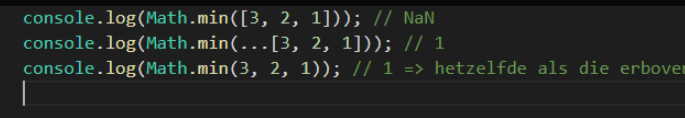
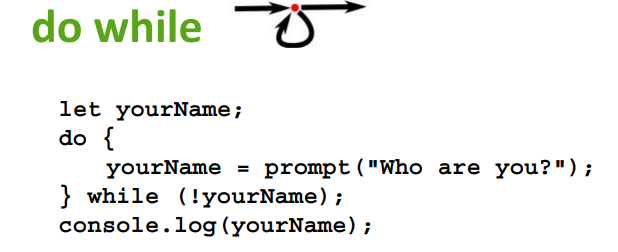
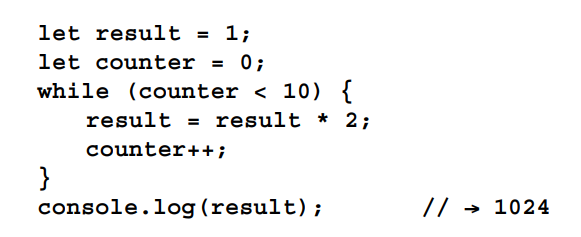
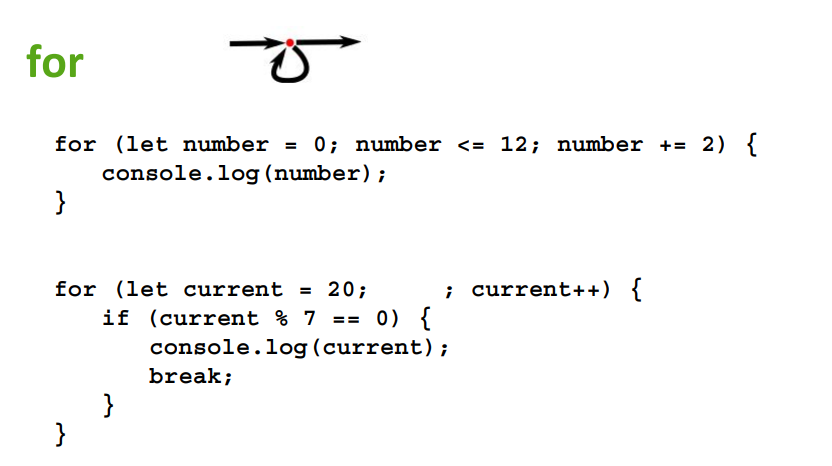
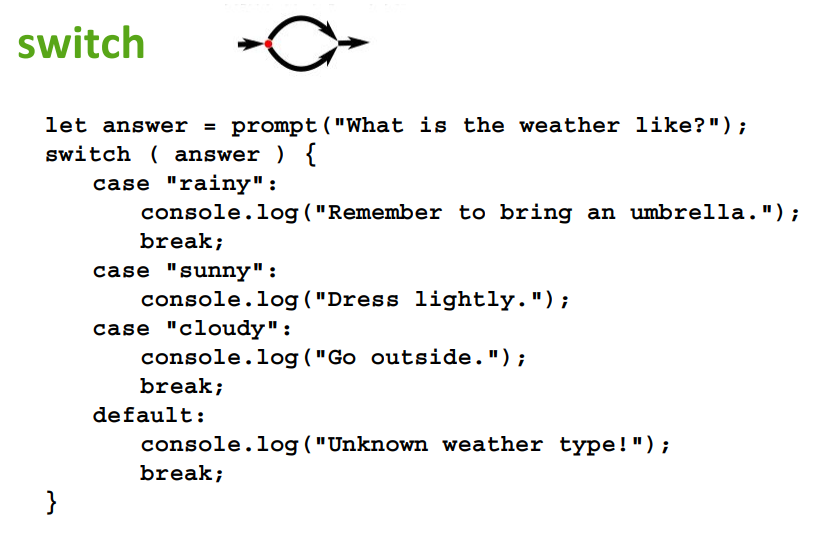
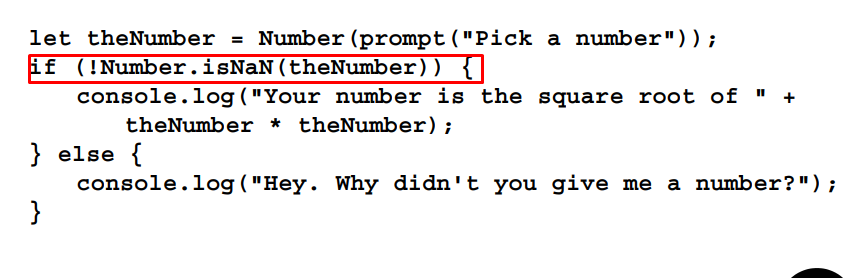
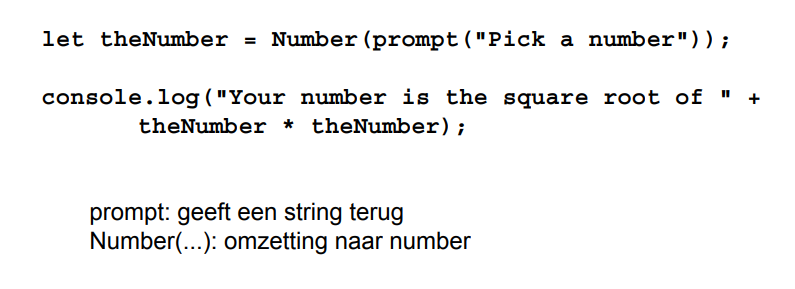
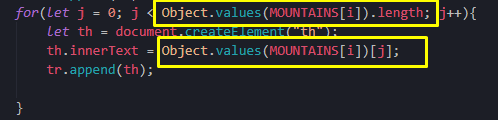
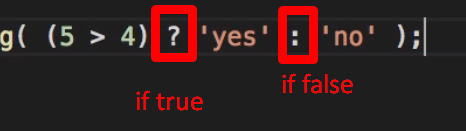
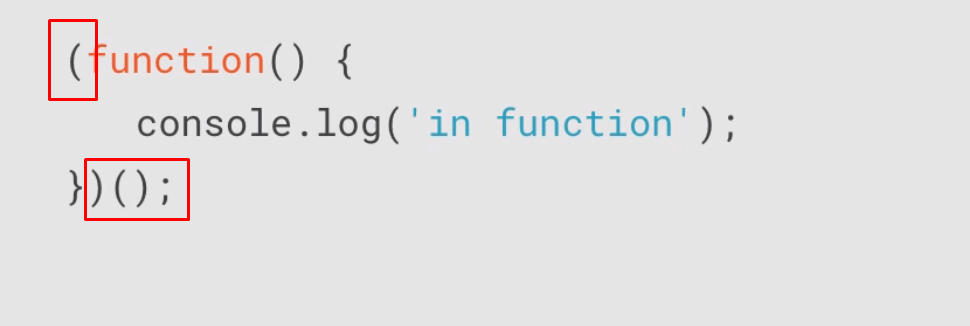
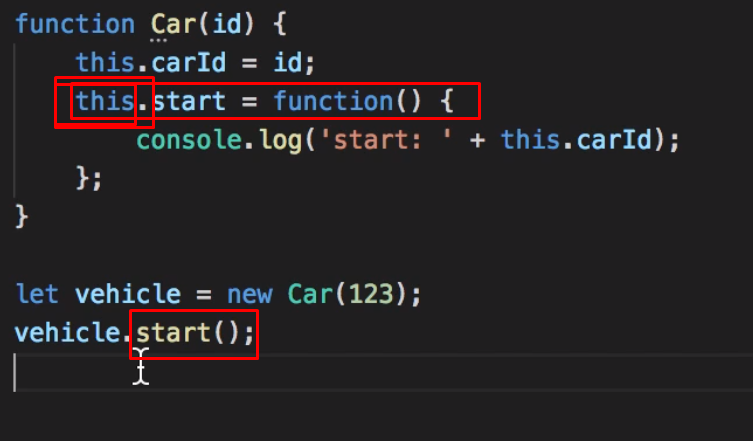
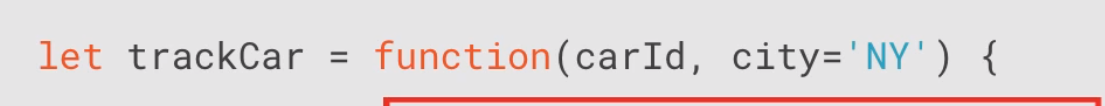
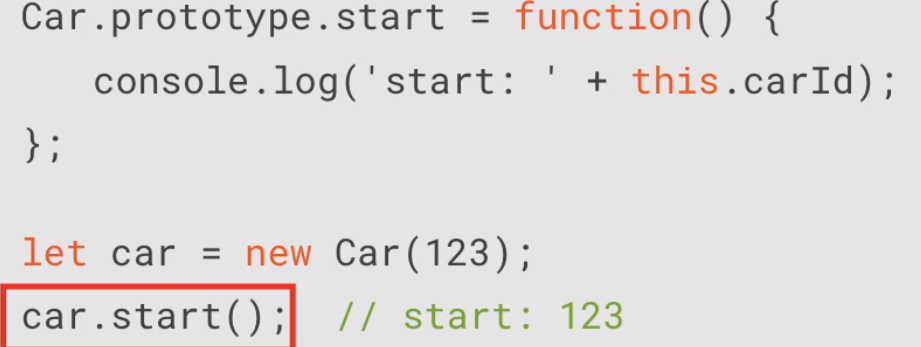
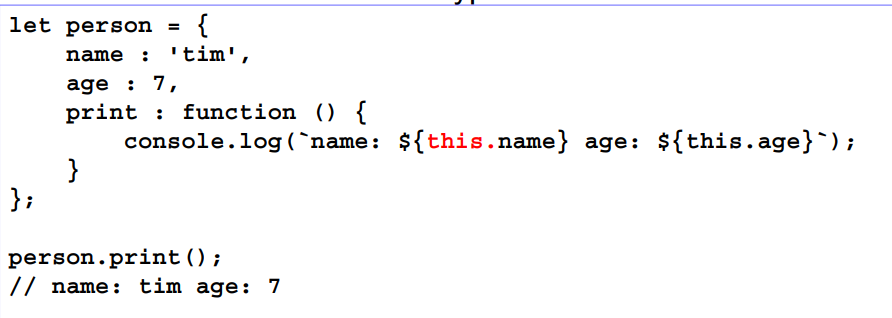
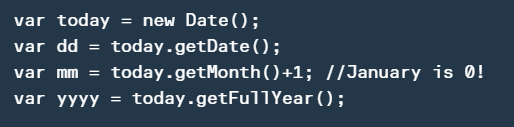
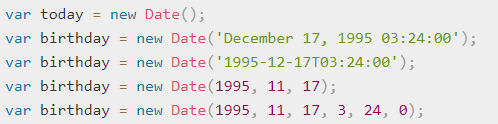
Webscripting

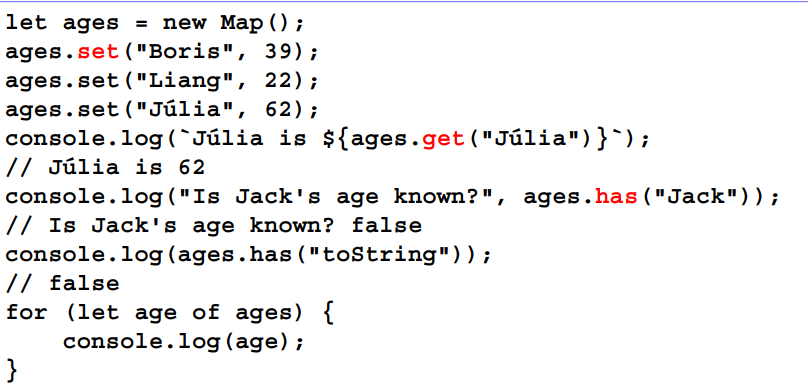
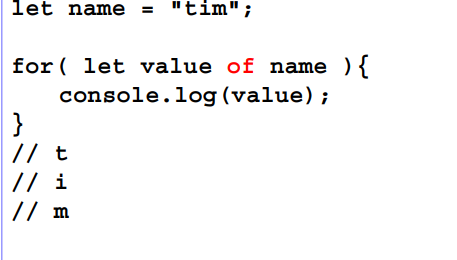
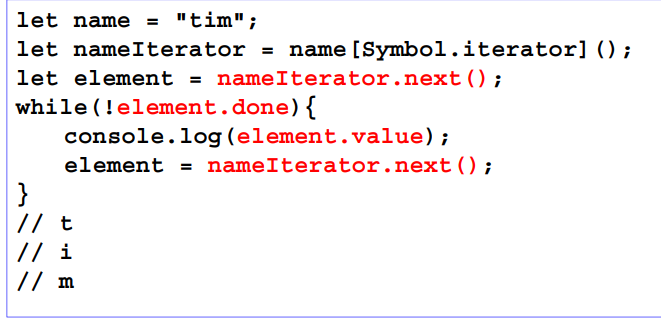
process.stdout.write = op zelfde lijn  
console.log = nieuwe lijn  


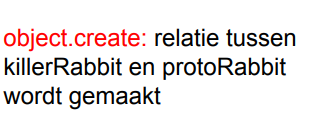
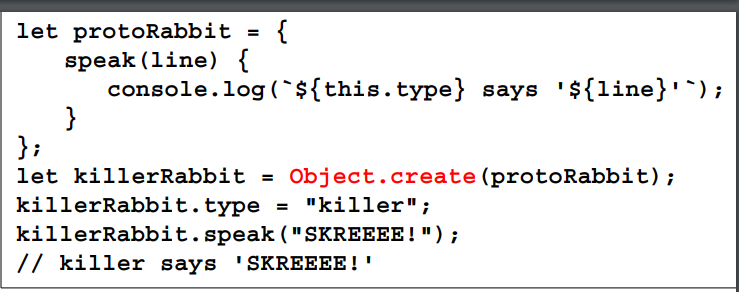
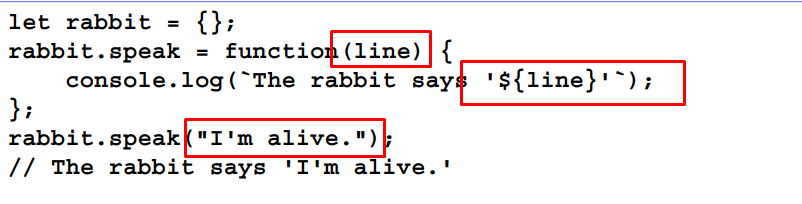
**Object values tellen in list!**  


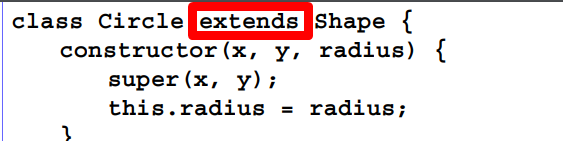
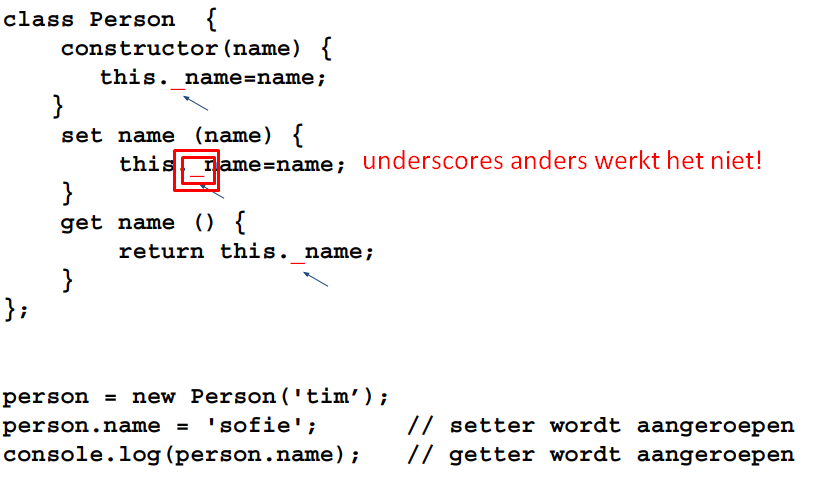
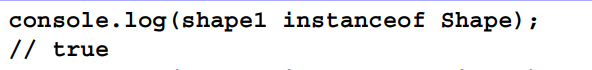
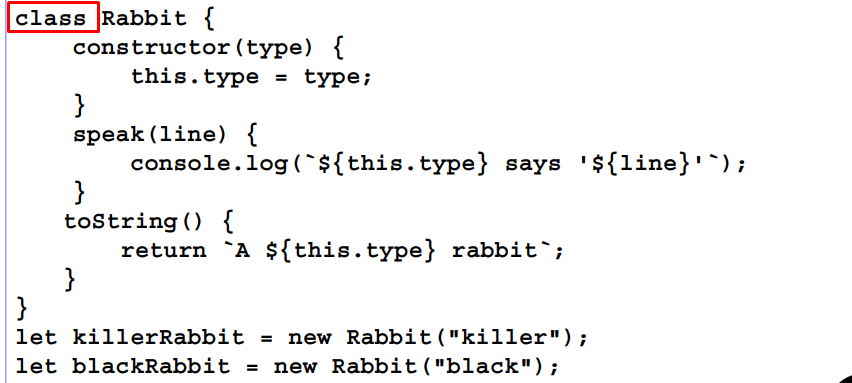
== probeert type conversion te doen dus 11 == yeet, zal 11 omzetten naar string  
=== doet dit niet, dus de types moeten hetzelfde zijn  
!=  
!== zelfde principe dus KEEP IN MIND QUEEN  
|| OR  
&& AND wordt altijd eerst gecheckt dus, eerst or checken? zo:  
**hoofdletters zijn kleiner dan gwne letters in ascii (we kijken dus of z eerder dan a komt, maar los dit op met touppercase) dan komt ofc alphabet eerst**  
**5 groter dan 4, if true print yes, else no**

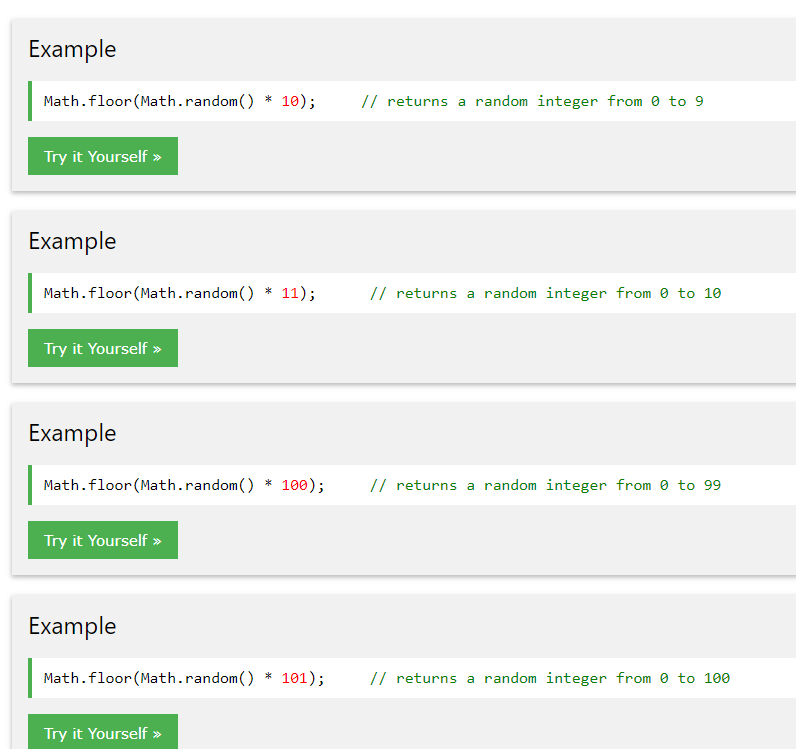
**FUNCTIE METEEN UITVOEREN^^^^^^**wat tussen de () staat is ofc meegegeven param  
**wanneer er geen city wordt meegegeven, wordt deze DEFAULT par gebruikt!!**  
**overriding: String.prototype oproepen met override naam**  
**THIIIIIIIIIIS**

**datums^^**





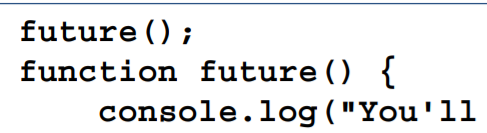
**Classes**   
ja die ${this} zever is tussen backticks, dus hier copy t ma ``````



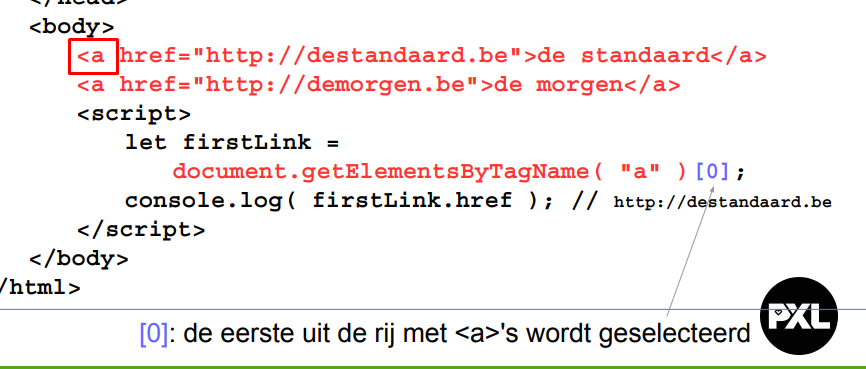
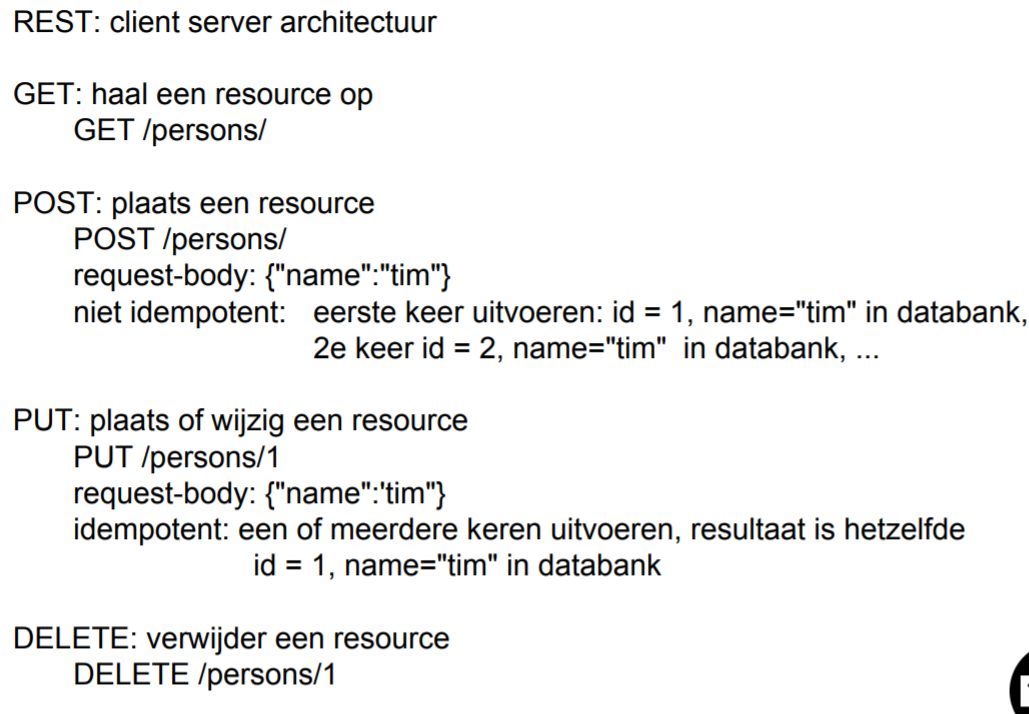
Ternary if  
Let age = 19;  
Let usertype = age < 18 ? ‘minor’ : ‘adult’;  
If age < 18, maak van usertype minor, als dat niet zo is maak er adult van (gwn een fancy if)

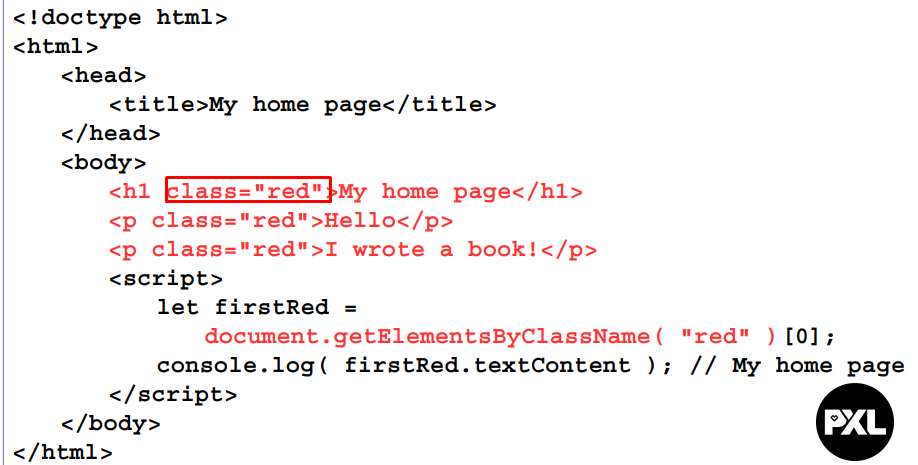
False == 0 –> true -- lijkt enigsinds op 0  
false === 0 –> false -- is exact hetzelfde 0

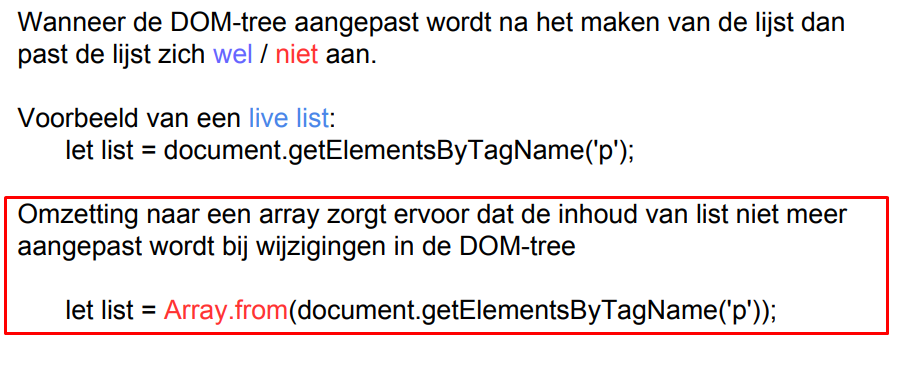
Function assignname( name ){  
 return name || “unknown” geeft name terug wanneer name truthy is, anders unknown

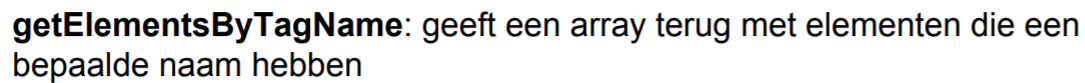
Constante variabalen – const  
gewone variabalen – Let  
Of niets declareren maar functie aanmaken

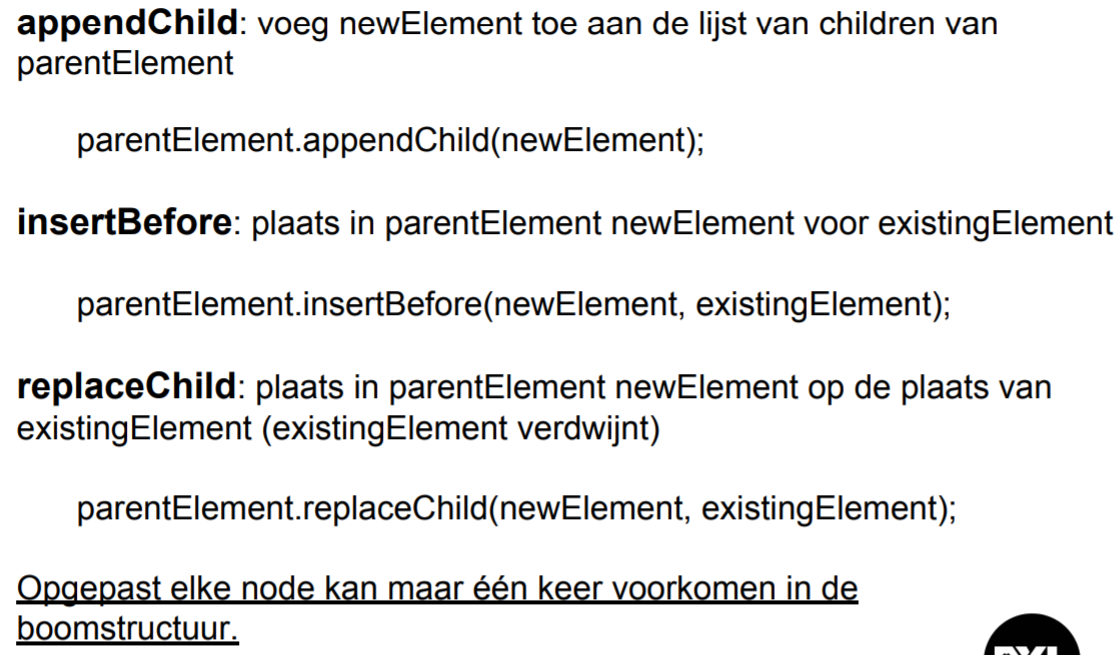
GENESTE CONST

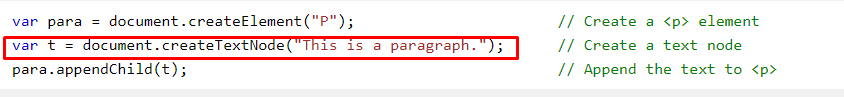
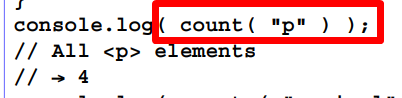
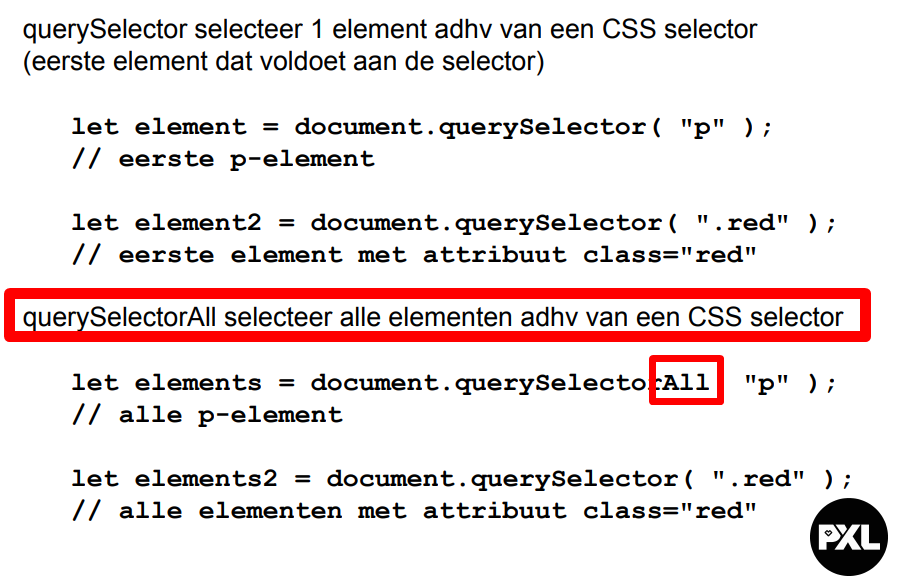
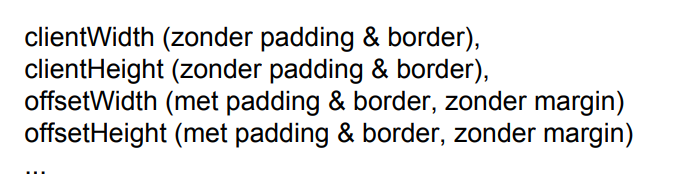
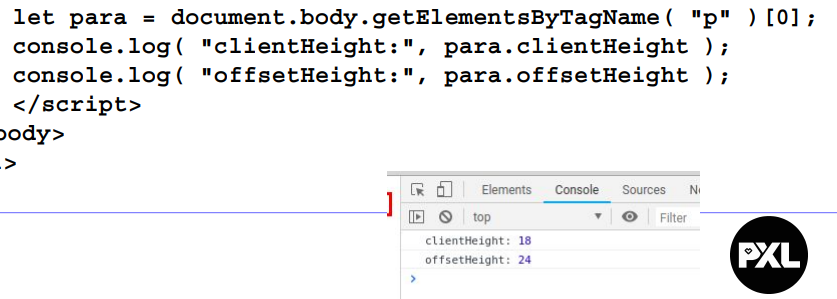




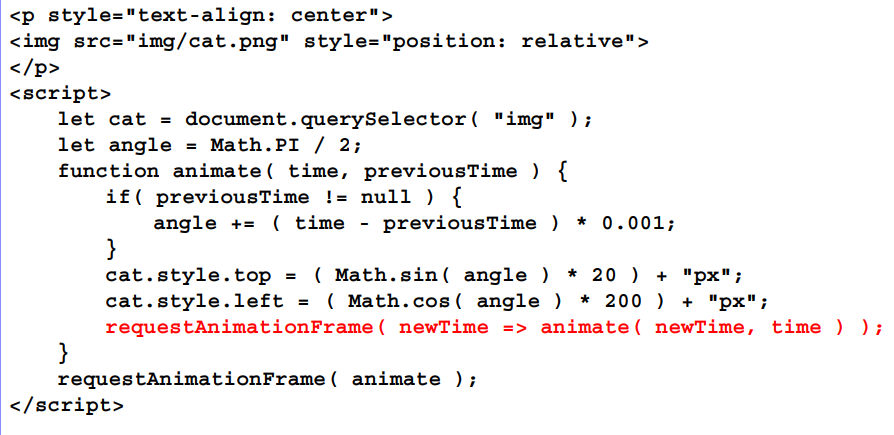
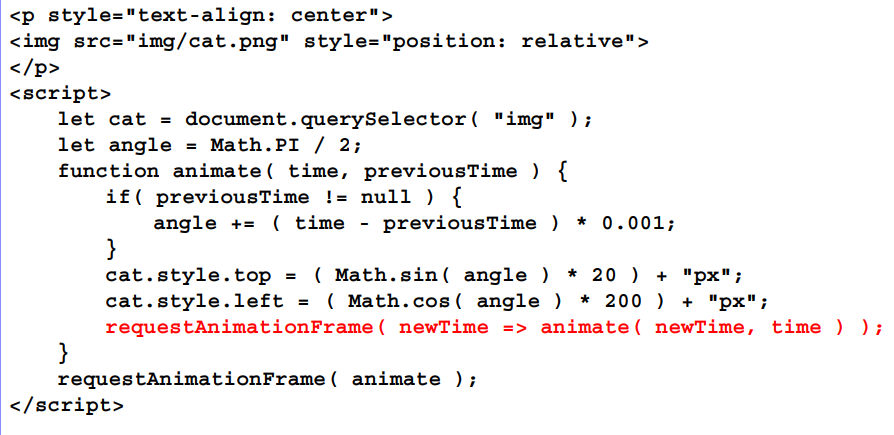




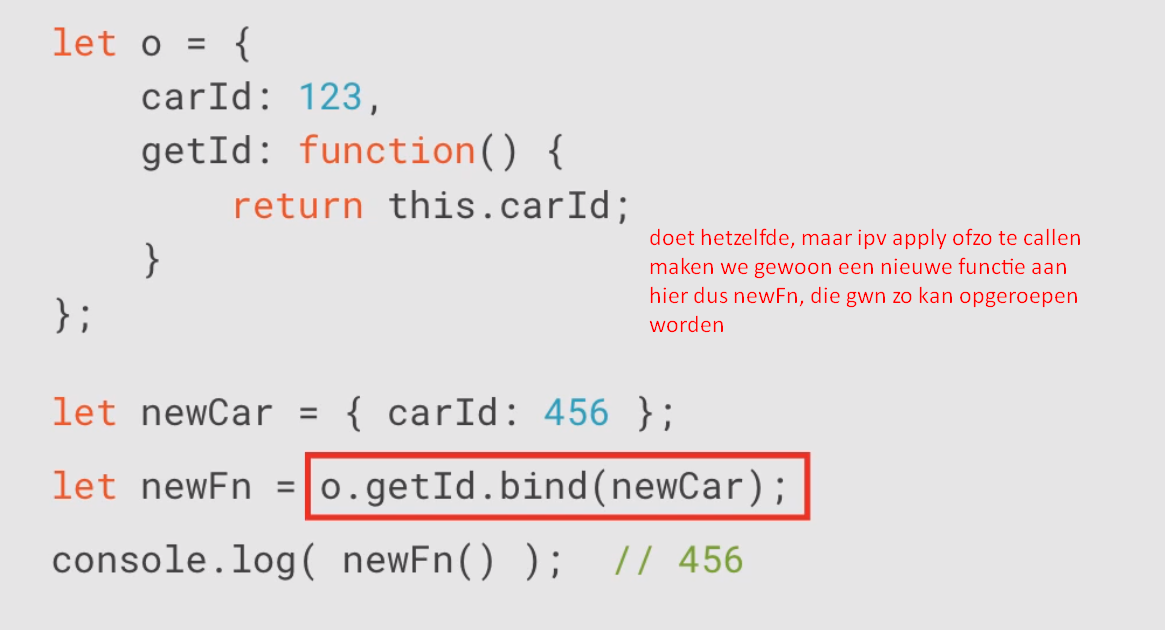


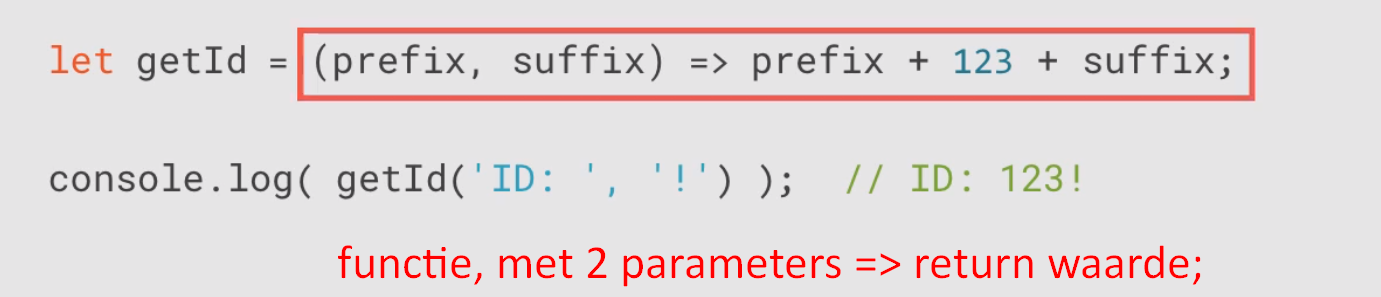
**Gebruik hier dan Let he, Anders Gilles boos**  
  


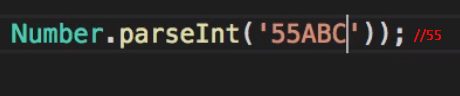
Om waardes terug te krijgen zoals hoogte van een element ofzo

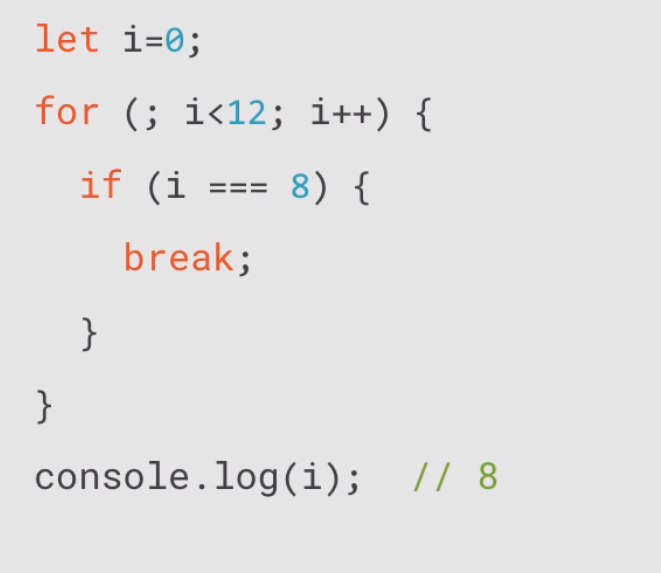
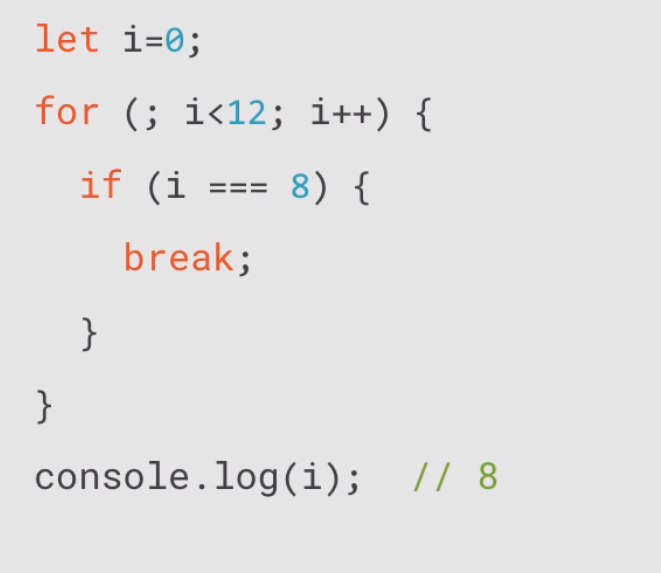
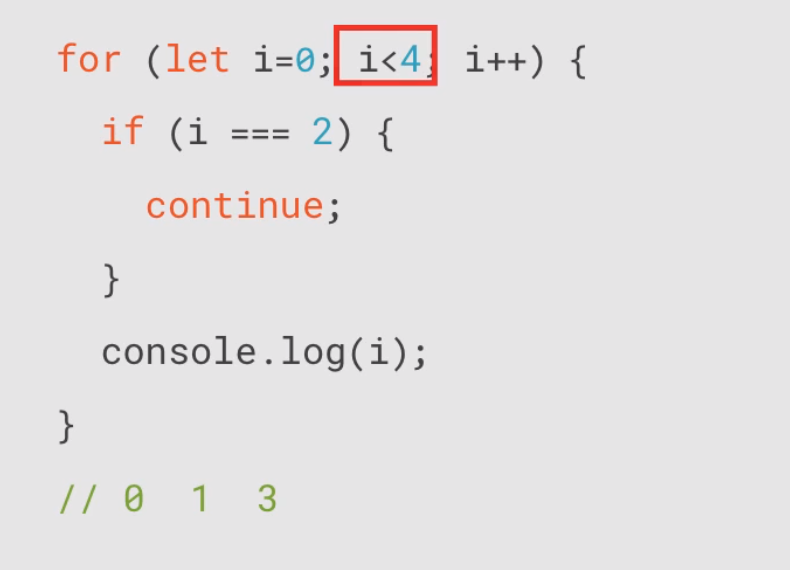
**Animaties idk either, just copy it**  
**Call/Apply**

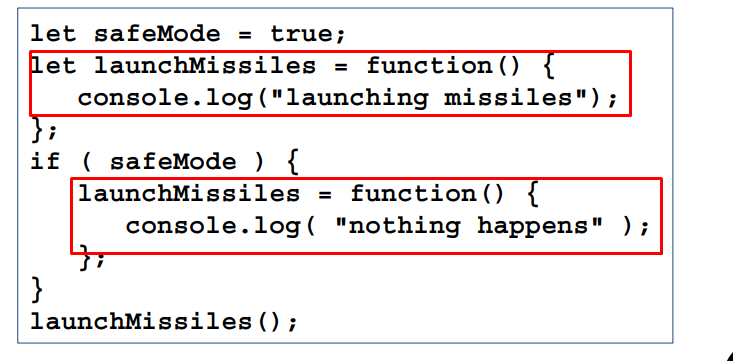


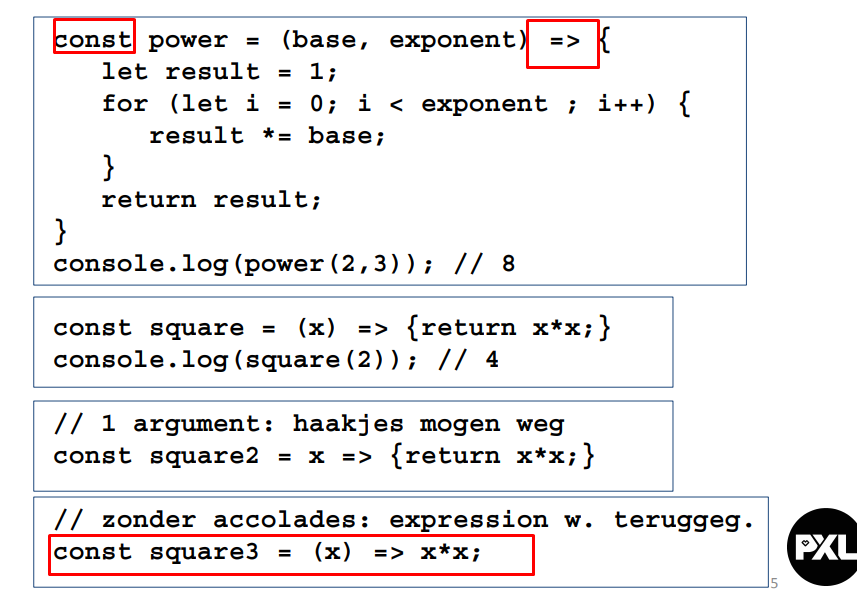
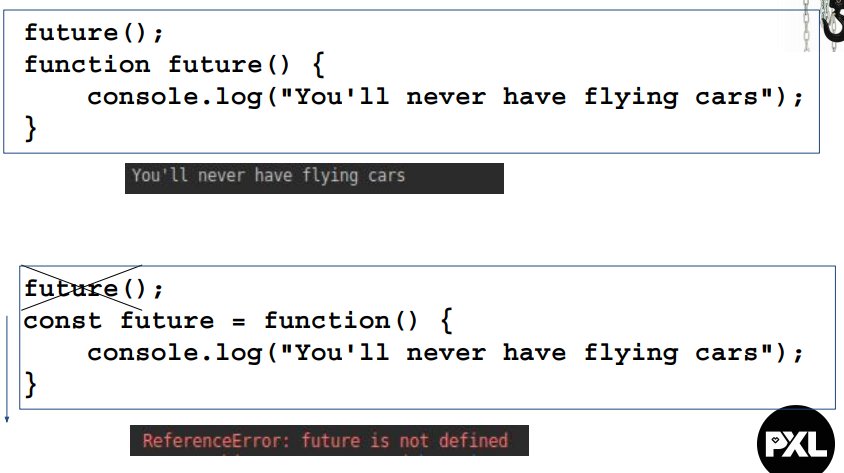


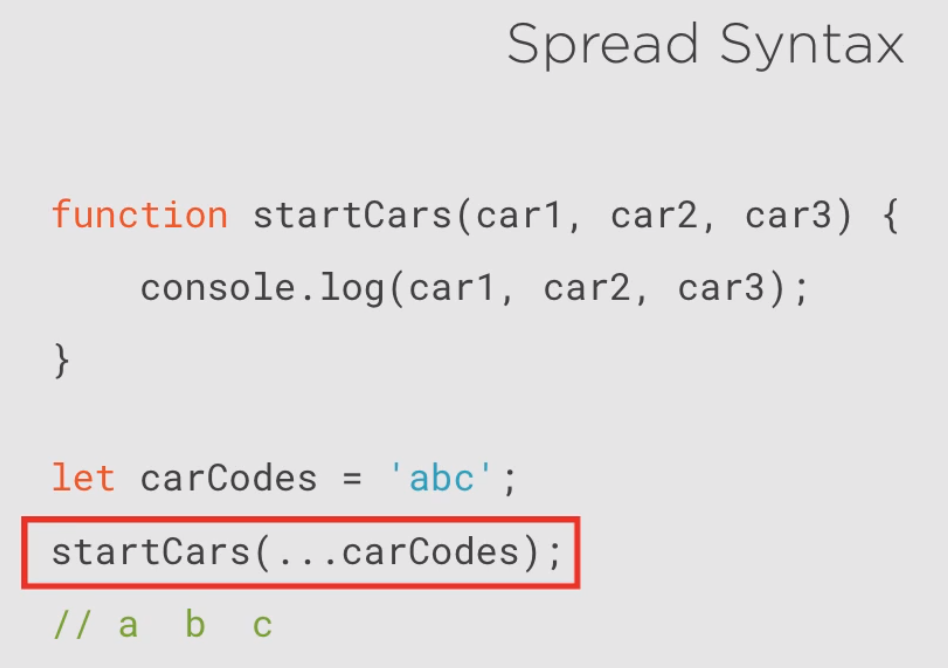
**Verkorte functies**  
**er bestaan geen THIS in arrow functions, we kunnen ipv () ook \_**

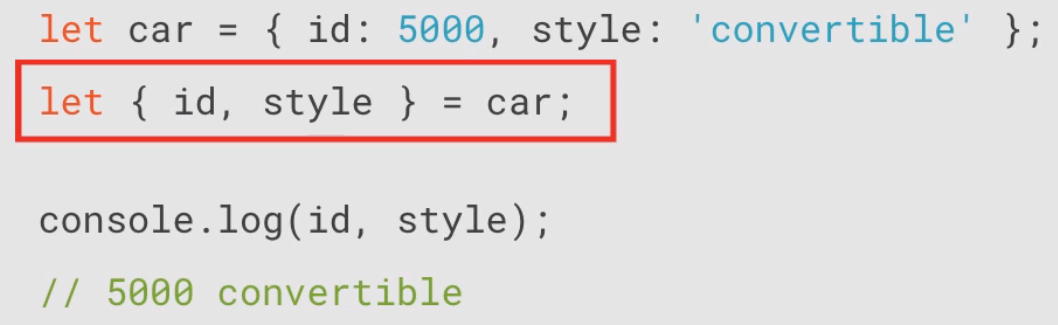
**Conversies**  


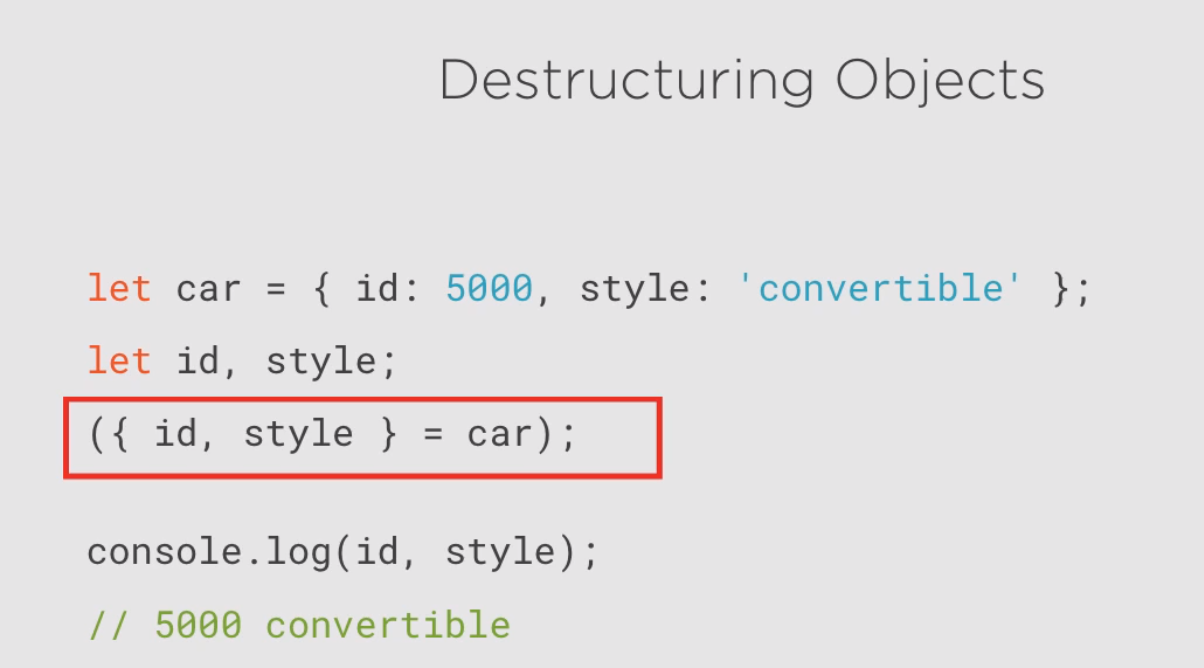
**Break/continue**  
initialiseren i voor de for, wanneer i gelijk is aan 8, stopt de for loop (break)  
wanneer i gelijk aan 2 is, skip de rest van het gehele forblok en doe meteen i++, hierdoor wordt console.log dus niet uitgevoerd

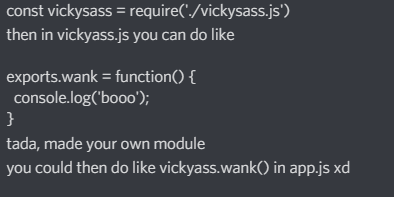


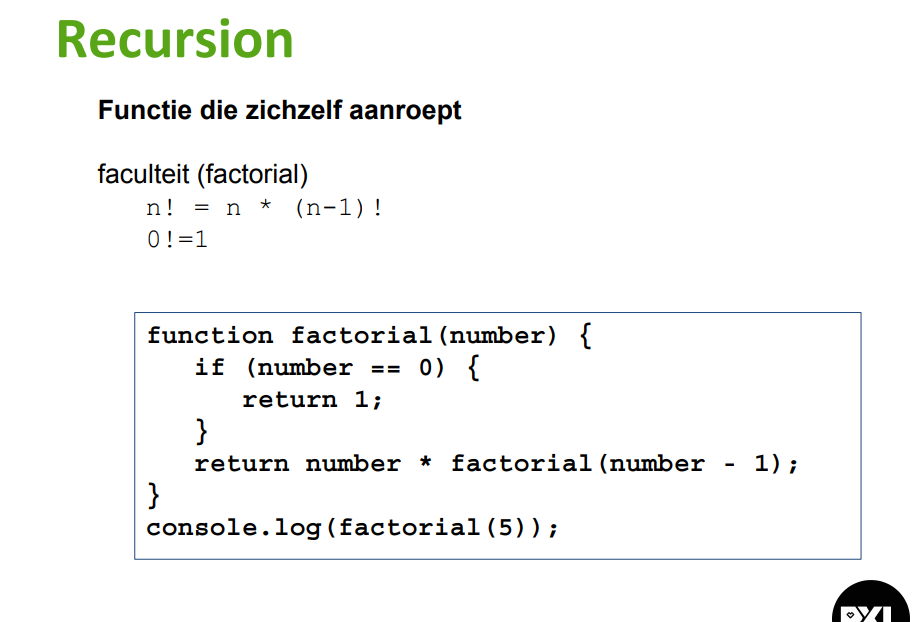
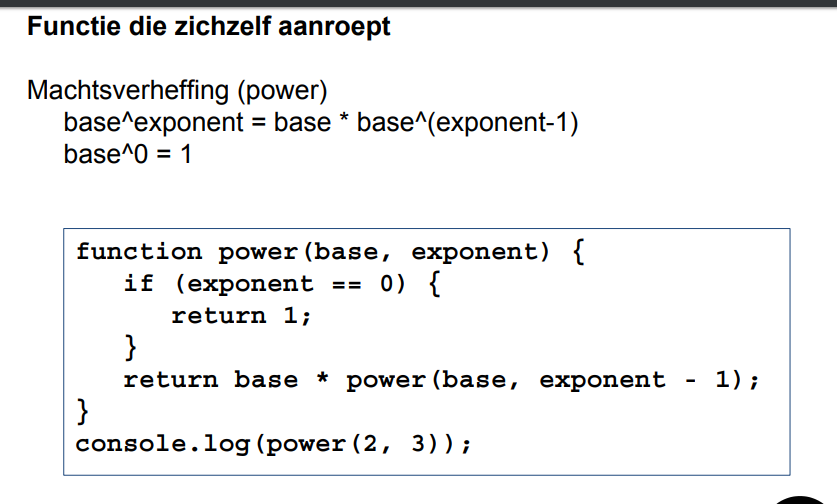


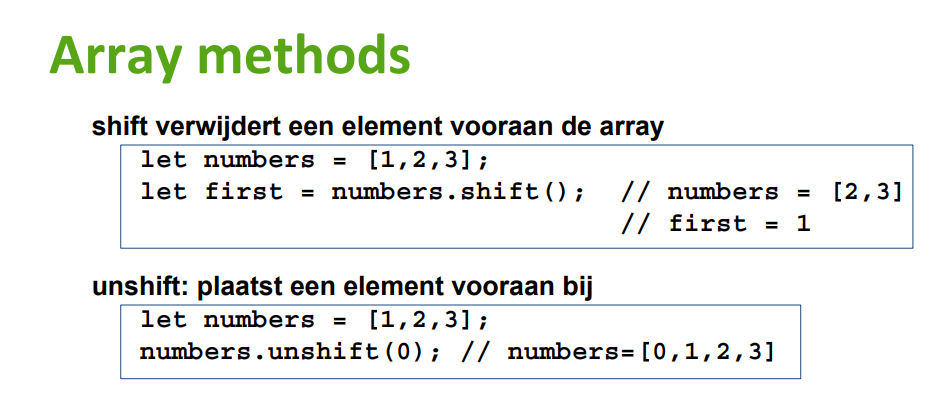
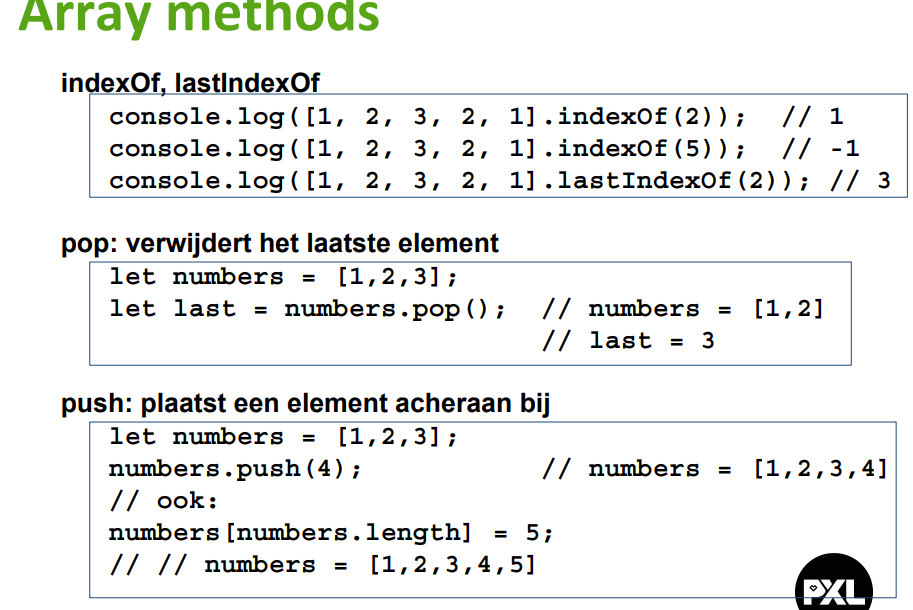
functie met 3 parameters, we spreken deze aan met de rest ...carCodes, deze splitst de string op en plaatst elk deel in de variabale

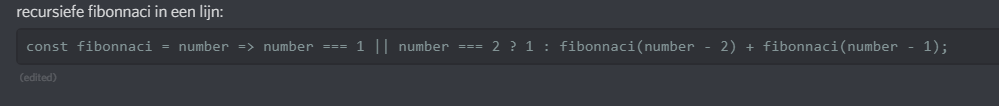
or

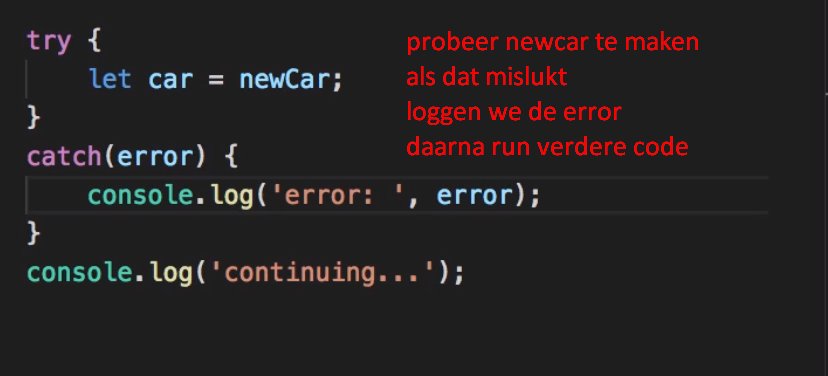
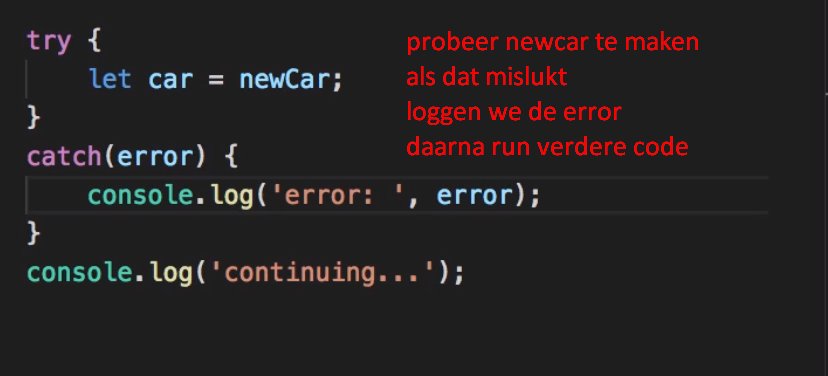
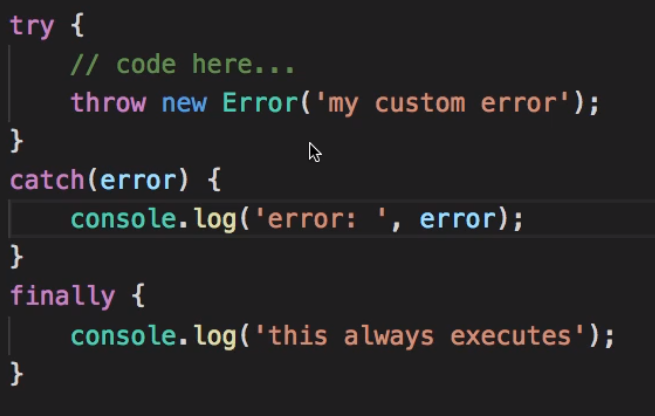
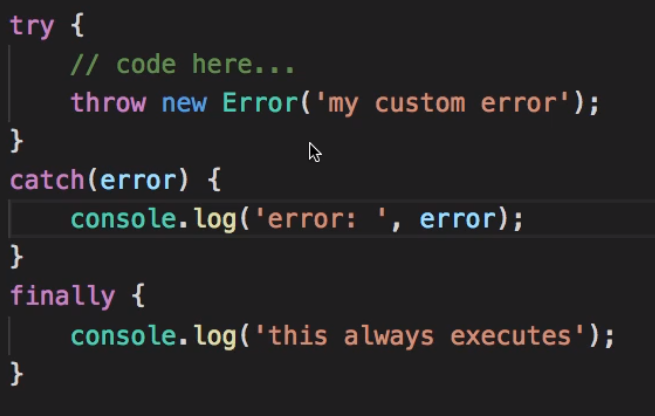
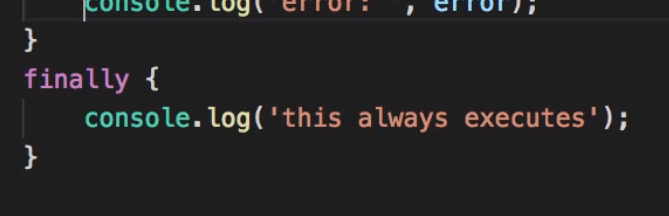
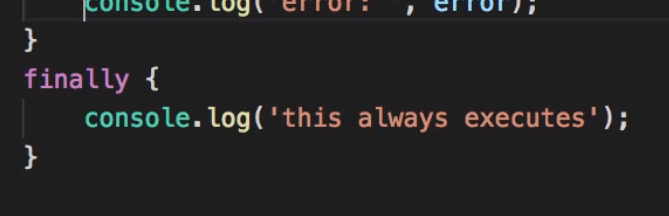


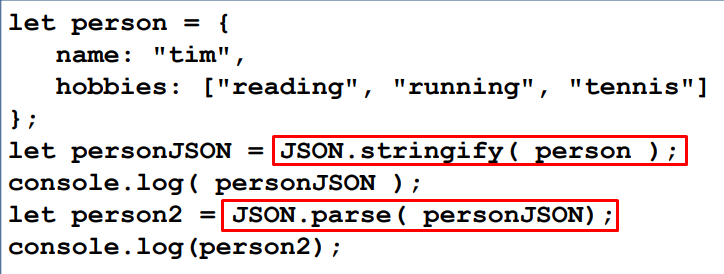
**Modules**  


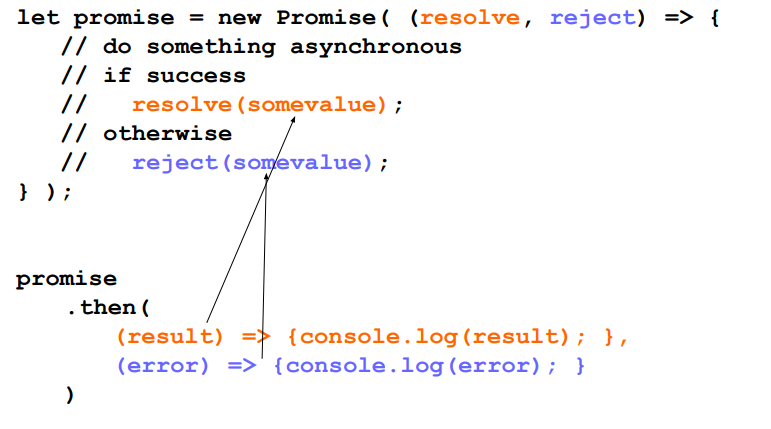
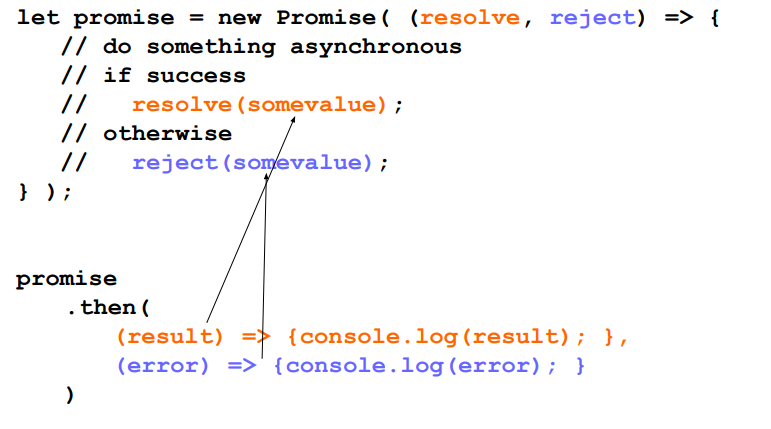
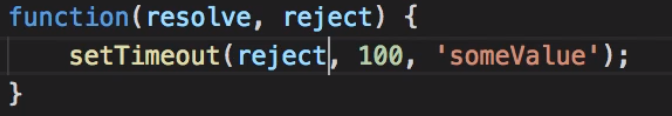
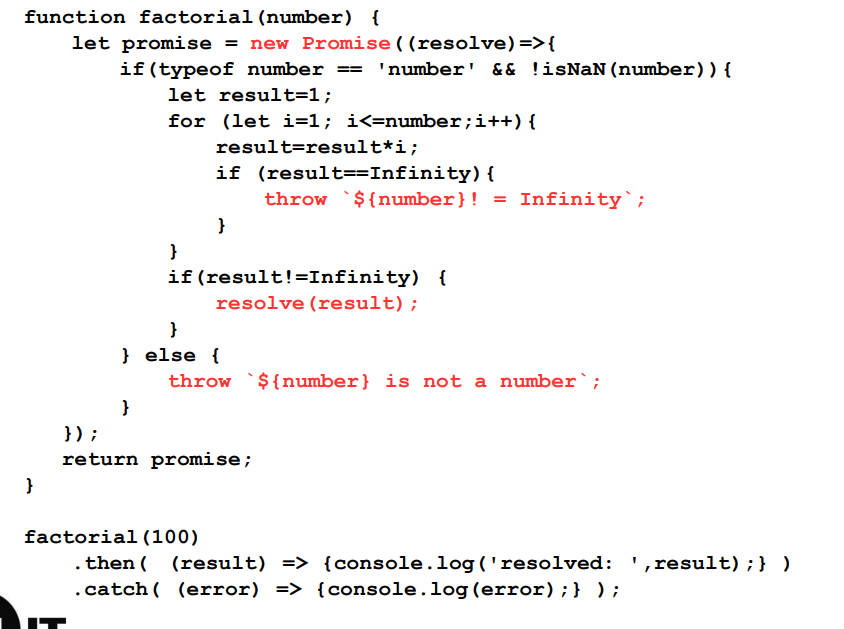
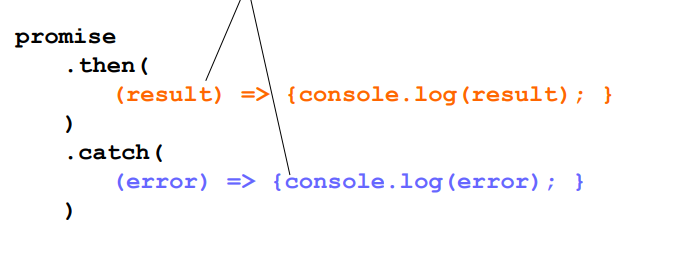


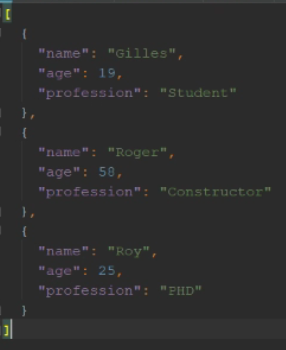


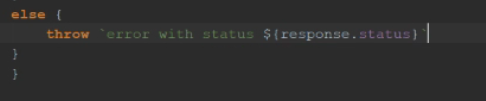
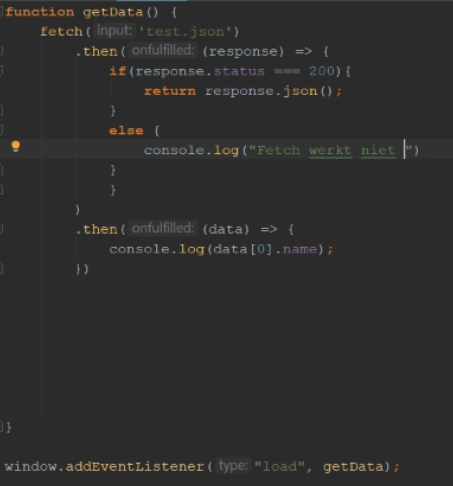
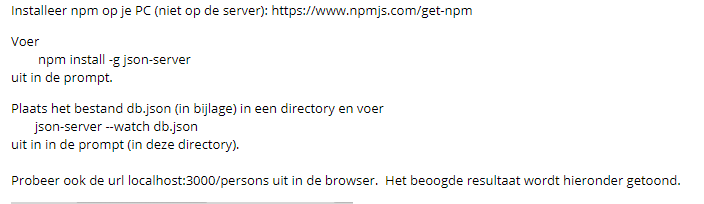
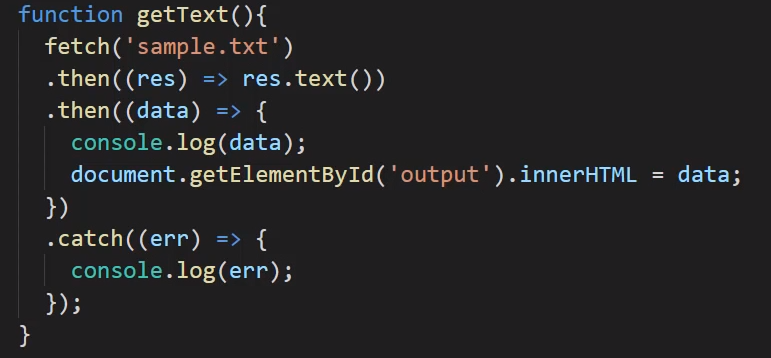
THANKS ROBBE^^  


**Try catch**  
**een finally wordt ALTIJD uitgevoerd**  
**hier maken we zelf de errormessage aan, deze error wordt dan geprint in de catchblok**

**JSON**  


**PROMISES**  
**of met een catchblok is beter**   
**geeft een reject na 100 miliseconden**

**AND finally{ FETCHES}**  
**json ziet er zo uit:**  


als die in de else gaat, doet die de volgende .then niet  
eerste then returned de inhoud, die wordt aan volgende then doorgegeven