Gevorderd gebruik van functies

Overzicht

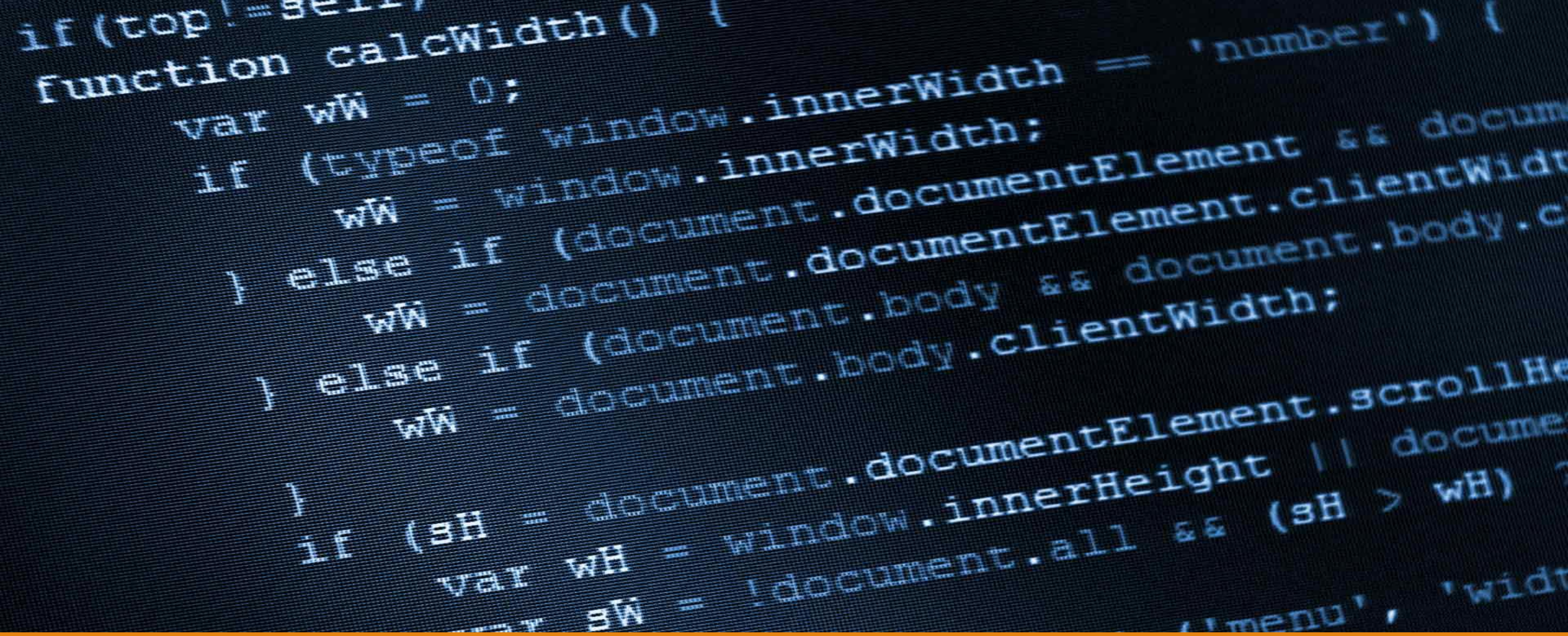
- Functies een variabel aantal argumenten meegeven
- Zelfuitvoerende functies *(“iffy’s”)*
- Bespreking van de term closures

Functies met een variabel aantal parameters

- In JavaScript is het niet verplicht om alle parameters mee op te geven
- In JavaScript is het mogelijk om meer parameters mee te geven dan gedefinieerd
  - Deze worden op de achtergrond in de extra parameter *“arguments”* geplaatst

De parameter arguments

- Toegevoegd aan elke functie binnen JavaScript
- Een *“array”* die alle argumenten van een functie bevat
  - Geen arrayfuncties beschikbaar!!


if (top != self)
function calcWidth() {
var wW = 0;
if (typeof window.innerWidth == 'number') {
wW = window.innerWidth;
} else if (document.documentElement && docum
wW = document.documentElement.clientWidt
} else if (document.body && document.body.c
wW = document.body.clientWidth;
}
if (sH = document.documentElement.scrollHe
var wH = window.innerHeight || docume
var sW = !document.all && (sH > wH)

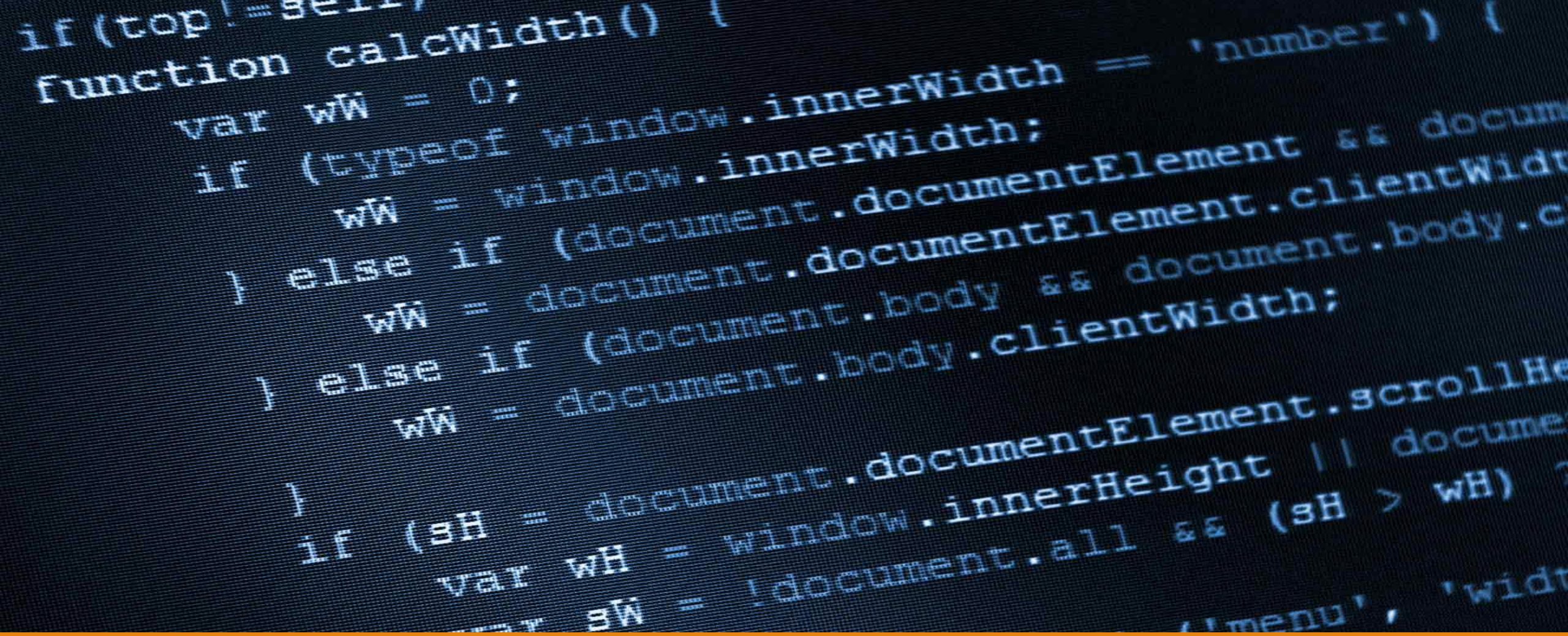
Demo

Immediately Invoked Function Expressions (IIFE)

- Functies die automatisch worden uitgevoerd
- Wordt veel gebruikt in bibliotheken *(zoals jQuery, …)*
- Uitgesproken als ‘iffy’

Uitvoeringscontext en global scope

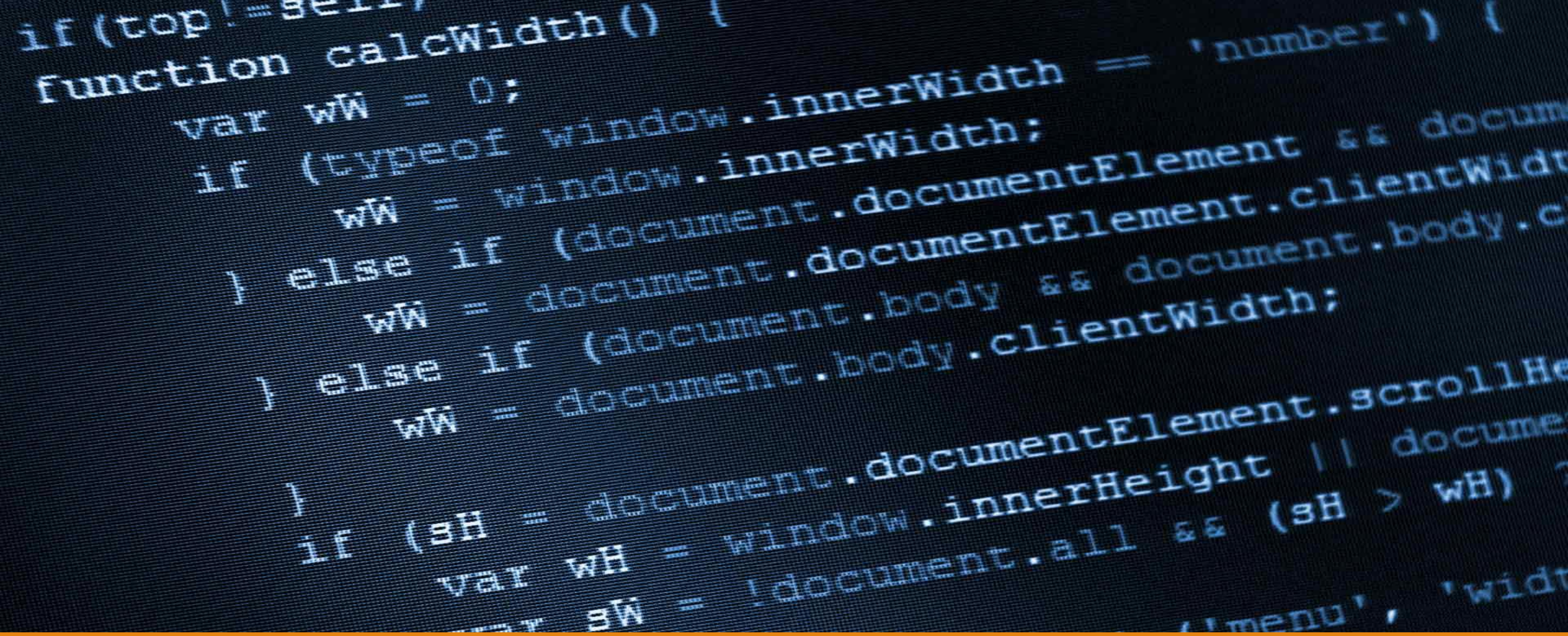
- Elke functie die wordt uitgevoerd creëert een nieuwe uitvoeringscontext
  - Variabelen en andere functies die binnen de hoofdfunctie worden gedeclareerd zijn wel zichtbaar binnen die functie maar niet erbuiten
  - Hierdoor kunnen er private variabelen en functies gedeclareerd worden zodat deze niet per ongeluk of expres van buitenaf gewijzigd worden
- Door het gebruik maken van een IIFE voorkomt u dat functies en variabelen in de global scope terecht komen


if (top != self)
function calcWidth() {
var wW = 0;
if (typeof window.innerWidth == 'number') {
wW = window.innerWidth;
} else if (document.documentElement && docum
wW = document.documentElement.clientWidt
} else if (document.body && document.body.c
wW = document.body.clientWidth;
}
if (sH = document.documentElement.scrollHe
var wH = window.innerHeight || docume
var sW = !document.all && (sH > wH)

Demo

Werken met closures

- Een functie blijft toegang behouden tot een variabele, ook als de functie zelf al is teruggekeerd
  - Hierdoor kunnen we data bijhouden tussen verschillende functie aanroepen


if (top != self)
function calcWidth() {
var wW = 0;
if (typeof window.innerWidth == 'number') {
wW = window.innerWidth;
} else if (document.documentElement && docum
wW = document.documentElement.clientWidt
} else if (document.body && document.body.c
wW = document.body.clientWidth;
}
if (sH = document.documentElement.scrollHe
var wH = window.innerHeight || docume
var sW = !document.all && (sH > wH)

Demo