

# Web advanced

---

JAVASCRIPT & JQUERY

# Jeffry Steegmans

---

**ASP.net ontwikkelaar** *(Hogeschool PXL)*

**Lector** *(PCVO Handel)*

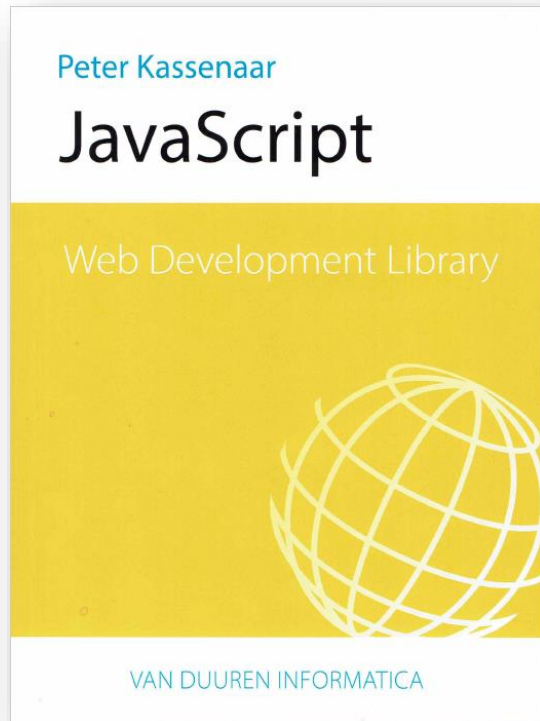
**Email:** [jeffry.steegmans@pxl.be](mailto:jeffry.steegmans@pxl.be)

# Cursus materiaal

---

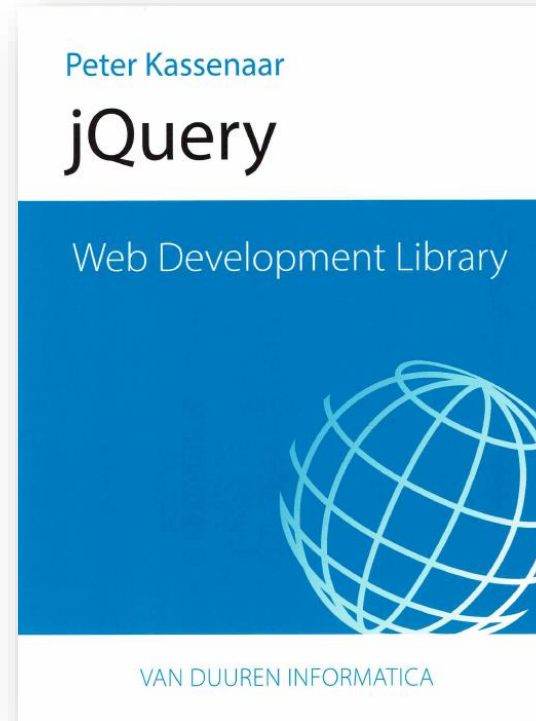
## JavaScript

Peter Kassenaar



## jQuery

Peter Kassenaar



# Microsoft Visual Studio Community

## Microsoft Visual Studio Code

---

**Gratis download:**

<https://www.visualstudio.com/downloads/>



# Kennismaking met JavaScript

---

# Overzicht

---

- Een korte geschiedenis van JavaScript
- Waarvoor wordt JavaScript gebruikt?
- Belangrijke begrippen die u moet kennen bij het werken met JavaScript
- Hoe JavaScript en HTML gecombineerd worden in webapps
- Een eerste script schrijven en de tags `<script>...</script>`
- Kennismaken met JavaScript-debugging

# Brendan Eich

---

- Ontwikkeld in 1995
- Combinatie van HTML en JavaScript was destijds bekend onder de naam Dynamic HTML (*DHTML*)
- Zonder JavaScript geen facebook, twitter, ...



# ECMAScript, JavaScript en versienummers

---

- Eerst liep het versienummer omhoog met het verschijnen van een nieuwe browserversie
- Nu is de ontwikkeling losgekoppeld van nieuwe versies van een browser en is de taal omgevormd tot een officiële, genormeerde en gestandaardiseerde programmeertaal
- European Computer Manufacturers Association (*ECMA*)
- Specificatie is vastgelegd in een document met het nummer 262 (*5<sup>de</sup> editie*) -> ECMAScript-262 5th Edition (*JavaScript*)



# Waarvoor wordt JavaScript gebruikt?

---

- **Formuliervalidatie:** *Gegevens in een webformulier op een pagina te controleren voordat het formulier verzonden wordt.*
- **Dynamische menu's en afbeeldingen:** *Menu's en afbeeldingen vervangen tijdens gebruik (fotocarrousel of uitklapmenu's)*
- **Aanpassingen van stijlen en animatie:** *Javascript kan in een pagina de aanwezigheid, positie en inhoud van elk element ophalen en manipuleren*
- **Ajax-webapplicaties:** *Na het laden van de pagina asynchroon delen van de pagina verversen of aanpassen*
- ...

# Kernbegrip – JavaScript core

---

- Bestaat uit een relatief kleine set instructies
  - Variabelen, lussen, teksten, arrays en objecten
- Bevat geen:
  - Opdrachten voor invoer en uitvoer
  - Netwerkmogelijkheden
  - Werken met bestanden
  - ...

# JavaScript-debuggers

---

- Een programmeeromgeving is niet compleet zonder debugger
- Alle moderne webbrowsers hebben een debugger (*F12*)



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# JavaScript-functies

---

- Alert is een ingebouwde functie.
  - U hoeft hem niet apart te definiëren
  - De browser zorgt ervoor dat het venster op het scherm wordt gezet, dat er een knop OK in staat enzovoort
- Een andere standaardfunctie is de functie *prompt()*
  - De gebruiker vragen iets in te vullen in een venster
  - De functie heeft 2 parameters
    - Tekst die in het venster verschijnt
    - Standaard tekst die in het tekst vak getoond wordt



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# Een inline event handler schrijven

---

- Een event plaatsen op een HTML element
  - Onclick
- Pas code uitvoeren na een gebeurtenis (event)
- Best practice
  - inline event handlers vermijden
  - eventlisteners creëren.



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# JavaScript-code in extern bestand

---

- Werkt ongeveer net zo als het opslaan van CSS-code in een extern stijlenbestand.
- Kan gekoppeld worden in meerdere HTML bestanden
  - Wijzigingen hoeven dan maar in 1 bestand doorgevoerd te worden
- De verwijzing tag MOET een complete tag zijn. De verkorte XHTML-notatie is ongeldig.

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# Statements, gegevens- typen en variabelen

---

# Overzicht

---

- De structuur van JavaScript-statements
- Verschillende gegevenstypen van JavaScript
- Variabelen declareren
- Gereserveerde woorden in JavaScript

# De syntaxis van JavaScript

---

- Eenvoudige syntaxis
- Niet zo streng als C, C# of Java
- Speciale voorbereidingen behalve de tag `<script>...</script>` zijn niet nodig
- Het toevoegen van deze regel zorgt er al voor dat het script werkt als een apart programma

# Statements

---

- Elk statements wordt afgesloten met een puntkomma
- Statements mag u (*net als HTML-code*) voor de leesbaarheid verdelen over verscheidene regels
- JavaScript negeert eventuele regelovergangen en inspringingen
- Plaats dus pas een puntkomma aan het einde van een statement en niet aan het eind van elke regel

# Hoofdletters en kleine letters

---

- JavaScript is case sensitive!
- Dit geldt voor de gereserveerde woorden en voor eigen functies en variabelen.
  - Anders maakt JavaScript ter plekke een nieuwe variabele aan

# Werken met variabelen

---

- De namen van variabelen mogen alle letters van het alfabet bevatten, zowel hoofd- als kleine letters
- Ook de getallen 0-9 en de underscore (\_) zijn toegestaan
- Namen mogen geen spaties of andere leestekens bevatten
- Het eerste teken van de variabelennaam moet een letter of de underscore zijn
- De naam van een variabele mag onbeperkt lang zijn (*moet op één tekstregel passen*)
- Mag geen gereserveerd woord (*false, return, if, ...*) zijn



# Gereserveerde woorden

---

break	else	instanceof	typeof
case	false	new	var
catch	finally	null	void
continue	for	return	while
debugger	function	switch	with
default	if	this	throw
delete	in	try	true
do			

# Toekomstig gereserveerde woorden

---

class

const

enum

export

extends

import

super

implements

interface

let

package

private

protected

public

static

yield

# Gegevenstypen

---

- JavaScript kent maar 2 gegevenstypen:
  - Primitive types (*primitieve typen*)
  - Object types (*complexe- of objecttypen*)

# Primitieve- of eenvoudige gegevenstypen

---

- **numbers (Getallen):** Elk willekeurig nummer
- **strings (Tekensreeksen):** Bestaan uit één of meer tekens (max 32000)
- **booleans (Logische waarden):** true of false
- **null:** De waarde van een ongedefinieerde variabele
- **NaN (Not a Number):** Waarde die sommige standaardfuncties teruggeven als zij als invoer een nummer verwachten maar een string krijgen
- **undefined:** variabele die wel gedeclareerd is, maar niet geïnitieerd

# Numbers

---

- Getallen kunnen zowel integers als floats zijn
- Een ingetypte waarde (*formulier*) geeft ALTIJD strings terug
  - Deze moeten eerst omgezet worden naar een getal!

# Getallen converteren met `parseInt()` en `parseFloat()`

---

- **`parseInt(waarde)`**: Set de parameter `waarde` om naar een geheel getal (*integer*). Als `waarde` niet naar een integer kan worden omgezet, wordt **NaN** teruggegeven
- **`parseFloat(waarde)`**: Zet de parameter `waarde` om naar een gebroken getal (*floating point*). Als `waarde` niet naar een gebroken getal kan worden omgezet, wordt **NaN** teruggegeven



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# Tekenreeksen of strings

---

- Reeksen tekens, die omsloten worden door dubbele of enkele aanhalingstekens
  - “Deze zin is een string”
  - ‘0’
  - ‘ik ben 30 jaar’
  - “222.5”
  - ‘7 + 7 = 14’
  - ...



# Lege string

---

- "" of ''
- Staat letterlijk niet in, zelfs geen spatie (*null*)

# Speciale tekens in string

---

- Regelterugloop (*carriage return*), tabs en andere niet-afdrukbare tekens
- Escapen door een backslash (\)
  - \n **Newline:** Een nieuwe regel
  - \t **Tab:** Verplaats de cursor naar de volgende tabstop
  - \\ **Backslash:** Toont 1 backslash
  - \" **Dubbele Quote:** Toont de dubbele quote
  - ...

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# Verschillende string functies

---

- **+**: Stringconcatenatie
- **length**: Aantal tekens in een string
- **charAt**: Karakter op een bepaalde index
- **substring**: Een deel van een string
- **indexOf**: De index positie van een karakter *(-1 indien niet gevonden)*
- **lastIndexOf**: De laatste positie van een karakter
- **replace**: string vervangen door andere string
- **toUpperCase/toLowerCase**: Omzetten naar hoofd/kleine letters

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# Booleaanse waarden

---

- `true` (*waar*) / `false` (*onwaar*)
- Veel functies, lussen of vergelijkingen retourneren een booleaanse waarde

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

# Demo

# Objecttypen

---

- Alle typen die niet tot een van de enkelvoudige gegevenstypen (*number*, *string*, *boolean*, *null*, *undefined* en *NaN*) behoren, zijn **complexe typen**, of **objecttypen**
  - object
  - array
  - function
- In een later hoofdstuk hierover meer



# Operatoren

---

# Overzicht

---

- Operatoren gebruiken
- De groepen operatoren die JavaScript kent
- Operatoren nuttig inzetten

# Variabelen bewerken met operatoren

---

- **Toewijzingsoperatoren** (*assignment operators*): toewijzen/wijzigen van een waarde aan een variabele
- **Wiskundige operatoren** (*arithmetic operators*): optellen, aftrekken, delen vermenigvuldigen en andere wiskundige bewerkingen
- **Stringoperatoren** (*string operators*): operatoren voor het manipuleren van string
- **Logische operatoren** (*logical operators*): Werkt met booleaanse waarden; meestal gebruikt voor het controleren van voorwaarden

# Variabelen bewerken met operatoren

---

- **Bitoperatoren** (*bitwise operators*): veranderen de afzonderlijke bits van een numerieke waarde. Werken met het binaire talstelsel
- **Vergelijkingsoperatoren** (*comparision operators*): testen van een relatie tussen twee waarden of objecten (*true of false*)
- **Speciale operatoren** (*special operators*): operatoren die niet onder een van de hiervoor genoemde typen vallen.
  - voorwaardelijke operator (? :)
  - new
  - delete

# Toewijzingsoperatoren

---

- `=` (*isgelijkaan*) teken
  - Eenvoudigste toewijzingsoperator
  - Wijst de waarde rechts van de operator toe aan de variabele links ervan
- Verkorte notatie (`+=`, `-=`, `*=`, `/=`, `%=`)
  - `a = a + 1 --> a += 1`
- Nog kortere notatie (`++`, `--`)
  - `a = a + 1 --> a++`



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

# Demo

# Wiskundige operatoren

---

- wiskundige bewerkingen uitvoeren op variabelen
  - +
  - -
  - \*
  - /
  - ...



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# Stringoperatoren

---

- JavaScript kent maar één operator die speciaal voor strings is bedoeld: de concatenatieoperator (+)
- Opletten dat het juiste gegevenstype gebruikt wordt
  - "50" + "5" = "505"



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# Logischeoperatoren

---

- Werken met booleaanse waarden
- Kijken of een bepaalde expressie waar of onwaar is
- **&&** (*AND*): retourneert **true** als **beide operanden waar** zijn
- **||** (*OR*): retourneert **true** als **één van beide operanden waar** is **én** als ze **beide waar** zijn
- **!** (*NOT*): Als de operand **true** is, retourneert de operator **false** en omgekeerd



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo

# Vergelijkingsoperatoren

---

- Geven een booleaanse waarden terug
- Kunnen ook werken met getallen



# Vergelijkingsoperatoren

---

- **==** (*is gelijk aan*): **true** als de **operanden gelijk** zijn
- **===** (*strikt is gelijk aan*): **true** als de **operanden gelijk** zijn **én van hetzelfde type**
- **!=** (*is niet gelijk aan*): **true** als de **operanden niet gelijk** zijn
- **!==** (*strikt is niet gelijk aan*): **true** als de **operanden niet gelijk** zijn **én niet van hetzelfde type** zijn



# Vergelijkingsoperatoren

---

- $>$  (*groter dan*): **true** als de **linkeroperand** **groter** is dan de rechter
- $<$  (*kleiner dan*): **true** als de **linkeroperand** **kleiner** is dan de rechter
- $>=$  (*groter dan of gelijk aan*): **true** als de **linkeroperand** **groter** is dan de rechter of **gelijk** is aan de rechter
- $<=$  (*kleiner dan of gelijk aan*): **true** als de **linkeroperand** **kleiner** is dan de rechter of **gelijk** is aan de rechter

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# Voorwaardelijkeoperator

---

- Wordt gekeken of er aan een bepaalde voorwaarde voldaan is
  - is de voorwaarde waar, dan is het resultaat van de expressie true en wordt een voorwaarde uitgevoerd die bij true hoort. Dit is het deel achter het vraagteken
  - is de expressie niet waar, dan wordt het resultaat false en wordt de voorwaarde uitgevoerd die bij false hoort. Dit is het deel achter de dubbele punt.

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo



# De operator typeof

---

- Voert geen bewerking uit met een variabele, maar retourneert het type ervan
  - **undefined**: 'undefined'
  - **null**: 'object'
  - **true of false**: 'boolean'
  - **een getal**: 'number'
  - **een string**: 'string'
  - **een functie**: 'function'
  - **alles wat geen functie of een van de overige typen is**: 'object'

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

# Demo



# Bewerkingsvolgorde

---

- Zodra er meer dan één operator in een statement staat, wordt de volgorde belangrijk waarin Javascript de operatoren afhandelt
  - Haakjes, functieaanroepen of arrayindex: () en []
  - Negatie (-), increment (++) en decrement (--)
  - Vermenigvuldigen (\*), delen (/) en modulo (%)
  - Optellen (+) en aftrekken (-)
  - Bitwise operatoren (<<, >> en >>>)
  - Vergelijkingsoperatoren (<, >, <= en >=)
  - Is gelijk aan en is ongelijk aan (==, === en !=, !==)
  - Bitwise AND (&)
  - Bitwise XOR (^)

# Bewerkingsvolgorde

---

- Bitwise OR (|)
  - Logisch AND (&&)
  - Logisch OR (||)
  - Voorwaardelijke operator (? :)
  - Toewijzingsoperator (=)
  - De komma(,), gebruikt voor het scheiden van parameters
- 
- Indien gelijk wordt de expressie van links naar rechts geëvalueerd
  - Kan beïnvloed worden door haakjes



```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

## Demo