In deze paragraaf ga je meer leren over de structuren in JS. Je hebt namelijk, net zoals bij Python condities, loops en ook functies.

Het is ook handig als je comments kunt maken, aangezien de complexiteit nu zal toenemen. Een single-line comment in JS maak je met #CODE //, voor multi-line gebruik je #CODE /\* \*/.

Condities vind je in JS voornamelijk in de vorm van if statements. Deze zien er als volgt uit:

#CODE

If (VOORWAARDE){

//DOE …

}

Het is ook mogelijk om een if-else statement te gebruiken, dat ziet er als volgt uit:

#CODE

If (VOORWAARDE){

//DOE …

} else {

//Doe …

}

Een andere optie is else if, dan kan er als volgt uitzien:

#CODE

If (VOORWAARDE){

//DOE …

} else if (VOORWAARDE){

//DOE …

}

Je kunt ook uitgebreidere voorwaarden opstellen, dit kan met AND en OR oftewel ‘en’ en ‘of’. In JS schrijf je AND als && en OR als ||.

Een voorbeeld kan zijn:

#CODE

datum = new Date();

uur = datum.getHours();

if (uur > 12 && uur < 18){

alert("Fijne middag");

} else if (uur > 18 && uur < 6) {

alert("Fijne avond");

} else if (uur > 6 && uur < 12) {

alert("Fijne ochtend");

}

JS heeft ook loops, oftewel herhalingsstructuren. Deze komen in de vorm van for- en while loops. Deze gaan beide door zolang een bepaalde eis voldaan wordt. Het verschil zit voornamelijk in de notatie.

Een for loop ziet er als volgt uit:

#CODE

for (i=0; i<AANTAL; i++) {

//doe iets

}

Het wordt vaak gebruikt om een door een lijst van waardes te gaan en iets met die waardes te doen. Een voorbeeld is:

#CODE

items = ['PHP', 'python', 'JS', 'HTML', 'CSS', 'C++', 'C#', 'C'];

for(i=0; i<items.length; i++){

alert(items[i]);

}

Dit ‘programma’ zal elk item uit ‘items’ laten zien. Het is ook te schrijven als:

#CODE

items = ['PHP', 'python', 'JS', 'HTML', 'CSS', 'C++', 'C#', 'C'];

for (item in items){

alert(items[item]);

}

Een while loop wordt vaak gebruikt bij processen die lang doorgaan. Denk aan een spel waarbij de speler kan bewegen zolang hij leeft. Het lastig om bij een klein programma een goed voorbeeld hiervan te geven. Een while loop kan er als volgt uitzien:

#CODE

i = 0;

amount = 10;

while(i < amount){

document.getElementById('element').textContent += i;

i++;

}

Dit ‘programma’ zal aan het element met id ‘element’ de getallen 0 tm 9 toevoegen. Let erop dat je de iterator (het getal dat steeds vergeleken wordt) steeds aanpast als je het niet voor altijd door wilt laten gaan. De loop stopt namelijk pas wanneer de conditie niet meer waar is.

Als laatste onderdeel van deze paragraaf behandelen we functies. Functies zijn stukken code die gegroepeerd zijn. Dit wordt vaak gebruikt voor herhaling, maar kan ook handig zijn voor een duidelijkere structuur. De syntax van een functie is:

#CODE

function FUNCTIE\_NAAM(NODIGE\_DATA){

//doe iets

}

Je kunt een functie ‘aanroepen’ doormiddel van #CODE FUNCTIE\_NAAM(NODIGE\_DATA); De functie zal dan worden uitgevoerd. Op de plek van NODIGE\_DATA kan data staan die de functie nodig heeft, denk aan variabelen die in de rest van de code voorkomen. Een voorbeeld van een kort programma met functies is:

#CODE

function updateTime(DoorTeSturenInfo){

datum = new Date();

tijd = datum.getTime();

document.getElementById('element').textContent = "Het is " + tijd + " milliseconde na 1 januari 1970. En \x22" + DoorTeSturenInfo + "\x22 werd doorgestuurd.";

}

function main(){

DoorTeSturenInfo = "Stuur door naar functie.";

setInterval(function(){

updateTime(DoorTeSturenInfo);

}, 1000);

}

main();

Dit programma zal de inhoud van het element met id ‘element’ vervangen met het aantal milliseconde sinds 1 januari en dit om de seconde verversen. Je hoeft niet de hele code te snappen, maar wel hoe de functie geschreven is en wordt opgeroepen ( updateTime(DoorTeSturenInfo); en main(); ). De ‘DoorTeSturenInfo’ is hier als voorbeeld gebruikt voor data dat buiten de functie bestaat. Een functie kan die data anders niet lezen (tenzij je global variables gebruikt, dit is buiten het bereik van de theorie).

#VRAGEN

1. Schrijf zelf een if statement
2. Pas de if statement aan en voeg er else-if aan toe.
3. Schrijf zelf een for loop.
4. Schrijf zelf een while loop.
5. Maak een kort programma met minimaal één functie.

#ANTWOORDEN