07-04-2023, CodeFounders

<https://www.w3schools.com/>

HTML

HyperTextMarkupLanguage

Manier om snel tekt over te dragen, taal om aan te geven hoe het eruit gaat zien. Taal waarmee je met de browser communiceert

Als je “html:5” intikt dan geeft visual studio code een voorbeeld van je hele html code.

Je moet een file opslaan met de goede extensie erachter. Dus in dit geval ‘voorbeeld.html’ anders ziet de computer niet dat het een html bestand is.

HTML bestaat uit 2 gedeeltes:

* Head 🡪 meta informatie (info over die pagina: titel, welke javascript, social media tekst)
* Body 🡪 data (wat je daadwerkelijk ziet)

HTML bestaat uit elements (onderdelen)

Element: begint met <html> …… </html>

<html>

<head>

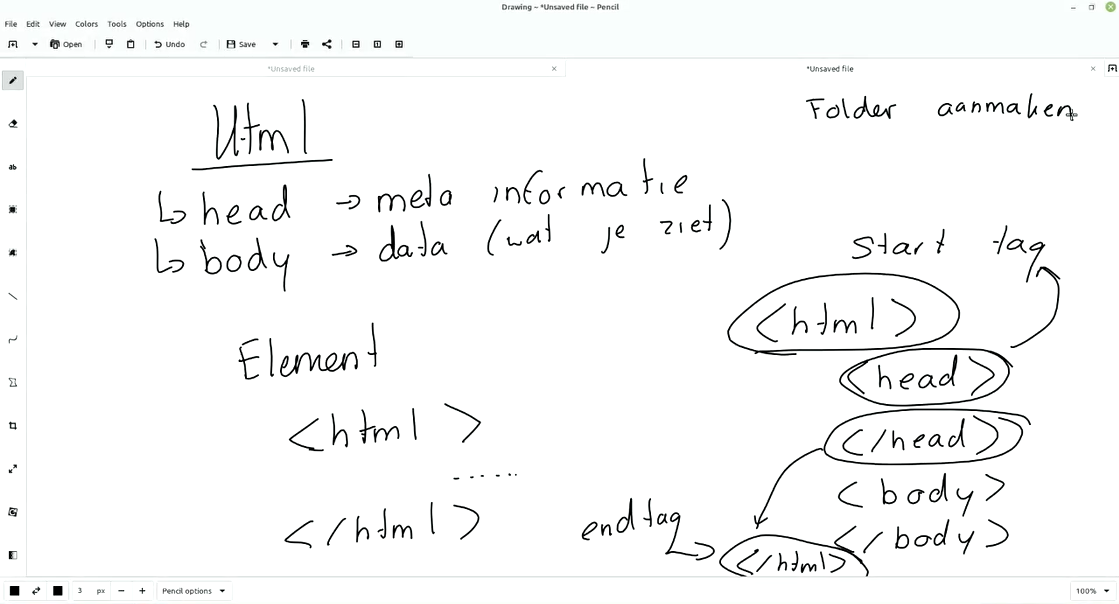
</head>

<body>

</body>

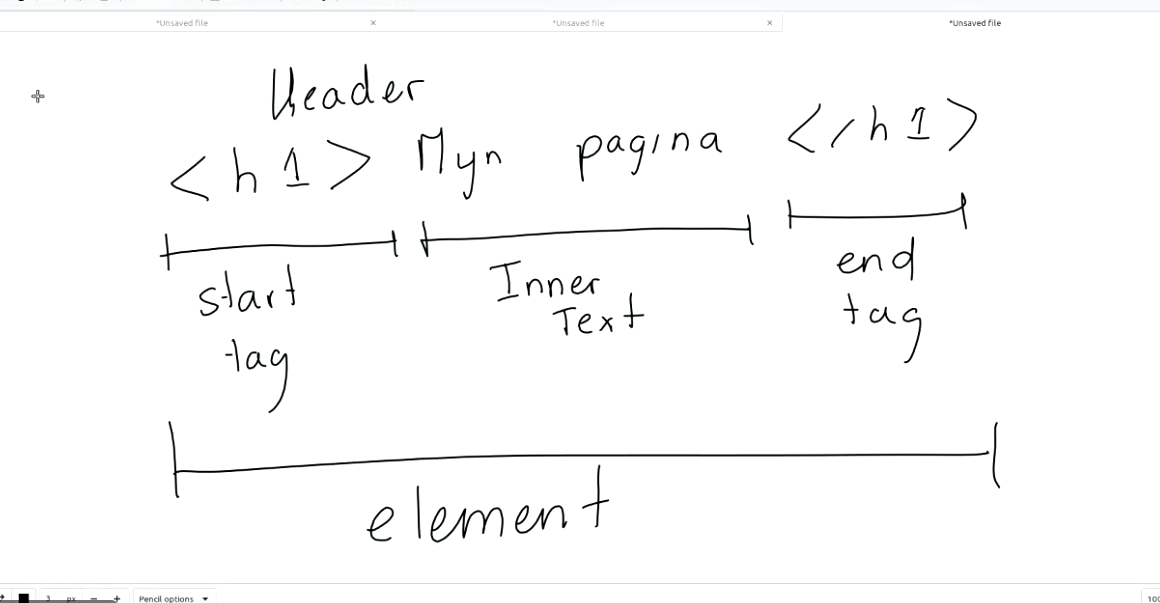
</html>

Je begint met een ‘start tag’ <head> en een end tag </html>



Stel je wilt een ‘Header’ op je pagina

<h1> Mijn pagina </h1>



Span = normale tekt

P = paragraaf = tekst met iets meer ruimte

Je moet eerst een bestand opslaan in visual … voordat je het kan zien in de browser.

F12 (Fn ook indrukken) in browser geeft webdeveloper om pagina aan te passen.

Alle aanpassingen die je hierin doet zijn direct actief. Je gebruikt dit meer om te testen. Want als je hier een aanpassing doet dan komt het niet in visual studio code.

Je kan ook een ‘table’ maken.

<table>

<tr>

<td> Name </td)

</tr>

</table>

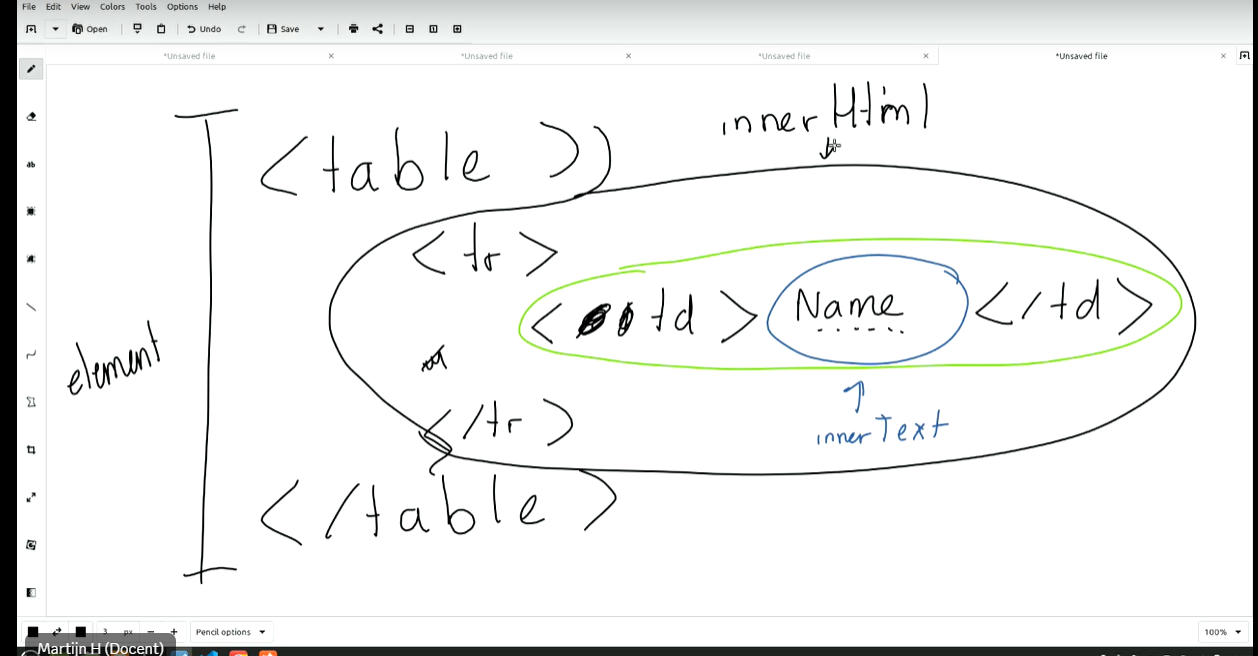
Tr=table row

Td= table data

Je kan oneindig veel table rows hebben

Table heeft geen inner tekst. Wel inner HTML

De table data heeft wél inner tekst



Afbeeldingen

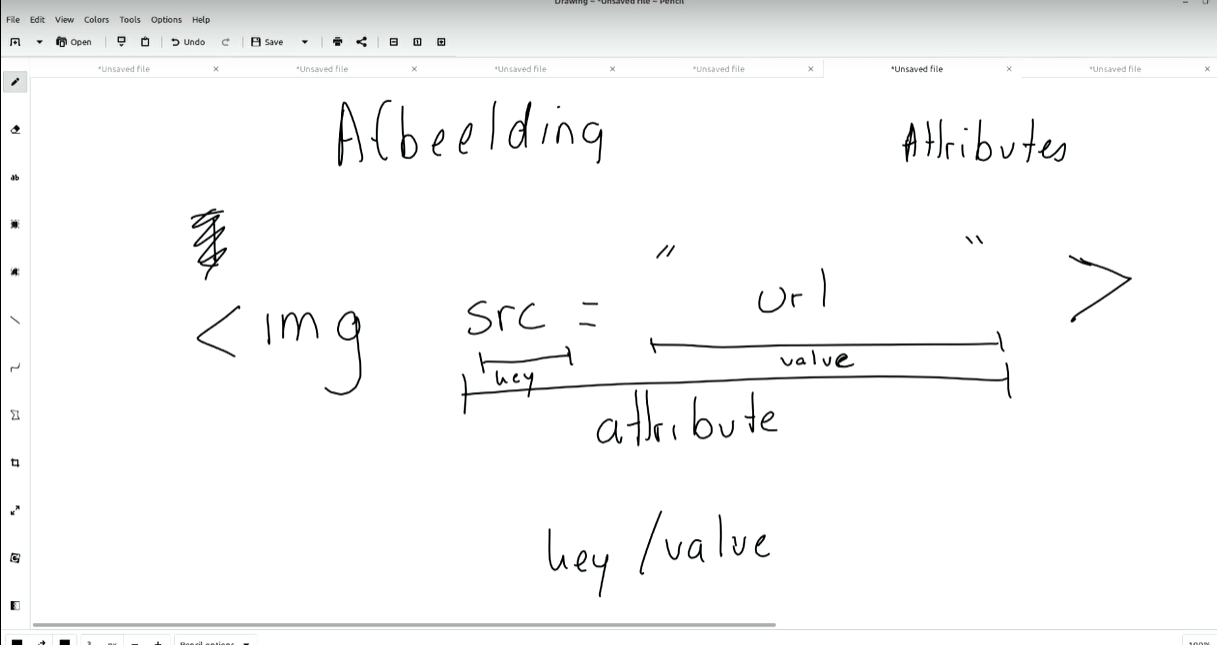
Element ‘Image’

<img

Je moet opgeven welke image je wilt weergeven. Dit doe je met eigenschappen ‘attributes’. Hiervoor gebruik je ‘src’ (source). Waar moet die afbeelding vandaan komen.

Er zijn meerdere soorten attributes, afhankelijk wat je weergeeft. Video heeft andere attributes als afbeelding.

De img tag heeft geen endtag niet nodig.



Opdracht: maak een table met 2 rijen

* Eerste rij plaats je een image
* Tweede rij plaats je tekst

HTML: wat we willen weergeven

CSS: hoe wordt het weergegeven

**CSS**

Cascading style sheets

Styling van een pagina.

4 verschillende manieren om te stylen.

* Inline styling
* Class selector
* Tag selector
* ID selector

*Inline styling*

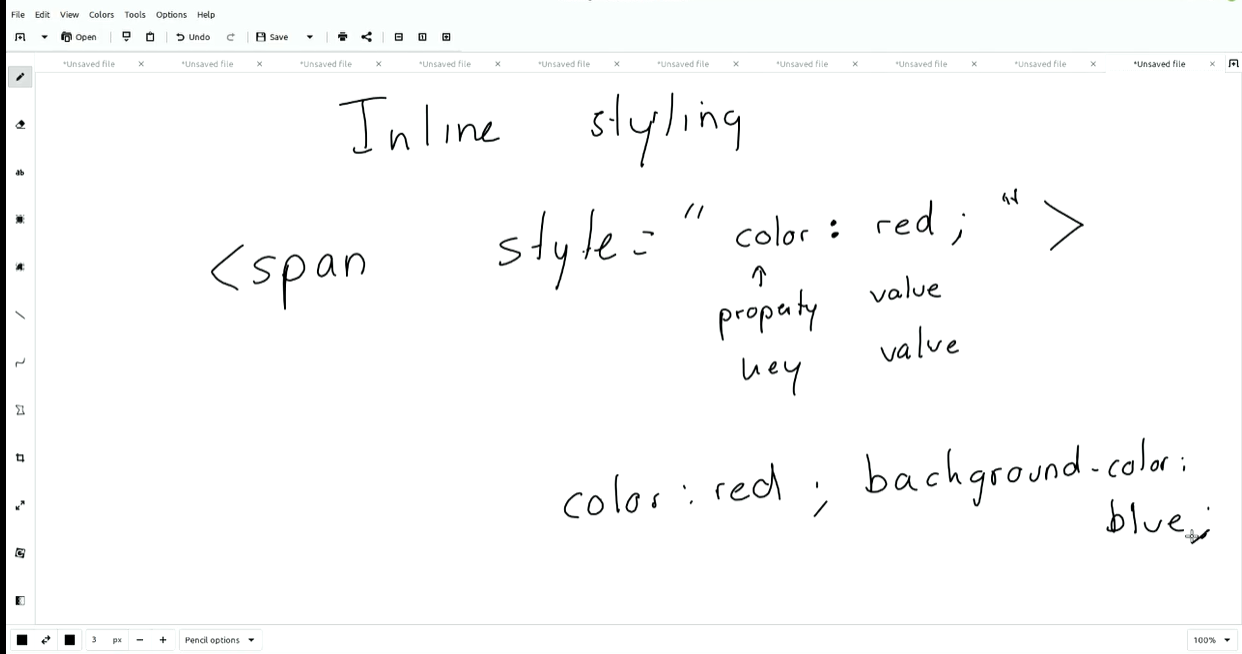
Wordt soms nog wel gebruikt in de praktijk, maar voorkeur gaat naar de andere 3 manieren

Je kan de span een style attribute meegeven. Hier kan je opgeven watvoor color die heeft.

<span style=”color:red;”>

Je kan ook meerdere stylen hierin kwijt

<span style=”color:red;background-color:blue;”>



Opdracht 2

* Geef de paragraaf een kleur
* Onderstreep de paragraaf

Zie index.html

S.O.C. = seperation of concern

Opsplitsen van verantwoordelijkheden

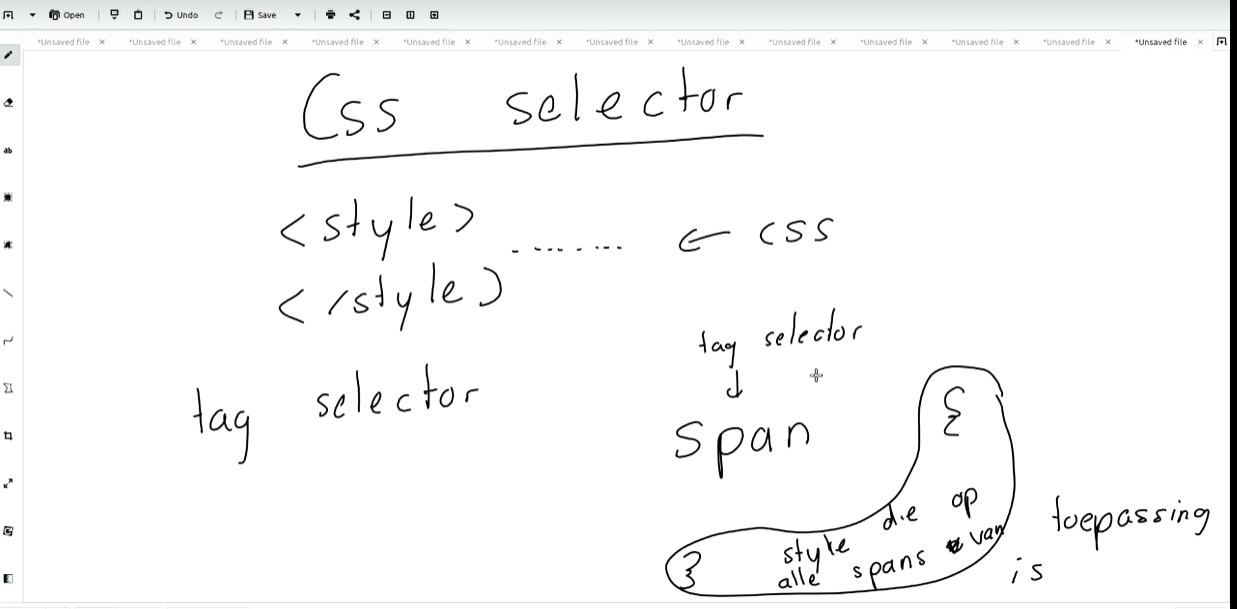
Je wilt eigenlijk alle verantwoordelijkheden verdelen. Anders wordt het één grote brei. Duidelijke scheiding

**CSS selector**

In de header van de pagina gaan we een style element toevoegen. Alles wat hierin zit wordt gezien als CSS. We moeten dan ook telkens aangeven welk element we willen stylen. Hiervoor gebruik je een **selector.**

* Tag selector
  + Span { }

Inline styling overschrijft selector want die is specifieker



Class selector

Gebruik je wanneer een label wilt instellen, bijvoorbeeld ‘achtergrond’. Dan krijgen alle elements een rode achtergrond

Dit komt in de body: <p class=”class1 class2”

De class is spatie separated

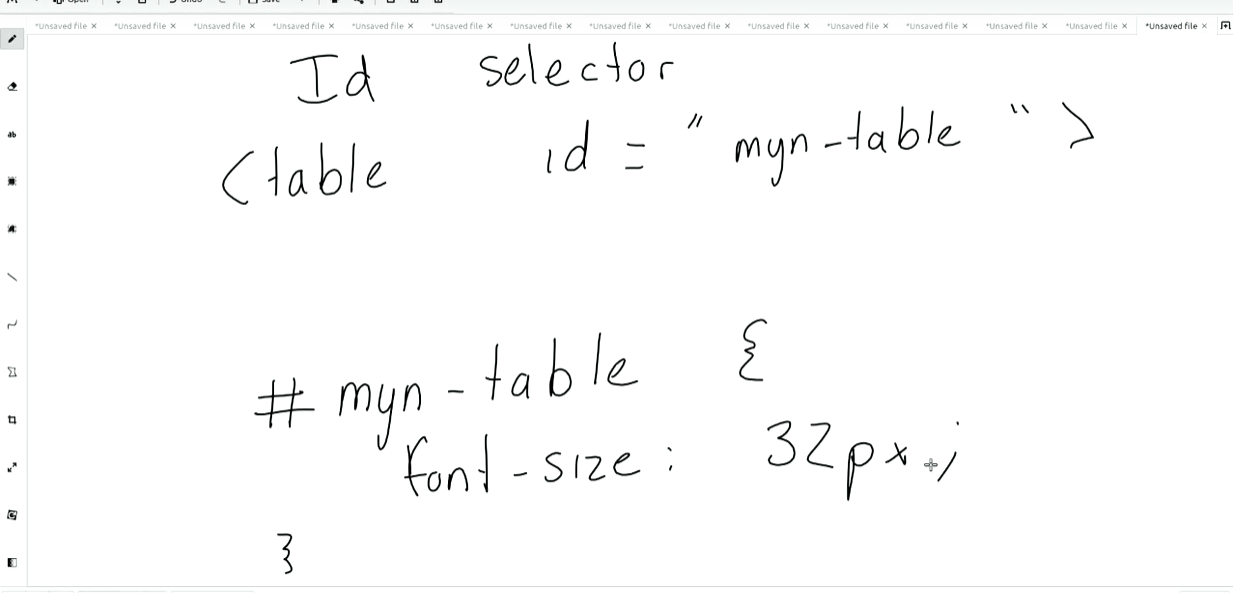
Dit komt in de header

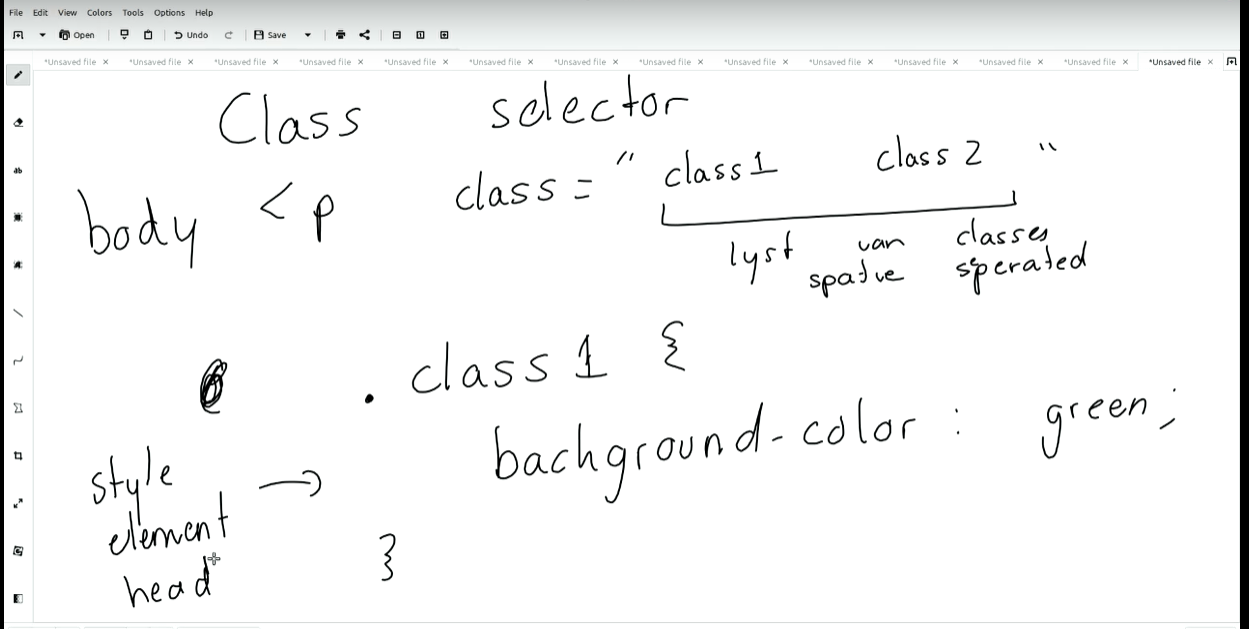
.class{ }

ID selector

(table id=”my-table”)

#my-table { font-size: 32px;}





Opdracht 3

Bedenk:

* 2 class selectors
* 2 tag selectors
* 2 id selectors

Per selector 2 properties aanpassen

In style gaan staan en ctrl + spatie. Geeft heel veel opties om te selecteren

**Javascript**

Java heeft niks met javascript te maken

Javascript is gemaakt om in de browser te draaien.

Gaat over de functionaliteit. Maakt je pagina dynamisch.

Script tag:

<script>

javascript

</script>

Javascript wordt van boven naar beneden uitgevoerd.

Je kan iets naar de console sturen 🡪 console.log(“het werkt!”)

Binnen javascript hbe je een aantal mogelijkheden

Variables = tijdelijke data opslag

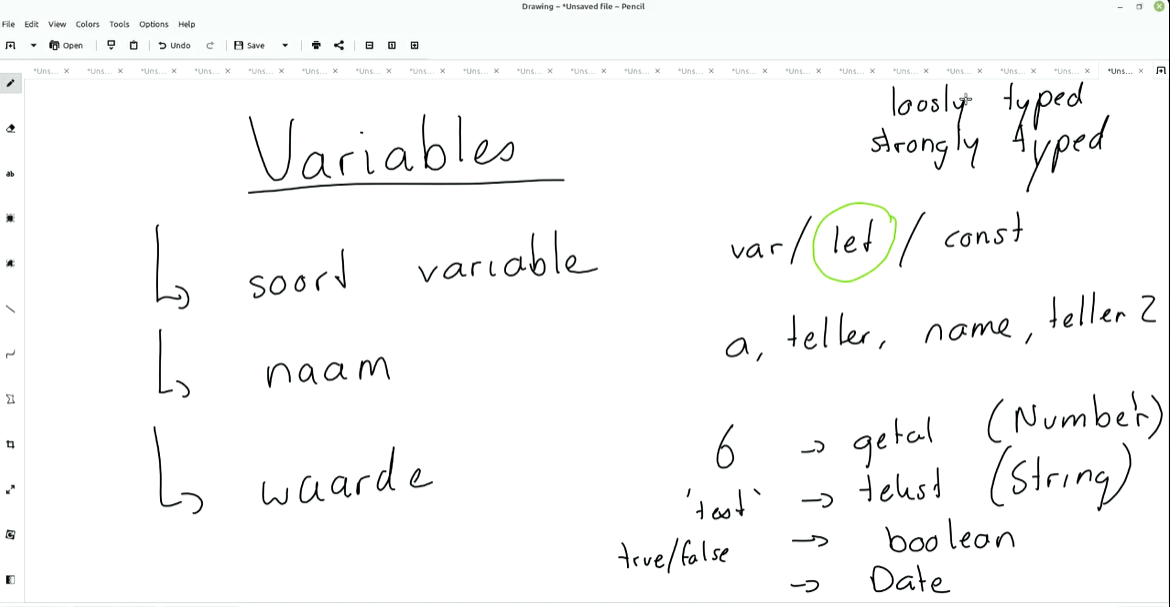
Functions = code block met een naam

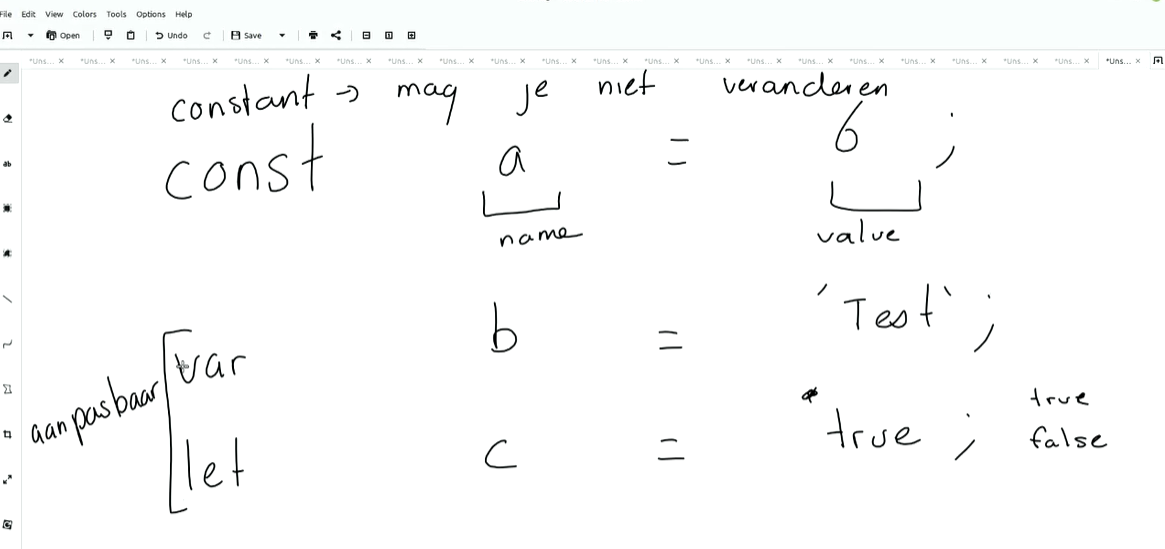
Variables

3 kenmerken:

* Soort: var/let/const
  + Const = constant
  + Verandert de waarde: dan var/let
* Naam: mag alles zijn
* Waarde: number (getal), string (tekst), boolean (true/false), date

Variable is loosly typed: betekent dat je het kan veranderen





Dubbele forward slash: wordt als comment gezien.

De “;” is niet noodzakelijk bij Javascript. Als iets op een nieuwe regel staat dan gaat het ervan uit dat het een nieuwe command is.

Verschil var en let = let kan niet opnieuw gedefinieerd worden. Var wel. Bij var wordt de vorige var overschreven.

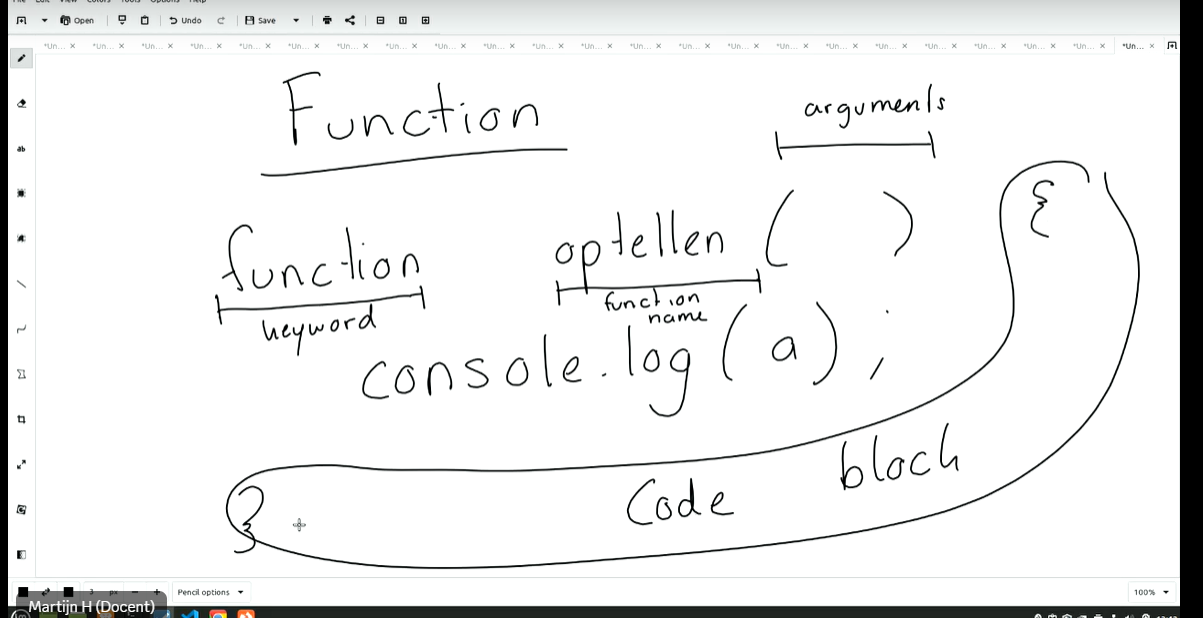
Function

Function optellen(….) {

Console.log(a);

}

Keyword: mag nooit een variable zijn, geeft aan wat de function wordt.



Optellen(getal1, getal2)

Getal1 en getal2 zijn de argumenten

Opdracht 3

Rekenmachine maken

* Maak het mogelijk om te +-\*/