

# Kennis Check Blok 3

## Kennis Check Blok 3

[datasource](#)

### HTML

#### 1. Wat is HTML?

HTML betekent **HyperText Markup Language**. Het is de **taal waarmee je een webpagina maakt**. HTML vertelt de browser wat er op een website moet staan, zoals:

- tekst (bijvoorbeeld een titel of paragraaf),
- afbeeldingen (foto's of plaatjes),
- knoppen en formulieren,
- en links naar andere pagina's.

Je gebruikt HTML om de **inhoud en structuur** van je website te bepalen. Denk aan het skelet van een huis: HTML zorgt ervoor dat alles op de juiste plek staat.

Later kun je met **CSS** de kleuren en opmaak toevoegen, en met **JavaScript** de pagina interactief maken.

#### 2. Wat gebeurt er als je in "Phoenix Code" op "Run" of "Live Server" klikt?

Dan wordt je HTML-bestand zichtbaar als een echte webpagina in de browser. Je ziet meteen het resultaat van jouw code.

#### 3. Waarom gebruiken we `<!DOCTYPE html>` aan het begin van elke HTML-pagina?

Deze regel vertelt de browser dat het om een moderne HTML5-pagina gaat. Zo weet de browser hoe hij de pagina moet weergeven.

#### 4. Waarom staat er tekst tussen `<h1>...</h1>` of `<p>...</p>`?

HTML werkt met tags. Die tags geven betekenis aan de tekst: `<h1>` betekent een kop (grote titel), `<p>` betekent een paragraaf (stukje tekst). Het zorgt voor structuur op je pagina.

#### 5. Wat is het verschil tussen `<strong>` en `<em>` in HTML?

`<strong>` maakt tekst vet en betekent: dit is belangrijk. `<em>` maakt tekst schuin en betekent: leg hier nadruk op. Beide zorgen dat de tekst opvalt, maar op een andere manier.

#### 6. Waarom gebruik je `<br>` in plaats van `<p>`?

Met `<br>` breek je alleen de regel af, bijvoorbeeld in een adres of opsomming. `<p>` is voor een heel nieuw stukje tekst, paragraaf (met ruimte ervoor en erna).

`<br>` staat voor "break", en `<p>` staat voor "paragraaf".

#### 7. Wat doet de `alt`-tekst bij een afbeelding?

Als de afbeelding niet wordt geladen, verschijnt de alt-tekst. Die beschrijft wat er op de afbeelding staat. Dat helpt ook mensen met een visuele beperking.

#### 8. Wat is het verschil tussen een `<ul>` en een `<ol>` lijst?

`<ul>` is een opsomming zonder volgorde (met **bolletjes**).

`<ol>` is een **genummerde** lijst, waarbij de volgorde belangrijk is (zoals een top 5).

#### 9. Waarom moet elke rij in een tabel evenveel kolommen hebben?

Anders weet de browser niet hoe hij de tabel netjes moet weergeven. Dan worden kolommen scheef of verdwijnen cellen. Elke rij moet dus evenveel `<td>` of `<th>` hebben.

#### 10. Wat zou een formulier kunnen doen op een echte website?

Een formulier kan gegevens van een gebruiker opslaan of verzenden. Denk aan: aanmelden, iets bestellen, contact opnemen of een vraag stellen. In de les doet het formulier nog niets, maar met PHP of JavaScript kun je het echt laten werken.

## CSS

### 1. Waarom gebruiken we CSS op een webpagina?

CSS zorgt ervoor dat een website er mooi uit ziet. Je kunt kleuren, lettertypes en indeling aanpassen. Zonder CSS ziet een site er saai uit: alleen zwart-wit tekst zonder stijl.

### 2. Wat is het verschil tussen inline CSS en een style-blok?

Inline CSS staat direct in het HTML-element. Een style-blok zet je bovenin de pagina in de `<head>`. Daarmee kun je stijlen herhalen en overzichtelijker werken.

### 3. Waarom is het handig om een style-blok te gebruiken in plaats van inline CSS?

Met een style-blok kun je meerdere elementen tegelijk opmaken. Je hoeft dan niet elke keer dezelfde stijlregels te herhalen. Dat maakt je code netter en makkelijker aan te passen.

### 4. Wat is een class in CSS en waarom gebruik je dat?

Een class is een naam die je aan meerdere HTML-elementen kunt geven. Zo kun je dezelfde opmaak toepassen op verschillende plekken. Handig als je bijvoorbeeld meerdere blokken dezelfde kleur wilt geven.

### 5. Wat is het verschil tussen een element-selector, een class-selector en een id-selector?

Een element-selector (zoals `p`) geldt voor alle paragrafen. Een class-selector (zoals `.belangrijk`) geldt voor elementen met die class. Een id-selector (zoals `#titel`) is uniek en gebruik je maar één keer.

## 6. Wat is het box-model en waarom is het belangrijk?

Het box-model laat zien hoe ruimte om een element werkt: de tekst zit in het midden (content), dan padding, een border en buitenste ruimte heet margin. Hiermee bepaal je hoe groot een blok is en hoeveel ruimte ertussen zit.

## 7. Wat is het verschil tussen margin en padding?

Padding is de ruimte binnenin een element, tussen de tekst en de rand. Margin is de ruimte buiten een element, tussen het blok en andere blokken.

## 8. Waarom gebruik je flexbox voor layout?

Flexbox maakt het makkelijk om blokken netjes naast of onder elkaar te zetten. Je kunt ze makkelijk centreren of ruimte tussen de blokken maken, zonder moeilijke trucjes.

## 9. Wat betekent responsive design en hoe helpt flexbox daarbij?

Responsive design betekent dat je website er goed uit ziet op elk scherm, groot of klein. Flexbox zorgt ervoor dat blokken automatisch aanpassen aan de schermgrootte.

## 10. Wat gebeurt er als je een element meerdere classes geeft?

Dan worden de stijlen van alle classes toegepast op dat element. Zo kun je stijlen combineren, bijvoorbeeld `.menu` en `.speciaal`. Dit geeft je veel controle over de opmaak.

# JavaScript

## 1. Wat doet JavaScript op een webpagina?

JavaScript zorgt ervoor dat de webpagina dingen kan doen, zoals reageren op klikken of iets laten zien als je op een knop drukt. Zonder JavaScript is een pagina stil. Met JavaScript kun je hem interactief maken.

## 2. Wat is het verschil tussen HTML, CSS en JavaScript?

HTML is de inhoud, CSS is de opmaak (zoals kleuren en lettertypes) en JavaScript zorgt voor actie (zoals iets doen als je klikt of typt).

## 3. Waarom gebruiken programmeurs vaak de console?

De console laat zien wat er in de code gebeurt. Je kunt er fouten opsporen en controleren of je code werkt zoals je denkt. Het is een soort gereedschapskist.

## 4. Wat is een variabele en waarom gebruik je die?

Een variabele is een soort doosje waarin je iets bewaart, zoals een naam of een getal. Zo kun je er later nog iets mee doen of het laten zien.

## 5. Wat is het verschil tussen `let` en `const`?

Met `let` kun je het later nog veranderen, met `const` niet. Dus `const` gebruik je als iets altijd hetzelfde moet blijven.

## 6. Waarom zijn functies handig in je code?

Een functie is een stukje code dat je een naam geeft. Je kunt het steeds opnieuw gebruiken. Zo hoef je niet elke keer alles opnieuw te typen.

## 7. Wat is een parameter in een functie?

Een parameter is informatie die je aan een functie meegeeft. Zo kun je de functie flexibeler maken, bijvoorbeeld iemand begroeten met zijn eigen naam.

## 8. Wat is een voorwaarde (if-statement) en waar gebruik je die voor?

Een voorwaarde kijkt of iets waar is. Bijvoorbeeld: "Ben je 18 of ouder?" Als ja, dan doe je iets. Als nee, dan iets anders. Zo kun je keuzes maken in je code.

## 9. Wat is een event in JavaScript?

Een event is iets dat gebeurt, zoals een klik of iets typen. Met JavaScript kun je zeggen: als dit gebeurt, voer dan deze code uit.

Voorbeelden van events zijn: 'click', 'mouseover', 'keydown', ....

## 10. Hoe haal je invoer uit een formulier met JavaScript?

Je gebruikt `document.getElementById("id").value` om op te halen wat iemand in een vakje heeft getypt. Zo kun je iets met die invoer doen, zoals berekenen of controleren.

# Opdracht

Maak de kennis-check.

# Inleveren

Aan het einde van de kennis-check ontvang je een certificaat. Maak een schermafdruck en lever deze in.

---

Revision #9

Created 20 June 2025 19:11:22 by Max

Updated 12 July 2025 08:17:20 by Max