Arrayer och Loopar

**Namnge:**

Tänk på att namnge variabler och funktioner utifrån camelCase standarden. Dvs första ordet ska ha små bokstäver och varje nytt ord därefter ska ha stor första bokstav.

* Varför? För att det ökar läsbarheten och är standard i JavaScript.
* Det kan finnas olika standardi olika språk.
* Tänk på att namnge variabler utifrån informationen som sparas i don.
* Booleanska variabler namnge fördelaktigt med prefix: ”is”
* **Tänk att det ska vara som en beskrivande etikett.**
* Tänk på att namnge funktioner uifrån vad dom gör
* Funktionsnamn bör innehålla ett verb

**Object {}**

**Arrayer [ ] Är en data lista**

Index talar om vilken plats på listan element ligger

**I programmeringsvärlden börjar man alltid på 0. Index 0**

name.**length** = Visar hur lång en array är.

Name.**push** = Lägga till ett nytt element i listan, lägger sig sist i listan.

Name.**pop** = Då tar man bort i listan, börjar längt bak.

Name.**shift** - Tar bort först i listan

Name.**unshift** – Lägger till först i listan

**Looping**

Det är vanligt att man använder loopar ihop med en array.

**While loop**

Const value = 0;

while(value < 100) { // Är value mindre än 100? Ja!

Console.log(value);

value++; // Den ökar med 1

value-- // Den minskar med 1

}

Den kommer loppa tills den är över 100. Då kommer den hoppa ur loopen. Viktigt att värdet alltid ändrar sig. Det måste avslutas, annars kraschar det.

**For-each loop** (foor of loop)

(Denna loopen kör fyra varv, igenom hela listan)

const foods = [’banan’, ’äpple’, ’bönor’, ’majs’]

for (const foodItem of foods) {

Console.log(foodItem)

}

Om man inte är intresserad av placeringen är ”for-each” loopen enklare att använda. Notera nyckelorder ”of”.

For loop

Lite mer flexibel

Const listOfNames = [”Anna”, ”Sally”, ”Michaela”]

For(var i = 0; i < listOfNames.lenght; i++) {

Console.log(listOfNames[i];

brake betyder att man har hittat det man söker och då kan man avbryta sin loop med brake