

Client-side scripting

Moderne browsers zijn tot meer in staat dan alleen het laten zien van opgemaakte tekst. In HTML-code kunt u een script (een stukje programma) opnemen, dat door de browser wordt gelezen en in het geheugen bewaard. Zo'n stukje code kan, bijvoorbeeld door het klikken op een knop, worden geactiveerd. Als de gebruiker niet op de knop klikt, wordt de code niet geactiveerd. Bij dit alles is de server niet betrokken. Het vindt plaats binnen de browser, op uw eigen computer. Het is client-side scripting.

De naam client-side scripting suggereert dat er ook server-side scripting bestaat. Dat is inderdaad het geval. Als u bijvoorbeeld in Google Maps een straatnaam intikt, gaat de server op zoek naar die naam, en de bijbehorende plattegrond en andere informatie, om het vervolgens naar u te sturen. Op de server is een script aanwezig dat hiervoor zorgt.

In oudere pagina's ziet u vaak een script op deze manier:

```
<SCRIPT language="JavaScript">
<!--
...
//-->
</SCRIPT>
```

Op de puntjes staat het feitelijke script. Vroeger moest men aangeven welke scripttaal gebruikt werd: `language="JavaScript"`. Omdat er ook browsers waren die geen scripts konden lezen, werd het script als HTML-commentaar in de pagina gezet, tussen `<!--` en `-->`. Browsers die het script niet konden lezen, zagen het als HTML-commentaar, andere browsers voerden het script uit.

In HTML5 is JavaScript standaard, alle moderne browsers ondersteunen deze taal. Er is dus weinig reden om bovenstaande gewoonte te handhaven.

Wij gebruiken voor scripts deze eenvoudige vorm:

```
<script>
...
</script>
```

Waar plaatst u het script in de HTML-code?

In principe kan het script in de header of in de body van de HTML-code staan. In veel gevallen maakt het script gebruik van elementen die op de pagina zijn gedefinieerd. In dat geval is vlak voor het einde van de body vaak de beste plaats om het script te plaatsen. Dus:

```
<body>
...

<script>
...
</script>
</body>
```

Later zullen we nog andere manieren zien om een script toe te voegen.

Commentaar

Het is goed gebruik om code commentaar toe te voegen. Commentaar is toelichting die bedoeld is voor de menselijke lezer, bijvoorbeeld om de werking van de code te verduidelijken. Commentaar is voor uzelf, voor mede-programmeurs of voor de docent. De browser negeert al het commentaar in de code.

Om commentaar in het script te zetten, gebruikt u de dubbele slash `//`. Alles wat na deze dubbele slash op dezelfde regel komt, is commentaar. Bijvoorbeeld:

```
<script>
...
function myFunction() { //wordt geactiveerd na klikken op knop
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
...
</script>
```

Commentaar kan ook zelfstandig op een regel staan:

```
<script>
...
// functie die wordt geactiveerd na klikken op knop
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
...
</script>
```

Als het commentaar niet op een regel past moet u elke regel van dubbele slashes voorzien:

```
// eerste regel commentaar
// tweede regel commentaar
```

Alternatief is om gebruik te maken van `/*` en `*/`. Alles wat tussen slash sterretje en sterretje slash staat is commentaar. Een dergelijk commentaar kan meerdere regels omvatten. Bijvoorbeeld:

```
/*
eerste regel commentaar
tweede regel
derde regel
*/
```

Zie ook de voorbeelden bij Single Line Comments en Multi-line Comments op de pagina [JS Comments](#).

Code uitschakelen door er commentaar van te maken

Het plaatsen van twee slashes in de code kan ook erg handig zijn in de testfase. De code die staat tussen `//` en het einde van de regel wordt immers niet uitgevoerd. Op die manier kunt u snel een stukje code (tijdelijk) uitschakelen.

De tag noscript

Behalve een script-tag die aangeeft waar JavaScript-code tussen de HTML-code staat, bestaat er ook een noscript-tag. Achterhaal zelf de betekenis van deze tag.