# Html Inleiding

**Zonder een gebruikersinterface blijft een programmeertaal onzichtbaar. Windows-formulieren of HTML leveren een tastbare interface waarmee programmeertalen gebruikers kunnen bereiken. In die module gaan we alle aspecten van de HTML gebruikersinterface van dichtbij bekijken.**

# Spaghetti of lasagne?

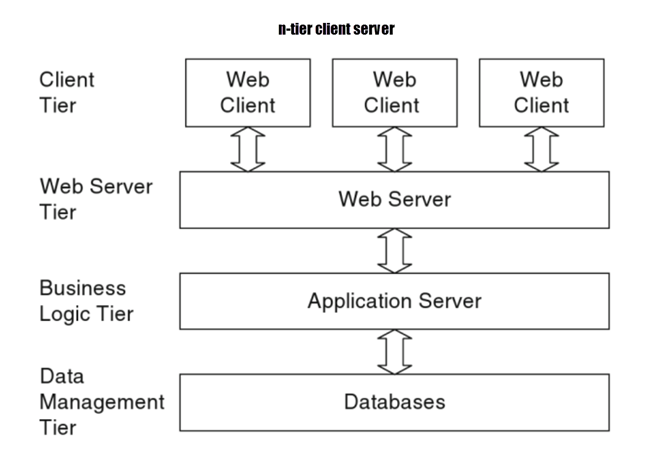
Nog niet zolang geleden stopten programmeurs alles in één enkel bestand met de naam index.htm of index.html en noemden het een webpagina.

Meer en meer geraakten web ontwikkelaars ervan overtuigd dat zo’n webpagina eigenlijk uit vier verschillende soorten dingen bestaat:

1. inhoud (content);
2. de presentatie of de opmaak van die inhoud;
3. interactie met de webserver;
4. interactie met de website;

Uiteindelijk zagen ze in dat je de code voor die drie verschillende dingen beter in aparte bestanden kan bijhouden.

# 4-tier



***n-tier client server***

Het opsplitsen van code al naargelang het opzet en doel maakt dat:

* code herbruikbaar is;
* code niet gedupliceerd wordt;
* code gemakkelijk te wijzigen is;
* code in team geschreven kan worden;

Maar de belangrijkste reden voor het opsplitsen van code al naargelang opzet en doel bestaat in de mogelijkheid om te voorzien in de vele verschillende manieren waarop gebruikers webcontent kunnen raadplegen. Sommige gebruikers beschikken over de laatste browsers, die de nieuwste snufjes van CSS en JavaScript ondersteunen. Andere gebruikers, scholen, bedrijven en overheid voeren een streng ITC beleid en beschikken misschien niet over de nieuwste browsers of schakelen JavaScript zelfs uit.

Slechtzienden bijvoorbeeld, die screen readers of screenvergroters gebruiken, vinden een flashy lay-out met veel toeters en bellen eerder hinderlijk.

De beste manier om aan al die verschillende behoeften van bezoekers te voldoen bestaat erin webpagina’s te bekijken als opgebouwd uit vier lagen.

Als we een website maken doorlopen we de volgende vier lagen. De laatste laag is optioneel:

1. we beginnen met de content in HTML formaat. Dat is de grondlaag die door om het even welke browser weergegeven kan worden;
2. daarna maken we de pagina op door een laag presentatie-informatie toe te voegen. Daarvoor gebruiken we CSS. De site zal er beter uitzien in browser met CSS stijlen;
3. na indeling en opmaak, introduceren we de ServerScripttalen zoals PHP en ASP.NET om een dynamisch werkende laag toe te voegen waarmee we content dynamisch kunnen maken.
4. en we vergeten JavaScript niet om een interactieve laag toe te voegen;

HTML krijgt alleen de taak van de indeling toegegewezen.

Als we de HTML, CSS, JavaScript en server-scripttalen gescheiden houden wordt het gemakkelijker om tenminste de grondlaag in bepaalde browsers weer te geven waarin de presentatie- en/of werkingslaag niet kan uitgevoerd worden. Die aanpak wordt ook wel progressieve verbetering genoemd.