

Hvad betyder det? Ud fra jeres intuition beskriv:

- **Hvad er formålet med dokumentation?**

Formålet med dokumentation er at beskrive ens kode eller program på en faglig måde så modtageren kan forstå hvad det går ud på.

- **Hvordan vil I gribe det an hvis I skulle beskrive et program og begrunde jeres valg?**

Vi ville starte med at læse koden i gennem også se om hvordan det virker alt efter hvordan vi forstår, det skriver vi hvad vi synes der er rigtig.

- **Hvilke værktøjer vil I bruge?**

Vi vil nok ikke rigtig bruge de vildeste værktøjer til at skrive en dokumentation udover, hvis vi f.eks. skriver dokumentationen på GitHub, kan vi anvende Markdown, som får teksten til at blive mere overskuelig og læsevenligt.

- **Stikord**

- Dokumentation med en faglig forståelse
- Struktureret tekst
- Skære ud i pap

Dårlig dokumentation

```
/// <param name="title" > The title of the CD </param>
/// <param name="author" > The author of the CD </param>
/// <param name="tracks" > The number of tracks on the CD </param>
```

Der er ingen grund til at skrive xml i ens kommentarer, samt så er funktionen meget selvforklarende med title, author, tracks.

```
// this class holds data for an employee
public class Employee {
    ...
}

```

```
// this function sends an email
void sendEmail() {
    ...
}
```

God dokumentation

```
// format matched hh:mm:ss EEE, MMM dd, yyy
Pattern timePattern = Pattern.compile("\\d*:\\d*:\\d* \\w*, \\w*, \\d*, \\d*");
```

Den er god fordi ét den er kort og to den er til at forstå ellers ville man tænke hvad er det man har lavet

```
/// <summary>
/// Checks to see if there's a comment. If so, returns a thank you message.
/// </summary>
function sourceCodeComment () {
    var comment = document.getElementById("Code Comment").value;
    if (comment != null && comment != '') {
        return console.log("Thank you for your comment.")
    }
}
```

Dette er en god men kort kommentar. Den beskriver overordnet hvad hele funktionen gør og hvad den returnerer.

```
// initialization of a List with each element of type Student, id and name.
List<Student> details = new List<Student>() {
    new Student{ id = 1, name = "Liza" },
    new Student{ id = 2, name = "Stewart" },
    new Student { id = 4, name = "Trish" }
};
```

Denne kommentar er god fordi den beskriver præcis hvad koden går ud på. Først laver de en klasse hvor der skal indeholde nogle specifikke studenter med deres id og navne.

```
// This function calculates prices, compares to sales
// promotions, checks if prices are valid, then
// send an email of promotion to user
public void doSomeThings() {
    // Calculate prices
    ...
    // Compare calculated prices with sales promotions
    ...
    // Check if calculated prices are valid
    ...
    // Send promotions to users
}
```

Denne kommentar er god fordi de skriver hvad den går ud på i toppen også i koden skriver de hvad er hvad og hvor det ligger.