att det inte skulle bli fel vid inmatning valde jag också att ha med kontroll av vad som matades in. Första försöket fick jag att fungera ganska snabbt, men när jag skulle lägga till möjligheten att trycka på avbryt, så tog det lite längre tid att få till det. I slutändan visade det sig att jag hade glömt att använda break inne i en av if-satserna, så även om jag matade in korrekt värde kördes loopen om igen.

Jag testade också mata in lite olika värden och kunde konstatera att matar jag in ett tal från 171 och uppåt så får jag resultatet Infinity, detta pga att JavaScript inte kan hantera så stora tal som resultatet blir. Från 22 och uppåt anges svaret med e+X (gångar 10 upphöjt till X).

### Tillägg efter muntlig redovisning

Efter den muntliga redovisningen har jag ändrat Uppgift 6, så att den visar felmeddelande vid felaktig inmatning eller om resultatet av fakultet är Infinity.