**Index Leerdocument**

Dart

1. Keyboard Shortcuts

Git

**Dart 1.**

void main ( ) { }

* What is main?

Main function of program. All functions execute within the main function.

<https://www.youtube.com/results?search_query=why+we+use+void+main>

Configuratiescherm\Gebruikersaccounts\Referentiebeheer

* Windows Referentie

[**https://www.freecodecamp.org/news/how-to-delete-a-git-branch-both-locally-and-remotely/**](https://www.freecodecamp.org/news/how-to-delete-a-git-branch-both-locally-and-remotely/)

[**https://github.com/Kunena/Kunena-Forum/wiki/Create-a-new-branch-with-git-and-manage-branches**](https://github.com/Kunena/Kunena-Forum/wiki/Create-a-new-branch-with-git-and-manage-branches)

[**https://devconnected.com/create-git-branch/**](https://devconnected.com/create-git-branch/)

**Training vindt plaats op de branch**

**Main voeg ik de code toe die staat**

**Eventueel door andere branch te mergen**

Git Bash Commands

pwd = present working directory: Met deze command toon Git Bash in welke directory je zit.

Clear = clear terminal Deze command wist de informatie in de terminal (voorgaande acties niet zichtbaar).

Crtl + l = fresh page in terminal Deze command creeert een schoon scherm in de terminal (voorgaande acties wel zichtbaar).

--help Geeft informatie over het commando wat je ervoor schrijft

Ls = list Deze command toont een lijst de bestanden in de directory waar je je in bevindt

Ls -l = list with details

Ls -a = list with hidden folders and files

Ls -la = list with details about files and hidden folders and files

cd + directory name= change directory (bijv (cd Desktop))

cd .. = Terug naar parent folder

cd = Terug naar basis folder

start . = Het zichtbaar openen van de map waar je je in Git Bash in bevindt.

Cd + directory + foldernaam = change directory direct naar het bestand waar je wilt zijn(bijv(cd Desktop/new-folder))

mkdir – make directory

touch – make file

cat kijken naar inhoud file in terminal

tab = maakt automatisch compleet

rm = remove file

rm -rf = remove folder

Git

Git init – initialiseert nieuwe repository folder

Git heeft een eigen systeem en slaat dus zelf objecten op:

**Git object types:** Dit zijn de type objecten die Git opslaat. In deze 4 objecten zit alle noodzakelijke data.

* Blob: Hier slaat Git files op. Elke file van elke type
* Tree: Hier slaat Git de informatie over de directory op
* Commit: Hier worden de verschillende versies van projecten opgeslagen
* Annotated Tag: Dit is een textpointer (verwijzer) naar een specifieke commit

**Hash Function**

1. Is a one way function (bijv passwords worden opgeslagen als hash.
2. Same hash function will always create same hash for the same input