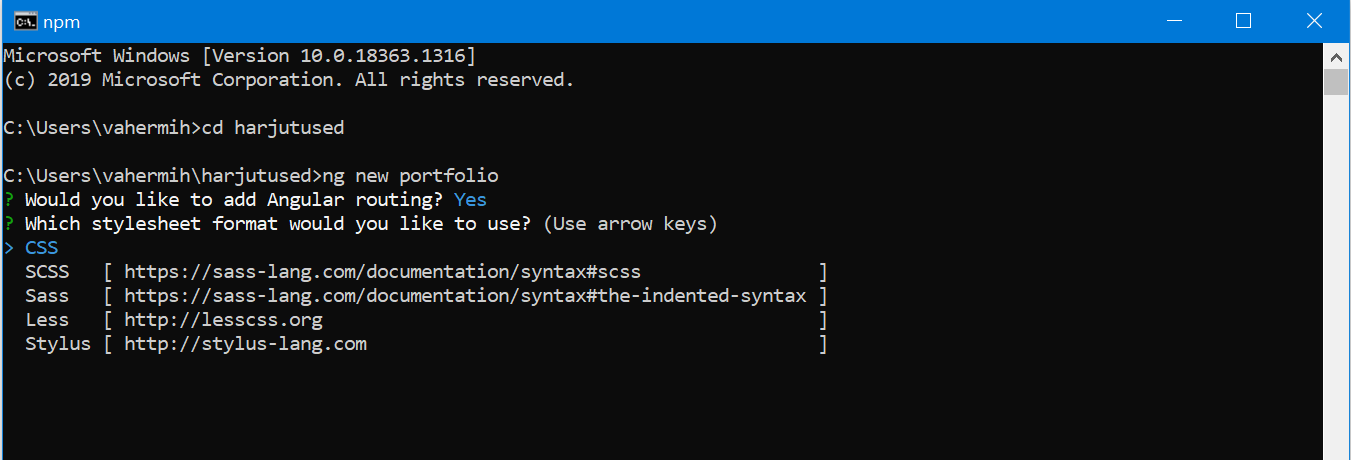
1. Tee uus projekt käsureal (ehk command line / käsuviip ( ehk command prompt))



Kõiki käske on võimalik näha kui kirjutada „ng“

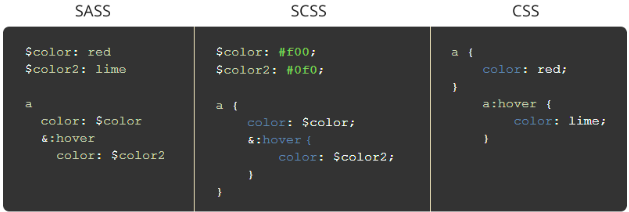
„**ng new PROJEKTI\_NIMI**“ tekitab uue projekti

**Angular routing** lisab app-routing.module faili ja <router-outlet></router-outlet> täägi app.component.html’i

(võid katsetada korra ja teha uue projekti vastates selle küsimusele „no“ ning jälgida mis on teistmoodi)

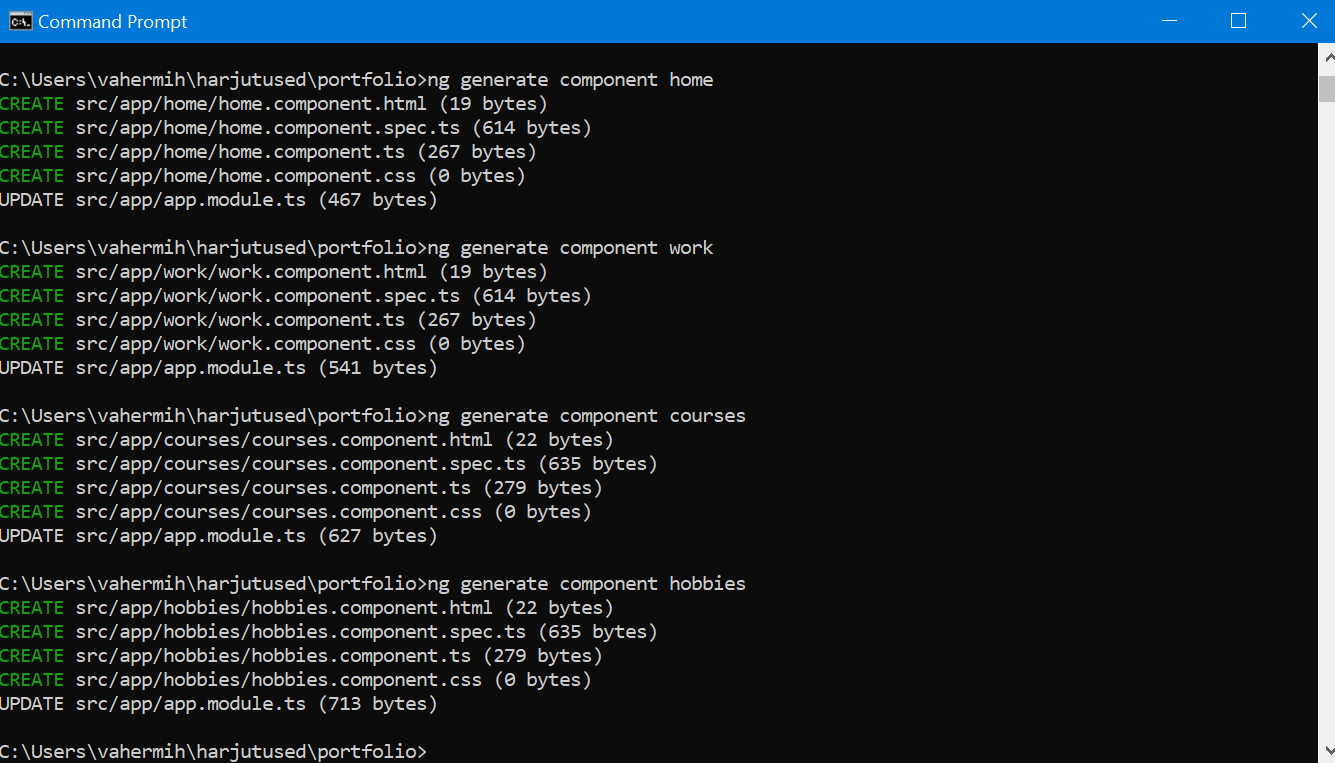
Kui siia vastata „no“, siis saab pärast failid manuaalselt tekitada

Kasutame CSS’i, teised on CSS’i laiendused. Stiilinäide teistest:





1. Tee neli Componenti



On näha, et tekitati igas Componendis neli faili:

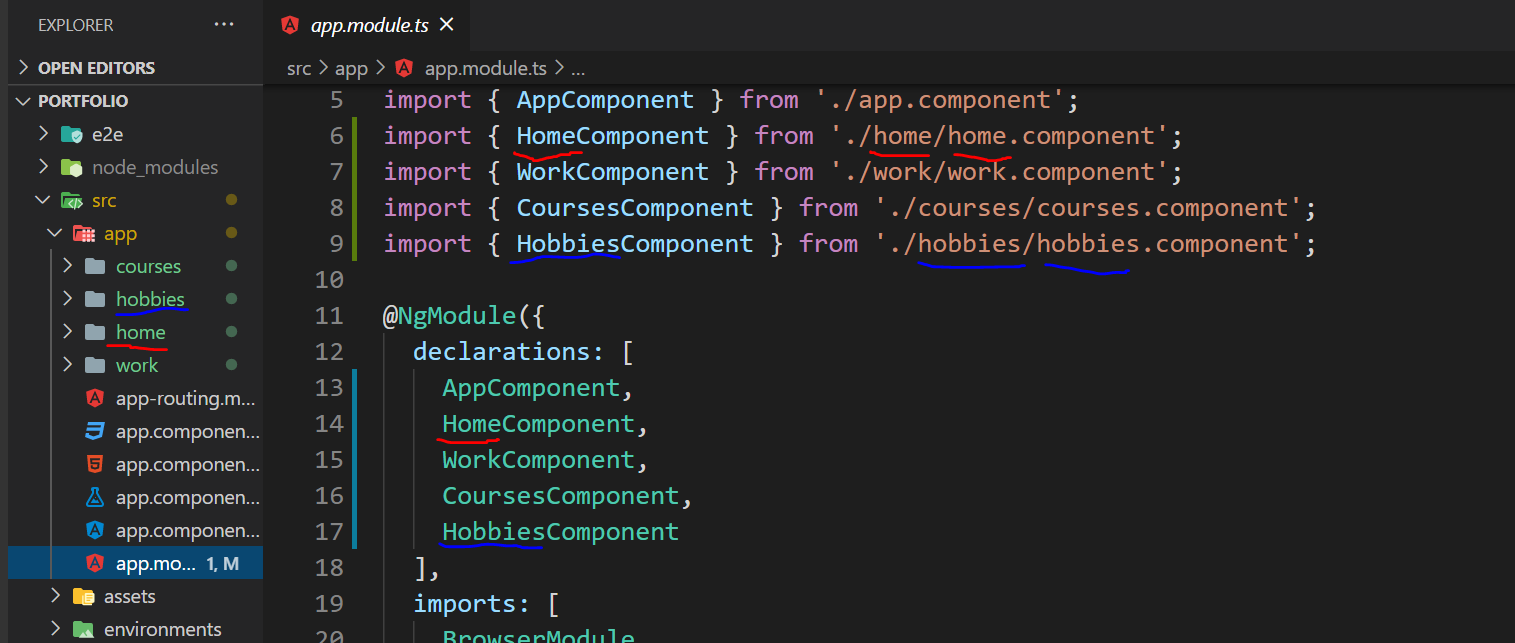
**.html** lõpuga (seda näeb kasutaja)

**.css** lõpuga (kujunduse jaoks)

**.ts** lõpuga (selle abil anname informatsiooni html faili)

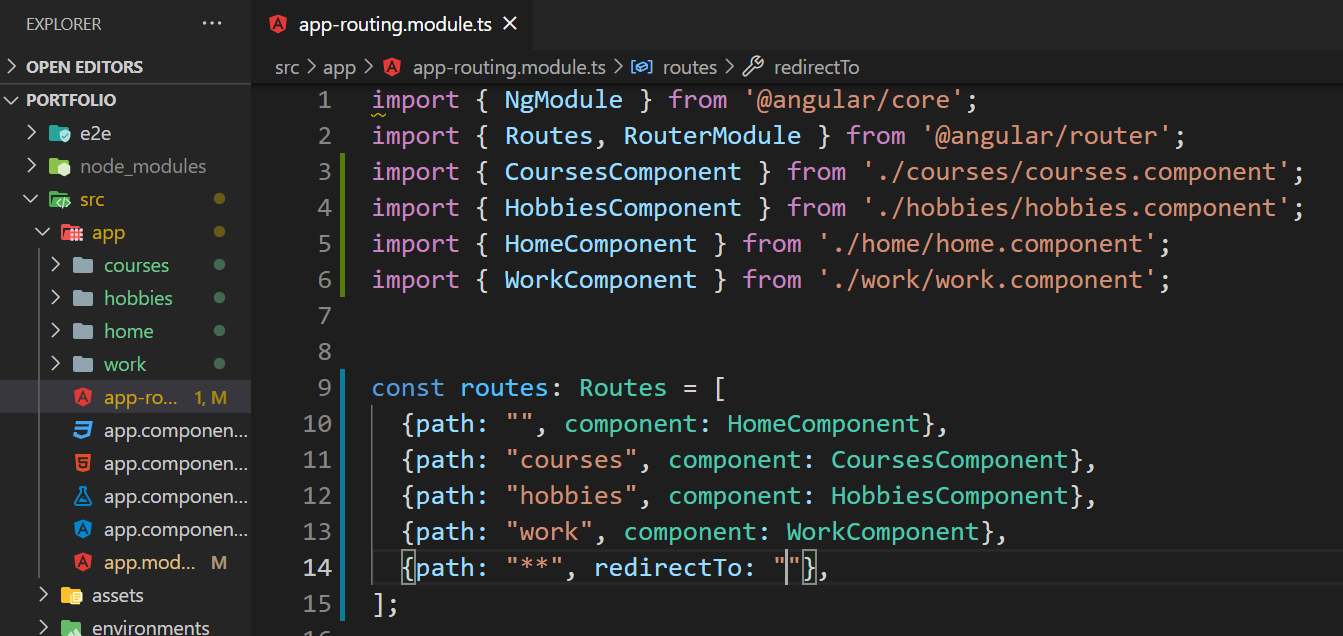
**.spec.ts** lõpuga (seda selle koolituse raames ei käsitle, tegemist on testimiste jaoks vajalike failidega)

Lisaks tehti UPDATE **app.module.ts** faili ehk deklareeriti iga Componendi genereerimisel see moodulisse. Seda võib kontrollida ka app.module seest:



Vasakul on kaustad kõik tähestikulises järjekorras. Failis on kõik genereerimise järjekorras (mis kõigepealt genereeriti, see on ülevalpool, vt ka genereerimise järjekorda ülal).

1. Lisa kõikidele Componentidele vastavad route-d ehk URL lingid



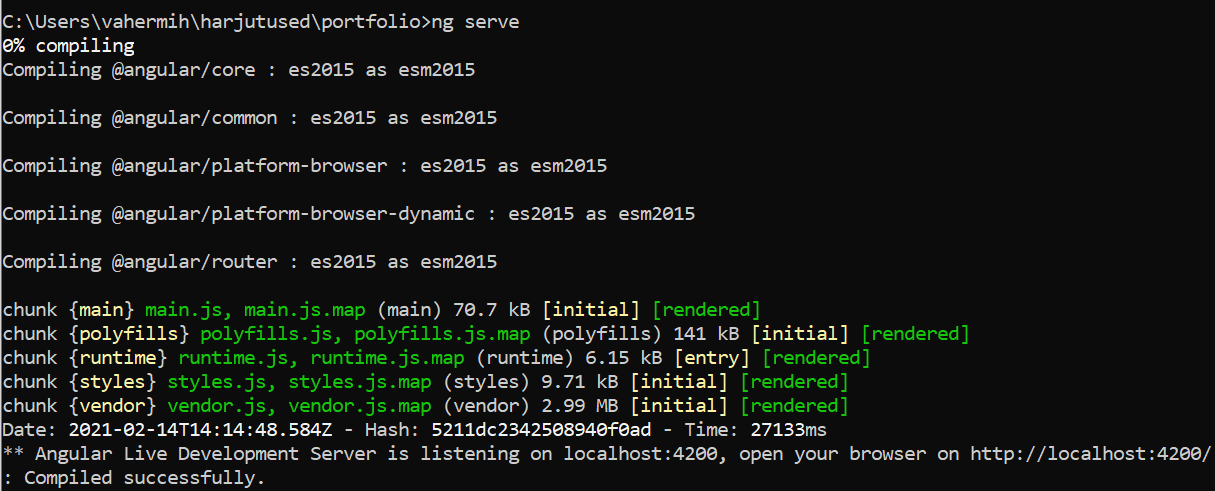
**Path:** mis lisatakse localhost:4200 lõppu

**Component:** mis component sellele vastavaks pannakse

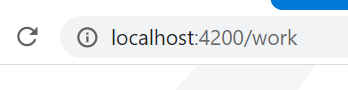
**RedirectTo:** kuhu URL-le see suunatakse

**„\*\*“:** kõik ülejäänud – sellest allapoole pole mõtet ühtegi route’i panna

1. Mine lehele ja vaata, kas kõik töötab korrektselt



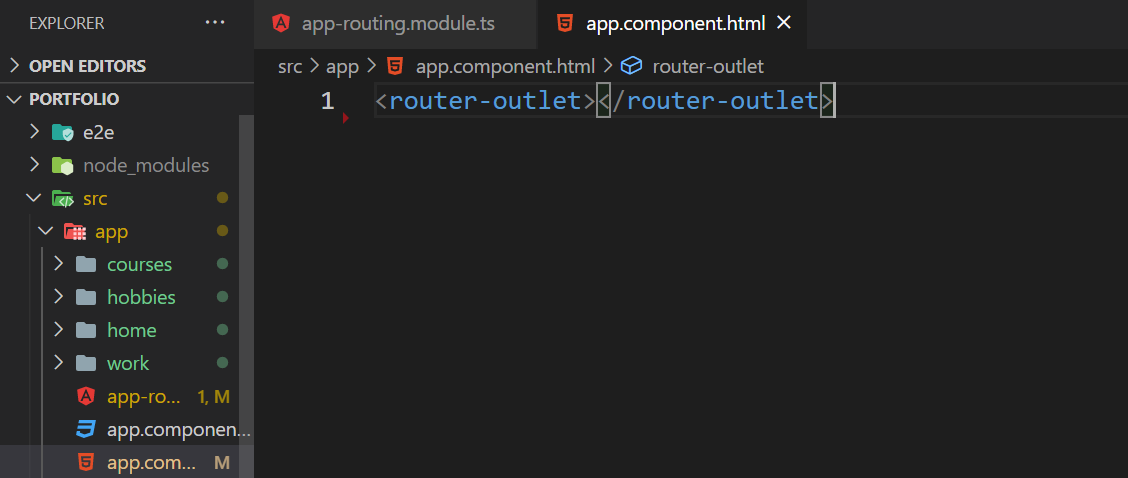
Kogu HTMLide sisu tuleb hetkel lehe allaossa:



Ja suvaline URL peaks suunama avalehele (localhost:4200/Google **->** localhost4200)

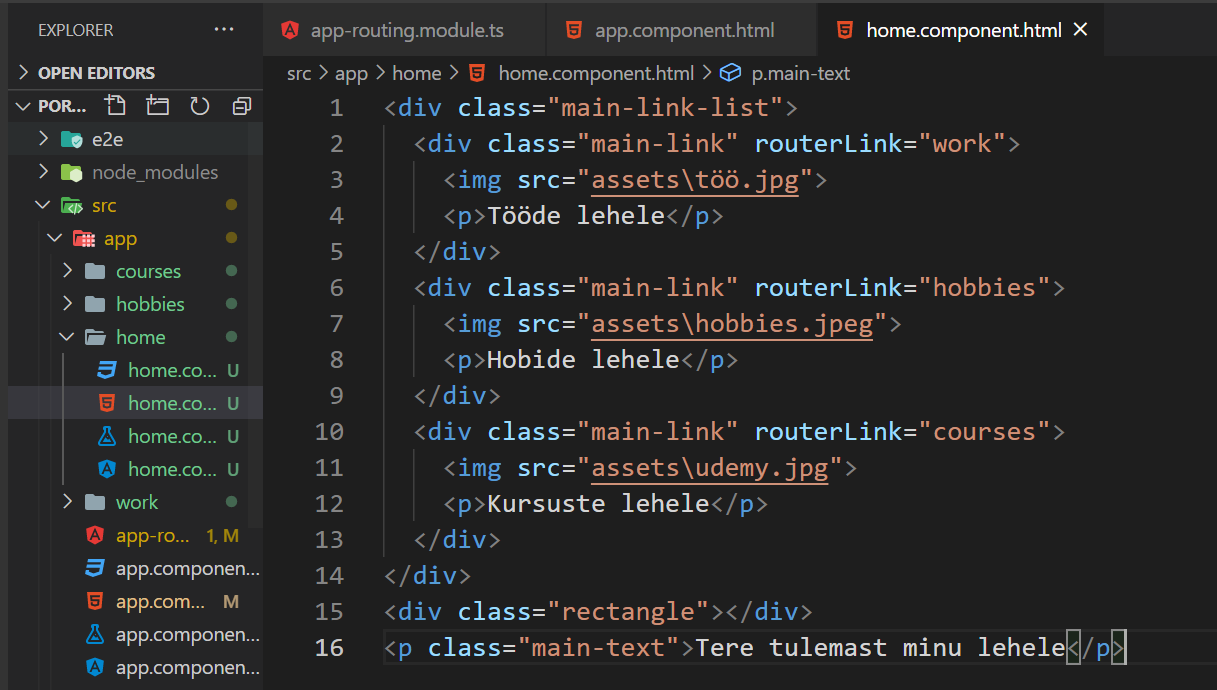
Tekst tuleb lehekülje allaossa, sest <router-outlet></router-outlet> lisab selle, mis **URLil me oleme**, mis **Componendi me sellele URLile vastavusse määrasime** ja mis selle **Componendi HTML failis** on.

1. Kustuta app.component.html seest kogu sisu ja jäta alles vaid router-outlet



Kontrolli nüüd iga URL peal (localhost:4200/**hobbies**, localhost:4200/**work**, localhost:4200/**courses**), mis lehele tuleb – vaata, mis on **app-routingu sees** vastavuses sellele **URL**ile ja siis omakorda, mis on selle **Componendi HTML faili sees**. Proovi neid omavahel vahetada ja katsetada, mis juhtub kui paned Work asjad Hobbies sisse ja Hobbies asjad Courses sisse kas HTMLis, app-routingus võid Componenti või URLi vahetada ning katsetada.

1. Täidame HomeComponendis ära html faili

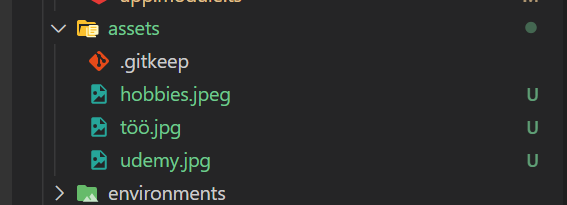


<div> <- division, koodiilu, HTML parameetrite panemiseks (class=““, routerLink)

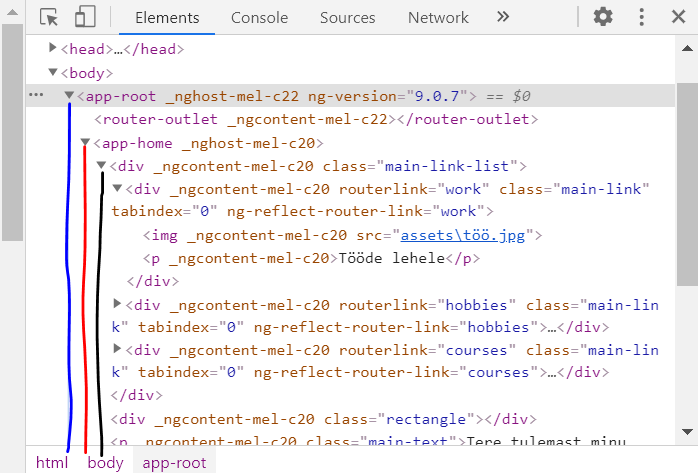
Class=““ <- hetkel CSSi ehk stiili panemise jaoks

<p> - paragraph

Lisa assets kausta ka pildid:



Lehele localhost:4200 minnes on nüüd pildid mittesobivates mõõtudes (kuna me pole kujundust ehk CSSi lisanud), aga saame vaadata Chrome-i sees elemente:



Siin on näha kogu Angulari struktuur – kõik mis on millegi sees, on ühe võrra paremal, kui nad on samal laiusel, siis nad on „võrdsed“:

<app-root> sees on <router-outlet>, mis võttis kasutusele <app-home>, kuna me oleme localhost4200 lehel

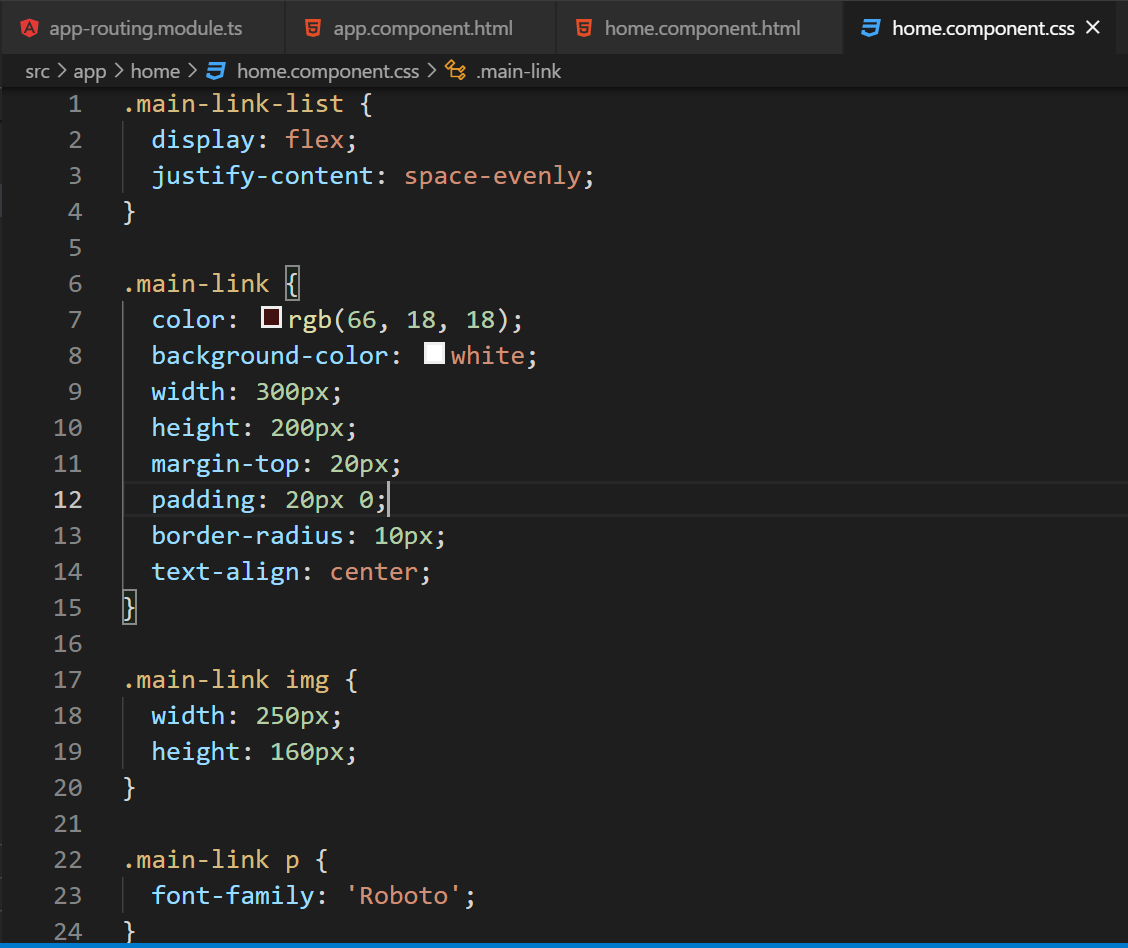
<app-home> sees on <div>, millel on klass „main-link-list“ ja mis ümbritseb kõiki ülejäänud div’e (millel on klass „main-link“)

Esimesel tasandil on div, millel on klass „main-link“ ning seda lahti tehes näeme, et seal on <img> ja <p> täägid (ingl k „tag“).

Koodiosa vaadates on näha, et meil on täpselt sama struktuur. Angular lisab omakorda täägidele ngcontent direktiivid (oranžis värvis), millega on Angularil ligipääs neile täägidele.

Järgmiseks saame anda HTMLi kujundust nende klasside kaudu.

1. Täida [home.component.css](http://home.component.css)



Display: flex 🡪 kõik ELEMENDID selle sees on **vasakult paremale**

Justify-content: space-evenly 🡪 need elemendid mis on vasakult paremale, nende vahel olev **ruum (space) jaotatakse** nii, et see on kõikide ümber võrdselt

Katsetage, pannes sinna asemele:

Justify-content: space-between;

Justify-content: space-around;

Color: 🡪 **tekstivärv**, katsetage teiste värvidega

Background-color: 🡪 **taustavärv**, katsetage teiste värvidega

Margin 🡪 **lükatakse** midagi sellest **elemendist** **kaugemale** igas suunas.

Margin-top 🡪 **lükatakse** midagi sellest elemendist kaugemale **ülespoole**. Katsetage pannes 200px;

Padding 🡪 **lükatakse** selle elemendi **sisemust sissepoole**. Kui on 1 väärtus, siis igas suunas, kui on kaks väärtust, siis esimene väärtus on üles-alla ja teine väärtus vasakule-paremale, kui on neli väärtust, siis on üles-paremale-alla-vasakule (nagu kell).

Katsetage Paddingut:

padding: 200px 0;

padding: 20px 20px;

padding: 0 200px;

padding: 200px;

padding: 0 20px 60px 40px;

Border-radius 🡪 mis kohast hakatakse „**kurv**i keerama“ – katsetage 100px;

Kui tahta ringi ristküliku asemel, siis on võimalik kirjutada:

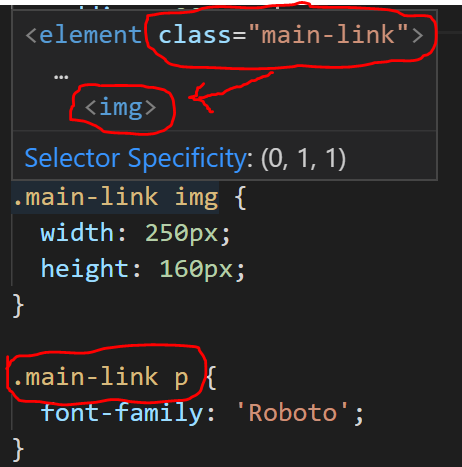
Border-radius: 50%; (keskelt hakatakse „kurvi keerama“)

Text-align: center; tekst läheb **keskele**

Width: **kõrgus**, katsetage suuremate/väiksemate pikslitega

Height: **laius**, katsetage suuremate/väiksemate pikslitega

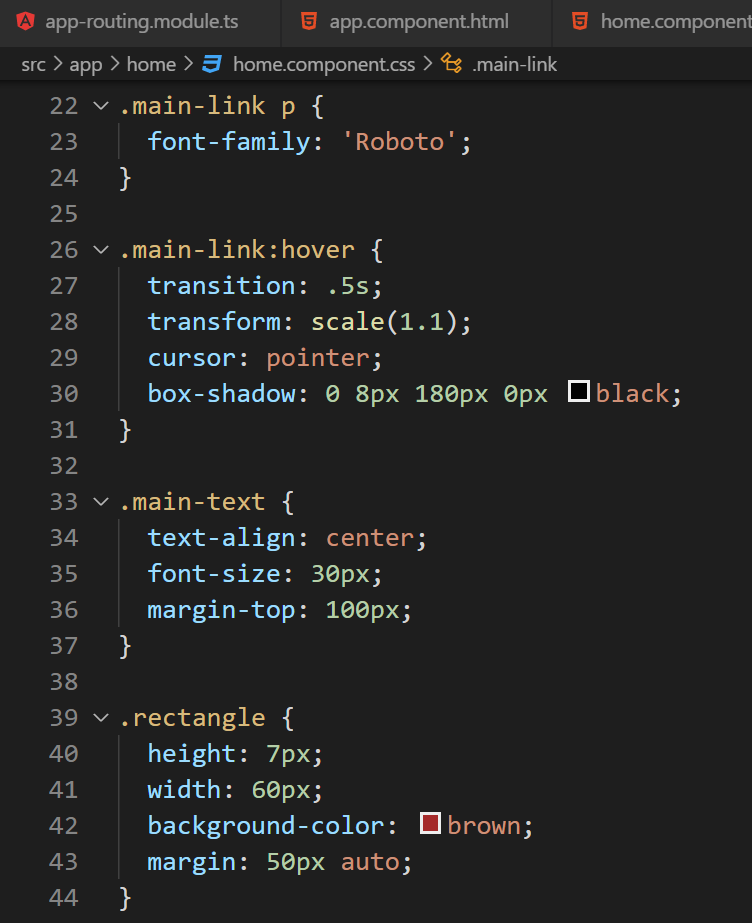
Hoidke hiirt „**main-link img**“ peal, saamaks aru mis tähendab **tühik** nende vahel



Hoidke hiirt „main-link p“ peal, näete sarnast asja. Minge vaadake HTML faili ning näete ka loogikat, miks on niimoodi kirjutatud -> tahame anda stiili paragraphile või image-le.

ME VÕIME ka selle asemel kirjutada class=“main-link-image“ HTMLi ja me ei pea kasutama .main-link img struktruuri, aga me säästame end ühe klassi lisamisest – KUIGI mõlemad viisid töötavad täpselt sama hästi ja korrektselt.

Kuna meil on vaid üks pilt HTMLis, siis võime ka kirjutada lihtsalt img (jätta klassi ära), kuid kui pilte tuleb koodiossa juurde, siis on hea täpsustada, kus täpselt see pilt asub.



:hover 🡪 **hiirega üleminek**

transition: .5s 🡪 mitu sekundit võtab allolev koodilõik **aega**. Katsetage pannes:

transition: 1s;

transition: 5s;

transition: .1s;

transform: scale(1.1); 🡪 **transformeeritakse** (muudetakse) olemasolevat 1.1korda suuremaks. Katsetage pannes:

transform: scale(0.6);

transform: scale(6);

cursor: pointer; 🡪 vajutatav käemärk kursoril, Katsetage teisi **kursorimärke**:

cursor: cell;

cursor: grab;

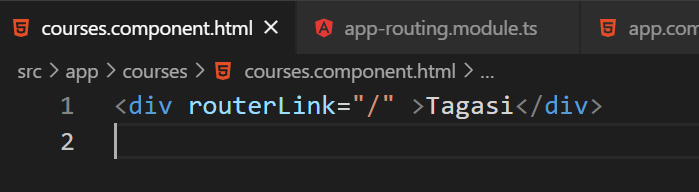
cursor: progress;

cursor: wait; jne

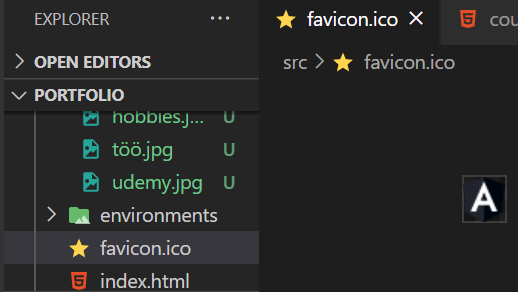
box-shadow: **vari kasti ümber.** Katsetage teiste värvide ja teiste (ekstreemsete) pikslitega

margin: auto; 🡪 auto ehk „**brauser leia ise** margin“, paneb sobivates olukordades keskele.

1. Igasse Componenti võib nüüd lisada HTMLi ossa ka tagasimineku:



1. Pisipildi muutmiseks võta internetist suvaline pilt ja asenda ära see fail:



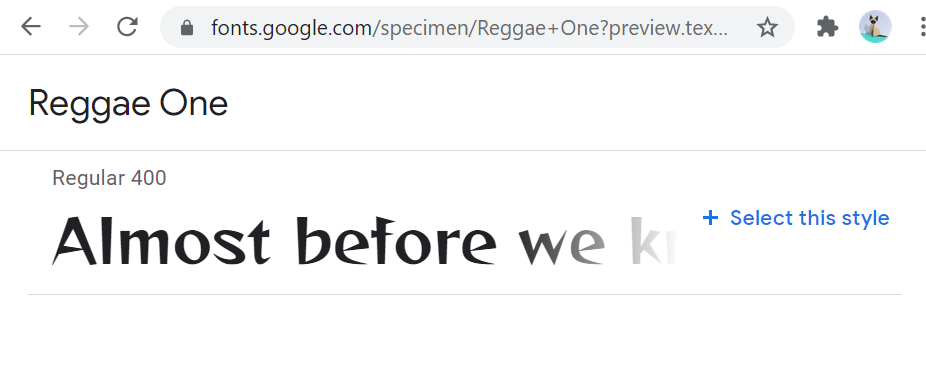
Pane uue faili nimeks ka **favicon.ico**, tõsta see favicon.ico peale ja vajuta „jah“ soovin asendada

1. Omale sobiva Fonti lisamine

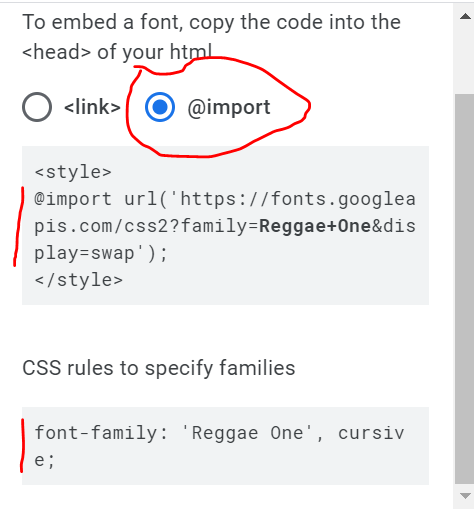
Mine lehele fonts.google.com

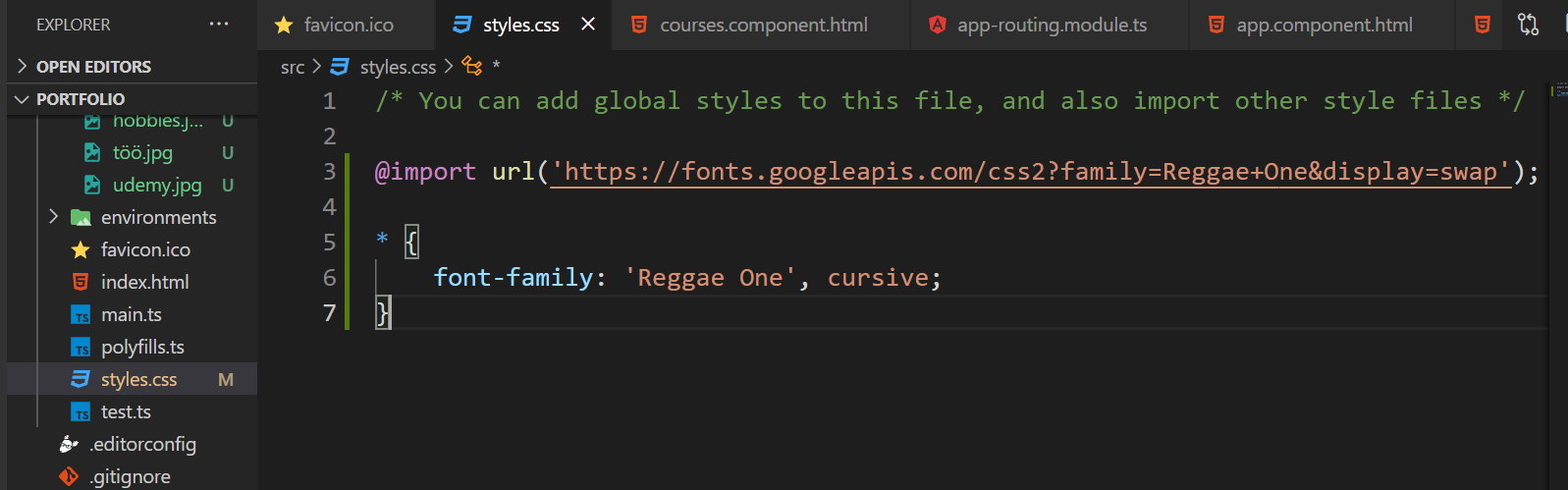
Vali sealt sobiv Font ja kliki sellele

Seejärel vajuta „Select this style“



Vajuta @import peale ja võta sealt @import osa, mis lisa styles.css faili. Alumine „font-family“ lisa sobivatele täägidele või klassidele, või kõigele (\*)





**ÜLEJÄÄNUD LEHED JÄÄVAD SOBIVAKS TEHA ISESEISVALT.**

Alati on võimalik minu poole küsimustega pöörduda, kuidas mida teha – kui tekivad ideed, aga ei oska neid realiseerida.

**Mõned näited**:

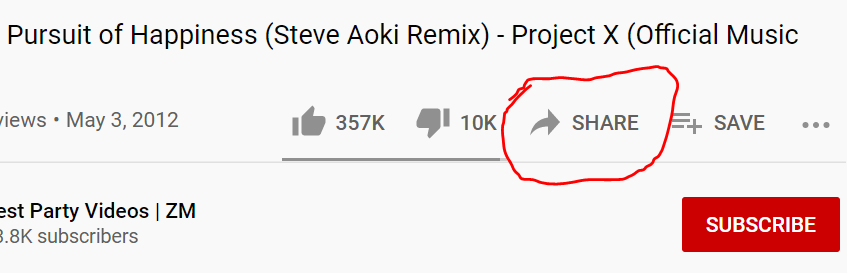
Lisa „assets“ kausta PDF fail.

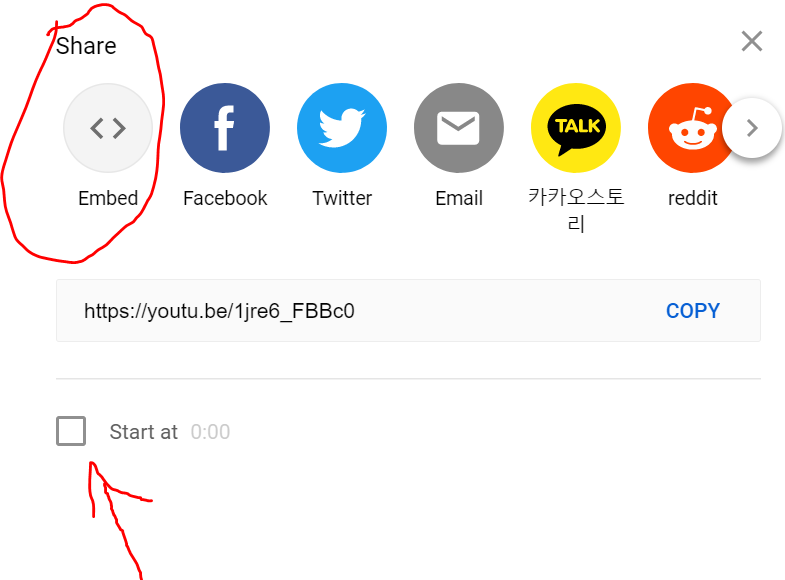
HTMLI:

**<a href="assets\Dokumendihaldus- konspekt.pdf" target="blank">Vaata mu tehtud tööd</a>**

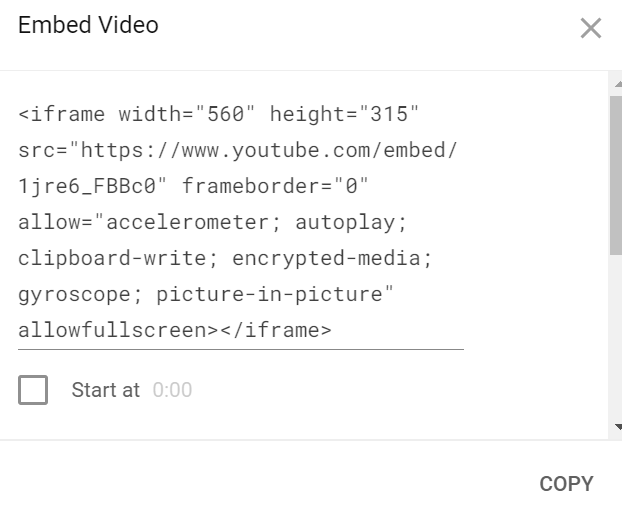
Target=“blank“ tähendab, et see avatakse uues aknas. Võite selle maha võtta, siis suunab nupuvajutus kasutajat uuele lehele olemasolevat lehelt

Youtube video lisamine:





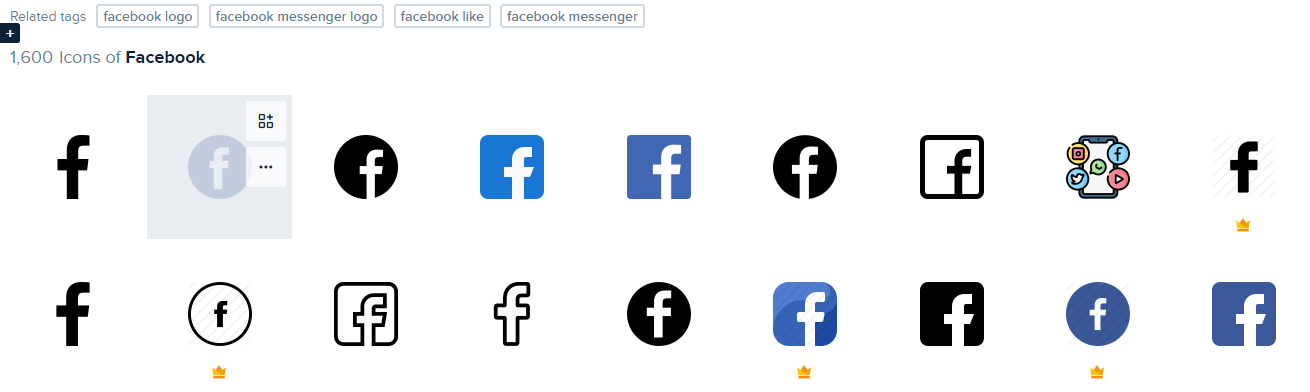
(kui tahad alustada mingilt kindlalt sekundilt)



Kopeeri koodilõik enda HTML faili. Võid kõike proovida muuta – width, height asemele teised numbrid, allow=““ jutumärkide sees midagi ära kustutada või terve allow ära kustutada.

Ikoonid allaossa:

Võta flaticon.com sobivad ikoonid ja lae see alla png failina. Jälgi, et ei oleks „krooni“ märki ehk tasulist ikooni.



Otsi see üles Downloads kaustast ja lisa see „Assets“ kausta.

Võta see kasutusele:

