

**Foodnest**  
**Grupp 16**

**Kodgranskning**  
**V. 1.9**  
**2021-04-15**

## Dokumenthistorik

Datum	Version	Beskrivning	Författare
210401	1.0	Påbörjade med syftet	Josefina Tranucharat Falk
210402	1.1	Skrev klart syftet och påbörjade ett stycke på processen	Josefina Tranucharat Falk
210403	1.2	Påbörjade skriva de generella riktlinjerna för namn konventionen	Josefina Tranucharat Falk
210407	1.3	Skrev klart de generella riktlinjerna samt de särskilda riktlinjerna för namn konventionen	Josefina Tranucharat Falk
210408	1.4	Fortsatte riktlinjerna för kod layouten	Josefina Tranucharat Falk
210412	1.5	Skrev klart alla riktlinjer för kod layouten och för kommentarerna. Uppdaterade även de särskilda riktlinjerna för namn konventionen. La in fler källor.	Josefina Tranucharat Falk
210412	1.6	Påbörjade skriva om processen	Josefina Tranucharat Falk
210413	1.7	Skrev klart processen och ändrade i syftet	Josefina Tranucharat Falk
210413	1.8	Skrev klart referenserna och ordlistan och fixade till små saker	Josefina Tranucharat Falk
210415	1.9	Tog bort referenser som va onödiga, tog bort namn i processen och la till text i riktlinje beskrivningen.	Josefina Tranucharat Falk
210420	2.0	La till text i riktlinjerna och la även in fler riktlinjer. La till ett ord i ordlistan och uppdaterade innehållsförteckningen	Josefina Tranucharat Falk

# Innehållsförteckning

Dokumenthistorik.....	1
Innehållsförteckning.....	2
Kodgranskning.....	3
Syfte.....	3
Ordlista.....	4
Referenser.....	5
Process.....	6
Riktlinjer för kod.....	8
Namn konventionen.....	8
Generella riktlinjer.....	8
Särskilda riktlinjer för Python.....	9
Särskilda riktlinjer för HTML.....	9
Särskilda riktlinjer för CSS.....	9
Särskilda riktlinjer för JavaScript.....	9
Särskilda riktlinjer för SQL.....	10
Kod layouten.....	11
Generella riktlinjer.....	11
Särskilda riktlinjer för Python.....	11
Särskilda riktlinjer för HTML.....	12
Särskilda riktlinjer för CSS.....	13
Särskilda riktlinjer för JavaScript.....	13
Särskilda riktlinjer för SQL.....	13
Kommentarer.....	14
Generella riktlinjer.....	14
Särskilda riktlinjer för Python.....	14
Särskilda riktlinjer för HTML.....	15
Särskilda riktlinjer för CSS.....	15
Särskilda riktlinjer för JavaScript.....	15
Särskilda riktlinjer för SQL.....	15
Granskningsprotokoll.....	16
Granskningsprotokoll 1.....	16
Granskningsprotokoll 2.....	16

# Kodgranskning

## Syfte

Syftet med detta dokumentet är att gruppen ska få en klarhet för hur man ska skriva kod korrekt för att kunna minimera risker för framtida problem. Det ska finnas riktlinjer för hur Python, SQL, HTML, CSS och JavaScript ska och inte ska skrivas. Detta ska ge gruppmedlemmarna vägledning för hur koden ska få en så bra struktur som möjligt. Det kommer även underlätta för alla utomstående att läsa och förstå koden.

Det finns en ordlista för ord som kommer att förekomma i dokumentet för att göra det enklare för en läsare som inte är insatt i projektet att förstå termer som inte är uppenbara.

En referenslista finns tillgänglig där det enkelt ska gå att spåra den angivna informationen som finns i detta dokument.

Det innehåller även instruktioner för hur processen av kodgranskningen ska gå till. Processen ska vara så tydlig och detaljerad som möjligt så att det ska gå att utföra processen utav den beskrivningen.

Syftet med processen är att hitta potentiella problem och se till så att koden följer de standard som finns.

Riktningarna finns för att se till så att den standard som finns följs. Riktlinjerna består utav hur namngivningen ska vara, hur placeringen av all kod ska se ut och även hur kommentarerna ska vara. Att följa riktlinjerna bidrar till bättre struktur och bättre läsbarhet i koden.

Checklista som ska användas vid gransknings mötet är i detta projekt riktlinjerna, som granskningsgruppen kommer att utgå ifrån.

Slutligen finns det protokoll dokumenterat från granskningarna. Det ska finnas minst två protokoll från minst två separata granskningar. Dessa protokoll ska vara så detaljerade som möjligt så att personer som inte deltagit i granskningen ska utifrån dessa kunna åtgärda de problem som hittats.

I protokollen ska det dokumenteras i vilken fil och rad det har uppstått ett fel men även vad för sort problem det är och hur allvarligt problemet är. Det ska även innehålla information om vem som ska åtgärda det eventuella problemet.

Det är viktigt att det står vilken version av koden det är som har granskats så det inte finns risk för att en person åtgärder problem på en version där en annan person redan har granskat och åtgärdat på.

# Ordlista

<b>Alias</b>	Det används för att ge tabeller eller kolumner ett temporärt namn. Förkortningen för alias är AS.
<b>Block kommentar</b>	En block kommentar är en kommentar i Python där man kommenterar en specifik rad kod med hashtag”#”.
<b>CamalCase</b>	Det används vid sammansatta ord istället för mellanslag eller understreck. Exempelvis JavaScript.
<b>CSS</b>	Är ett språk som används för att styla en webbsida.
<b>HTML</b>	Är ett språk som används för att skapa en webbsida.
<b>JavaScript</b>	Världens mest populära språk inom webbutveckling. Med JavaScript kan man implementera komplexa funktioner
<b>Python</b>	Ett programmeringsspråk
<b>Review process</b>	En process som granskar aktiviteter och ser till att kvalitet hålls och ska upptäcka eventuella problem.
<b>SQL</b>	Ett språk för att hantera databaser.
<b>Zoom</b>	En plattform där olika personer kan ha videosamtal med varandra. Främst används den för konferenssamtal.

## Referenser

[1] A. Jain, "Difference Between Single and Double Quotes in Python", *AskPython*, [Online]. Tillgänglig:

<https://www.askpython.com/python/string/difference-between-single-and-double-quotes-in-python>

Hämtad: 2021-04-12

[2] B. Warsaw, G. Van Rossum, N. Coghlan, "PEP 8 -- Style Guide for Python Code", *Python*, [Online]. Aug. 01, 2013.

Tillgänglig:

<https://www.python.org/dev/peps/pep-0008/#naming-conventions>

Hämtad: 2021-04-03

[3] CKAN, "CSS coding standards", [Online].

Tillgänglig:

<https://docs.ckan.org/en/ckan-2.7.3/contributing/css.html>

Hämtad: 2021-04-07

[4] I. Sommerville, *Software engineering: Software management*. 10. uppl., Edinburgh Gate, Storbritannien: Pearson, 2015.

[5] S.Holywell, "SQL Style Guide", *SQL Style*, [Online].

Tillgänglig:

<https://www.sqlstyle.guide/#naming-conventions>

Hämtad: 2021-04-07

[6] W3Schools, "HTML Style Guide and Coding Conventions", [Online].

Tillgänglig:

[https://www.w3schools.com/html/html5\\_syntax.asp](https://www.w3schools.com/html/html5_syntax.asp)

Hämtad: 2021-04-07

[7] W3Schools, "JavaScript Style Guidelines", [Online].

Tillgänglig:

[https://www.w3schools.com/js/js\\_conventions.asp](https://www.w3schools.com/js/js_conventions.asp)

Hämtad: 2021-04-12

[8] W3Schools, "SQL Comments", [Online].

Tillgänglig:

[https://www.w3schools.com/sql/sql\\_comments.asp](https://www.w3schools.com/sql/sql_comments.asp)

Hämtad: 2021-04-12

## Process

Processen kommer att genomföras enligt Review [4], den kommer även att anpassas för detta projekt på så sätt att gransknings mötet kommer att ske via Zoom på grund av pandemin, Covid-19. Deltagarna som kommer att delta är personer från detta projekt, Foodnest.

Syftet med review processen är att den ska omfatta alla delar som krävs för en granskning. Under processen får alla deltagare en överblick över den skrivna koden, hur den är uppbyggd och om den följer de standard som finns. En granskning som denna ska projektgruppen göra för att upptäcka eventuella fel så tidigt som möjligt. Detta ska underlätta för hela teamet genom bättre förståelse av kod skrivning men samtidigt hindra framtida problem som kan uppstå. En fördel med att upptäcka problem i koden tidigt är att det kommer spara gruppen en del tid. Vid allvarliga problem skulle en konsekvens kunna vara att en alltför stor del av koden som skulle behöva ändras och det skulle inte gruppens tidsbudget klara av.

Processen kommer gå till på följande:

1. Första steget är så kallat *Pre-review activities*. Här sker planeringen och förberedelserna. För kodgranskningen är det den som är ordförande som sköter detta. Uppgifterna är att sätta ihop ett review team och sedan bestämma tid och plats för själva kodgranskningen.

Under denna tiden kommer teamet ha tillgång till dokumentet med instruktioner på hur processen kommer att gå till men även alla riktlinjer och all kod. Gruppmedlemmarna får läsa och kolla igenom allt så de förstår vad som ska göras.

Sedan ska varje gruppmedlem individuellt arbeta med kodgranskningen för att se om det finns problem eller potentiella framtida problem i koden men även om de följer de angivna riktlinjerna. Teamet får även varsin individuella mall på vad som behövs fyllas i under granskningen. Allt som hittas ska dokumenteras för att kunna gås igenom gemensamt under *the review meeting*. Detta är även bra för resten av gruppmedlemmarna att ha tillgång till vid eventuell frånvaro från det gemensamma mötet.

Det är viktigt att medlemmarna granskar en och samma version av koden så att det inte sker några problem eller förvirringar under gransknings mötet

2. Därefter sker *The review meeting*. När mötet startar får teamet kort information om hur det ska gå till sedan ska all kod gås igenom tillsammans med de alla individuella

kommentarerna och anmärkningarna från alla gruppmedlemmarna. Processen ska inte ta mer än sammanlagt två timmar, så de koder som har högst prioritet och ska utföras först är Python, HTML och SQL, men det är viktigt att all kod ska granskas.

Det ska finnas en i teamet som är ordförande som leder mötet och en som är sekreterare som ska dokumentera alla ändringar som behövs göras. Slutligen ska den som är ordförande även sammanställa ett dokument med alla kommentarer och åtgärder som teamet har kommit överens om under granskningen som ska tillämpas.

3. Sista steget är *Post-review activities*. Efter att process mötet är slut ska alla ändringar göras. Detta kan vara allt från små- till stora åtgärder. Hela teamet ska ta del av anmärkningarna och lösa problemen men det är den som är ordförande som ser till att alla ändringar i kodningen görs.

När ändringarna är gjorda ska ordförande kontrollera så att alla dokumenterade kommentarer faktiskt är åtgärdade. Vid vissa tillfällen kan det även behövas en granskning till över ändringarna av kommentarerna.



## Riktlinjer för kod

Riktlinjerna är framtagna från andra riktlinjer [1]-[3] & [5]-[8] med en del anpassningar.

Syftet med riktlinjerna är att alla gruppmedlemmar ska skriva kod på samma sätt så att det blir en gemensam struktur. Vid tillfällen där en riktlinje har flera sätt att skriva kod, har en av dessa valts ut som en riktlinje för detta projekt på grund av tidsbudget och för att undvika otydligheter. Det finns även extra riktlinjer som lagts till av relevans.

Förkortningarna för dessa ID står för följande:

- Första bokstaven står för vilken sorts kod det är det vill säga, G = generella, P = Python, H = HTML, C = CSS, J = JavaScript och S = SQL.
- Resterande bokstäver står för vad det är för sorts riktlinje så att säga, NK = Namn Konventioner, KL = Kod layouten och K = Kommentarer.

Det finns riktlinjer som är tillagda i efterhand på grund av att det uppkommit oklarheter under kodgranskningarna där det inte fanns en riktlinje för just det problemet. Dessa riktlinjer finns inte med i referenserna som givits och kommer därför vara skapta riktlinjer för just detta projekt.

Vid tillfällen där en riktlinjer gäller på majoriteten av koderna skrivs riktlinjen under generella riktlinjer och sedan under de koder riktlinjen inte gäller hos skrivs den under den särskilda kodens rubrik.

## Namn konventioner

Namn i detta sammanhanget är alla namnen som kan skapas, det vill säga funktioner, variabler, filnamn, etc. Dessa är anpassade för projektet på så sätt att det ska vara enklare att skilja riktlinjerna åt för alla koder.

### Generella riktlinjer

**GNK1:** Filnamnet ska vara kort med endast små bokstäver.

**GNK2:** Namnet ska vara på engelska.

**GNK3:** Namnet ska inte vara för generella eller för bokstavliga och läsaren ska förstå vad namnet står för eventuellt vad den gör.

**GNK4:** Namnet ska inte innehåller inga oförutsägbara förkortningar.

**GNK5:** Namnet ska inte vara för långt och inte vara för beskrivande.

Bra:

harrypotter\_filmer (exempel som är bra för python & SQL)

Dåligt:

en\_lista\_för\_alla\_harry\_potter\_filmer

**GNK6:** Namnet ska endast vara i små bokstäver.

**GNK7:** Bokstäverna “l” (litet L), “I” (stort i) och “O” (stort o) ska inte användas som ensamma bokstäver i ett namn.

**GNK8:** I ett namn med flera ord ska understreck “\_” eller bindestreck “-” användas där det vanligtvis skulle varit ett mellanslag.

**GNK9:** Alla namn ska vara korrekt stavade.

**GNK10:** Det ska inte finnas några särkrivningar.

### Särskilda riktlinjer för Python

**PNK1:** Vid flera ord i ett namn ska orden separeras med understreck “\_”.

### Särskilda riktlinjer för HTML

**HNK1:** Vid flera ord i ett namn ska orden separeras med bindestreck “-”.

### Särskilda riktlinjer för CSS

**CNK1:** Vid flera ord i ett namn ska orden separeras med bindestreck “-”.

### Särskilda riktlinjer för JavaScript

**JNK1:** Vid flera ord i ett namn används camelCase.

`function` toCelsius

**Särskilda riktlinjer för SQL**

**SNK1:** Namnet ska inte innehålla mer än 30 bokstäver.

**SNK2:** Namnet ska börja med en bokstav.

**SNK3:** Namnet ska inte sluta med ett understreck “\_”.

**SNK4:** Namnet ska endast innehålla bokstäver, siffror och understreck “\_”.

**SNK5:** Namnet ska inte innehålla onödigt många understreck “\_”.

**SNK6:** Ett tabellnamn ska inte innehålla prefix som är beskrivande.  
Exempelvis: tbl.

**SNK7:** Namnet på en tabell ska inte vara samma namn som en kolumn.

**SNK8:** Namnet på en relationstabell ska inte vara ett sammansatt namn av två tabeller.

Bra:

filmer

Dåligt:

twilight\_frozen

**SNK9:** Namnet på kolumner ska vara ett enda namn.

Bra:

skådespelare

Dåligt:

alla\_skådespelare

**SNK10:** Vid användning av korrelationer (alias) ska första bokstaven av varje ord från objektet vara korrelationens namn.

```
SELECT first_name AS fn
```

**SNK11:** Vid tillfällen där det redan finns en korrelation med ett visst namn tillsätts en siffra.

```
FROM staff AS s1  
JOIN students AS s2  
ON s2.mentor_id = s1.staff_num;
```

**SNK12:** Syntaxer ska alltid vara i stora bokstäver.

## Kod layouten

### Generella riktlinjer

**GKL1:** Det ska inte finnas kod som är felplacerade

**GKL2:** Koden ska innehålla mellanslag både innan och efter “=”.

### Särskilda riktlinjer för Python

**PKL1:** En indentering ska vara fyra mellanslag.

**PKL2:** Där innehållet börjar bli för långt innanför parenteserna “( )” ska indragen göras enligt nedan.

```
def saker(sak_ett, sak_två,  
          sak_tre, sak_fyra):  
    return sak_ett
```

**PKL3:** Där innehållet börjar bli för långt innanför de hårda parenteserna “[ ]” ska indragen göras enligt nedan.

```
filmsamling = [  
    1, 2, 3,  
    4, 5, 6,  
    7, 8, 9,  
]
```

**PKL4:** En rad med kod ska inte innehålla mer än 79 tecken.

**PKL5:** En rad med kommentarer ska inte innehålla mer än 72 tecken.

**PKL6:** “import” ska vara längst upp bland koden.

**PKL7:** Vid citering av ord eller namn ska enkla citattecken ‘ användas.

```
sentence = 'string_one'
```

**PKL8:** Vid citering av text ska dubbla citattecken “ användas.

```
hey = "Hello World!"
```

**Särskilda riktlinjer för HTML**

**HKL1:** En indentering ska vara två mellanslag.

**HKL2:** Koden innehåller en <title>.

**HKL3:** Varje element stängs med </

**HKL4:** Vid citering ska det användas dubbla citattecken.

```
input type="password"
```

**HKL5:** Koden ska inte innehålla mer än nödvändiga mellanslag.

Bra:

```
input type="password" id="psw" name="psw"
```

Dåligt:

```
input type = "password" id = "psw" name = "psw"
```

**HKL6:** Koden ska inte innehålla mer än nödvändiga blankrader om flera elementer tillhör varandra.

Bra:

```
<h1>Populära städer</h1>

<h2>New york city</h2>
<p>New york city är en stad som...</p>
```

Dåligt:

```
<h1>Populära städer</h1>

<h2>New york city</h2>

<p>New york city är en stad som...</p>
```

**Särskilda riktlinjer för CSS**

**CKL1:** En indentering ska vara två mellanslag.

**CKL2:** Vid citering ska det användas dubbla citattecken.

```
.media {  
  background: #FFF url("blablalba")  
}
```

**CKL3:** Efter kolon ":" ska det finnas ett mellanrum.

```
overflow: hidden;
```

**CKL4:** Det ska finnas ett mellanrum innan måsvingen "{".

```
.media {
```

**CKL5:** Vid användning av färger ska hexadecimal användas med en hashtag "#" innan.

```
background: #FFF
```

### Särskilda riktlinjer för JavaScript

**JKL1:** En indenteringen ska vara två mellanslag.

```
function toCelsius(fahrenheit) {  
  return (5 / 9) * (fahrenheit - 32);  
}
```

**JKL2:** En rad med kod ska inte innehålla mer än 80 tecken.

**JKL3:** Det ska finnas ett mellanrum innan måsvingen "{".

```
function toCelsius(fahrenheit) {
```

### Särskilda riktlinjer för SQL

**SKL1:** Mellanrum ska användas när likamedtecken "=" används.

```
WHERE f.name = 'Anna'
```

**SKL2:** Vid citering ska det användas enkla citattecken.

```
WHERE l.name = 'Andersson'
```

**SKL3:** Efter ett kommatecken "," ska det finnas ett mellanrum.

```
SELECT f.name, l.name
```

## Kommentarer

### Generella riktlinjer

**GK1:** Kommentarer ska vara i fulla meningar.

**GK2:** Första bokstaven i meningen ska vara en stor bokstav.

### Särskilda riktlinjer för Python

**PK1:** Kommentarer med en hashtag “#” ska kommenteras som en block kommentar.

```
# The family has animals  
animals = cat + dog
```

**PK2:** Block kommentarer ska innehålla ett mellanslag mellan hashtaggen “#” och texten.

Bra:

```
# The family has animals
```

Dåligt:

```
#The family has animals
```

**PK3:** Block kommentarer ska användas för att kommentera koden på raden under kommentaren.

**PK4:** Kommentarer som beskriver funktioner ska kommenteras med tre citattecken """.

**PK5:** Kommentarer som är för långa för en rad och ska kommenteras med tre citattecken """ och ska delas upp i fler rader. De tre citattecken """ som stänger kommentaren ska vara på en egen rad utan kommentar.

```
"""  
Denna funktionen hämtar valuta  
USD = x SEK  
EURO = x SEK  
"""
```

**PK6:** Kommentarer som beskriver funktioner ska vara under `def` raden.

### Särskilda riktlinjer för HTML

**HK1:** Kommentarer ska kommenteras med `<!--` och avslutas med `-->`.

```
<!-- Din kommentar -->
```

**HK2:** Vid längre kommentarer ska kommentarerna vara uppdelade i flera rader. `<!--` och `-->` ska ha egna rader utan kommentarer på samma rad.

```
<!--  
    Detta är en lång kommentar. Detta är en lång kommentar  
    Detta är en lång kommentar. Detta är en lång kommentar  
-->
```

### Särskilda riktlinjer för CSS

**CK1:** Kommentarer ska börja med `/*` och avslutas med `*/`

**CK2:** Vid längre kommentarer ska kommentaren vara uppdelade i flera rader. `/*` och `*/` ska ha egna rader utan kommentarer på samma rad.

### Särskilda riktlinjer för JavaScript

**JK1:** Kommentarer ska kommenteras med två snedstreck `/**`.

**JK2:** Vid kommentarer inuti en kod ska det kommentaren börjas med `/*` och avslutas `*/`

### Särskilda riktlinjer för SQL

**SK1:** Korta kommentarer ska kommenteras med två bindestreck `--`.

**SK2:** Vid längre kommentarer ska kommentarerna börja med `/*` och avslutas med `*/`.

`/*` och `*/` ska vara på samma rad som kommentarens början respektive slut.

```
/*SELECT * FROM Movies;
```

```
SELECT * FROM Actors;
```

```
SELECT * FROM Genre;*/
```



# **Granskningsprotokoll**

## **Granskningsprotokoll 1**

[da336a\\_kodgranskning1\\_foodnest\\_vt21](#)

## **Granskningsprotokoll 2**

[da336a\\_kodgranskning2\\_foodnest\\_vt21](#)