

Van website naar webapp

Vier lesbrieven van 50 minuten. Voor beginnende programmeurs in het VO. Leerlingen bouwen, voorspellen, testen en leggen uit wat hun code doet.

De leerroute

- Les 1 - HTML en CSS: een eigen projectpagina.
- Les 2 - JavaScript: een knop, een keuze en een score.
- Les 3 - PHP: invoer verwerken op de server.
- Les 4 - Arrays, lussen en tijdelijk opslaan met JSON.

Wat heb je nodig?

Een computer met browser, een eigen Webbouwers VO-project en de groepslink van de docent. Voor de vrije PHP-proeven in les 3 en 4 moet de aparte PHP-werkplaats door de beheerder zijn aangesloten.

Zo gebruik je deze bundel

Elke les heeft twee leerlingpagina's. Achterin staan docentnotities, antwoorden en een presentatiescript. Werk met bijnamen en fictieve inhoud: gepubliceerde leerlingprojecten zijn via een link openbaar.

Eerst begrijpen, dan uitbreiden

Voorspel wat er gebeurt. Verander één ding. Test het resultaat. Beschrijf daarna in je eigen woorden waarom het werkt. Een werkend programma is pas het begin: je uitleg telt mee.

Voor de docent

Vooraf klaarzetten

- Maak een testgroep en controleer een leerlinglogin. Gebruik bij voorkeur een blanco VO-startopzet.
- Probeer de HTML/JavaScript-voorvertoning en de PHP-werkplaats zelf.
- Controleer of leerlingen hun project kunnen opslaan en heropenen.
- Laat leerlingen een andere tab voor de lesbrief gebruiken. Maak afspraken over openbare inhoud.

Lesorganisatie

Werk in tweetallen met twee rollen: de bouwer typt, de uitlegger voorspelt en controleert. Wissel halverwege. Laat ieder tweetal kort één bewuste wijziging demonstreren.

Een fout is onderzoeksmateriaal

Laat leerlingen eerst lezen: wat staat er precies, bij welke regel, en wat heb ik als laatste veranderd? Herstel één ding tegelijk. Laat ze hun fout en oplossing vastleggen.

Wat draait waar?

- HTML en CSS vormen de pagina in de browser.
- JavaScript reageert in de browser. Het voorbeeld wordt bewust gestart.
- PHP krijgt invoer via de werkplaats en geeft een antwoord van de aparte server terug.
- In deze basisreeks worden de browserquiz en de PHP-proef los getest. Een quizscore kan handmatig als PHP-invoer worden meegestuurd.

Als de vrije PHP-werkplaats nog niet gereed is

Les 1 en 2 en de vaste PHP-lessen werken wel. Les 3 en 4 over zelfgeschreven PHP wachten tot de aparte werkplaats aangesloten en getest is. Presenteer die uitvoering niet als actief voordat dit is gecontroleerd.

Les 1 - Geef je verhaal een vorm

Doel

Je kunt uitleggen wat HTML en CSS doen en een kleine projectpagina veranderen.

Start - 5 minuten

Open een blanco VO-project. Bedenk een fictief onderwerp: een planeet, hobby, boek of museum. Gebruik een bijnaam als je iets over jezelf schrijft.

Onderzoek - 10 minuten

Bekijk de HTML hieronder. Voorspel welke tekst de grootste kop wordt en welke onderdelen een lijst vormen. Plaats de code in het HTML-vak.

```
<!-- HTML beschrijft de onderdelen. -->
<main>
  <h1>Mijn ontdekkingen</h1>
  <p>Vandaag onderzoek ik de ruimte.</p>
  <h2>Dit wil ik weten</h2>
  <ul>
    <li>Waarom lijkt Mars rood?</li>
    <li>Hoe ver staat de maan?</li>
  </ul>
</main>
```

Bouwen - 15 minuten

- Verander de hoofdkop en de introductie.
- Voeg een derde vraag aan de lijst toe.
- Voeg een tweede alinea toe.
- Schrijf boven één onderdeel een HTML-comment in je eigen woorden.

Les 1 - Test en verklaar

Vormgeven - 10 minuten

Plaats deze code in het CSS-vak. Verander daarna twee kleuren en één afstand. Klik steeds op Bekijk resultaat.

```
/* CSS bepaalt de vormgeving. */
body {
  font-family: system-ui, sans-serif;
  background: #eef3ff;
  color: #173451;
  padding: 2rem;
}
main {
  background: white;
  padding: 2rem;
  border-radius: 18px;
}
h1 { color: #245dd8; }
```

Laat zien - 5 minuten

Laat je partner een regel aanwijzen. Vertel welk zichtbaar onderdeel daarbij hoort. Laat ook zien wat er verandert als je die regel aanpast.

Afsluiten - 5 minuten

- Bewaar je project en open het opnieuw.
- Noteer: wat is het verschil tussen HTML en CSS?
- Noem één wijziging en beschrijf het effect.

Klaar?

Geef de tussenkop een eigen kleur. Voeg een CSS-comment toe dat uitlegt waarom je die kleur hebt gekozen. Houd de tekst goed leesbaar.

Inleveren

Je projectpagina, één eigen comment en een uitleg van drie zinnen.

Les 2 - Laat een knop reageren

Doel

Je kunt een klikgebeurtenis, een variabele en een voorwaarde herkennen en aanpassen.

Voorspellen - 10 minuten

Open het VO-voorbeeld met de ruimtequiz. Klik nog niet. Zoek in JavaScript waar de score wordt onthouden. Zoek daarna de voorwaarde die bepaalt of een antwoord goed is.

Probeer - 10 minuten

- Voer de quiz uit en kies een fout antwoord.
- Start opnieuw en kies het juiste antwoord.
- Leg uit waarom je niet meerdere antwoorden achter elkaar kunt kiezen.

Vergelijk de code

```
let punten = 0;

knoppen.forEach(knop =>
  knop.addEventListener('click', () => {
    const goed = knop.dataset.goed === 'ja';
    if (goed) punten += 1;
    document.getElementById('score').textContent = punten;
    knoppen.forEach(k => k.disabled = true);
  })
);
```

Dit fragment hoort bij het volledige ruimtequiz-voorbeeld. De variabele knoppen wordt daar eerder gemaakt.

Bouwen - 15 minuten

Maak een andere vraag met drie antwoorden. Geef precies één antwoord data-goed="ja". Verander de feedbackteksten. Laat een goed antwoord twee punten opleveren.

Les 2 - Je programma uitleggen

Onderzoek - 5 minuten

- Wat verandert er als je alleen de knoptekst wijzigt, maar data-goed laat staan?
- Waarom begint punten met 0?
- Welke regels brengen de quiz terug naar de beginsituatie?

Testplan - 5 minuten

Maak een tabel met vier proeven: fout antwoord, goed antwoord, tweede klik en opnieuw starten. Schrijf eerst de verwachte uitkomst op. Noteer daarna de echte uitkomst.

Uitleg - 5 minuten

Voeg twee JavaScript-comments toe. Leg met eigen woorden uit wat de klikfunctie doet en waarom de knoppen na het antwoord uitgaan.

Inleveren

- Een eigen quizvraag met drie antwoorden.
- Een score van twee punten bij het juiste antwoord.
- Je testtabel en twee comments.

Klaar?

Bedenk hoe je een tweede vraag zou toevoegen. Teken eerst de stappen op papier: wanneer verandert de vraag, wanneer worden knoppen weer actief en welke waarde moet de score behouden?

Reflectie

Een knop kan zichtbaar zijn zonder iets te doen. Welke verbinding maakt JavaScript tussen de HTML en jouw programma?

Les 3 - Vraag het aan PHP

Doel

Je schrijft zelf PHP die invoer leest, controleert en een antwoord teruggeeft. Je kunt het verschil met JavaScript in de browser uitleggen.

Start - 5 minuten

Open Zelf PHP schrijven. Laad het voorbeeld Tekst en variabelen. In het invoervak heet een veld naam. Geef het een fictieve waarde.

Voorspellen - 10 minuten

Voorspel de tekst die PHP terugstuurt. Verander daarna alleen de invoer en voer de code uit. Verander vervolgens de begroeting in de code.

```
<?php
$naam = $_POST['naam'] ?? 'Wereld';

echo '<h1>Hallo, '
    . htmlspecialchars($naam, ENT_QUOTES, 'UTF-8')
    . '!\</h1>';
```

Onderzoek

- Welke waarde leest \$_POST?
- Wat gebeurt er als je het invoerveld naam verwijdert?
- Waarom wordt htmlspecialchars gebruikt als invoer in HTML terechtkomt?

Verder - 10 minuten

Laad het voorbeeld Rekenen en een voorwaarde. Maak twee invoervelden: goed met waarde 3 en totaal met waarde 5. Voorspel het percentage vóór je de code uitvoert.

Les 3 - Controleer je invoer

```
<?php
$goed = (int) ($_POST['goed'] ?? 0);
$totaal = (int) ($_POST['totaal'] ?? 0);

if ($totaal < 1 || $goed < 0 || $goed > $totaal) {
    echo 'Controleer je invoer.';
} else {
    $percentage = round($goed / $totaal * 100);
    echo 'Je hebt ' . $percentage . '% goed.';
}
```

Bouwen en testen - 15 minuten

- Test 3 van 5, 5 van 5 en 0 van 5.
- Test daarna totaal=0 en goed=8 met totaal=5.
- Voeg een passende complimenttekst toe voor een percentage van 80 of hoger.

Afsluiten - 10 minuten

Kopieer een zelfgekozen quizscore handmatig naar de PHP-invoervelden. Leg uit: JavaScript reageerde in jouw browser; nu is invoer verstuurd en verwerkt in de aparte PHP-werkplaats. De quiz en de PHP-proef zijn in deze les nog losse onderdelen.

Inleveren

Je PHP-code, vijf testgevallen en een uitleg van waar PHP wordt uitgevoerd. Noem ook waarom je controleert voordat je deelt.

Les 4 - Van lijst naar opgeslagen tekst

Doel

Je gebruikt een array en een foreach-lus en onderzoekt hoe gegevens als JSON kunnen worden geschreven en teruggelezen.

Voorspellen - 10 minuten

Laad het PHP-voorbeeld Een lijst met een lus. Voorspel hoeveel lijstpunten de uitvoer bevat. Voeg daarna twee planeten toe.

```
<?php
$planeten = ['Mercurius', 'Venus', 'Aarde', 'Mars'];

echo '<ul>';
foreach ($planeten as $planeet) {
    echo '<li>'
        . htmlspecialchars($planeet, ENT_QUOTES, 'UTF-8')
        . '</li>';
}
echo '</ul>';
```

Onderzoek - 10 minuten

- Wat bewaart een array?
- Hoe vaak wordt de code tussen de accolades uitgevoerd?
- Wat verandert er als de array leeg is?

Bouwen - 10 minuten

Maak een eigen lijst met vier fictieve onderwerpen. Voeg boven de lus een comment toe die uitlegt wat de lus doet. Bekijk de tekstuitvoer en de HTML-weergave.

Les 4 - Schrijven en teruglezen

Proef - 10 minuten

Laad Tijdelijk schrijven en teruglezen. Maak het invoerveld bericht met een korte fictieve notitie.

```
<?php
$bericht = $_POST['bericht'] ?? 'Mijn eerste notitie';
$bestand = 'notitie.json';

file_put_contents(
    $bestand,
    json_encode(['bericht' => $bericht])
);

$terug = json_decode(file_get_contents($bestand), true);
echo htmlspecialchars($terug['bericht'], ENT_QUOTES, 'UTF-8');
```

Belangrijk onderscheid

Het bestand bestaat alleen tijdens deze ene proef. Iedere volgende proef krijgt weer een lege oefenmap. Jouw code blijft wel in je Webbouwers-project bewaard. Dit voorbeeld is dus nog geen blijvend gastenboek of scorebord.

Afsluiten - 10 minuten

- Leg uit wat `json_encode` en `json_decode` doen.
- Leg uit wat het verschil is tussen code bewaren en een gegevensbestand bewaren.
- Bedenk welke afspraken nodig zijn voor een echt, blijvend scorebord. Denk aan namen, toegang en verwijderen.

Inleveren

Een werkende lijst, een tijdelijke JSON-proef en je uitleg over de levensduur van beide soorten opslag.

Docentnotities en beoordeling

Kernantwoorden

- Les 1: h1 is de hoofdkop, p een alinea, ul een lijst en li een lijstpunt. CSS bepaalt onder meer kleuren en ruimte.
- Les 2: de tekst op een knop verandert niet vanzelf data-goed. De variabele punten onthoudt de score. disabled voorkomt een tweede antwoord.
- Les 3: 3/5 is 60%, 5/5 is 100%, 0/5 is 0%. Totaal 0 en goed boven totaal moeten worden afgewezen.
- Les 4: foreach herhaalt voor iedere arraywaarde. JSON zet gegevens om naar tekst en terug. De tijdelijke bestanden verdwijnen na elke proef.

Beoordelen op drie vragen

- Werkt het volgens de afgesproken testgevallen?
- Kan de leerling aanwijzen welke regels het gedrag veroorzaken?
- Kan de leerling een bewuste wijziging voorspellen, testen en uitleggen?

Ondersteuning

Laat beginners eerst alleen tekst, kleuren en invoer wijzigen. Geef daarna één ontbrekende regel. Laat sterkere leerlingen randgevallen toevoegen, functies maken en uitleggen waarom invoercontrole nodig is.

Praktische grenzen van deze basisreeks

De eigen PHP draait in een aparte tijdelijke runner. Er zijn geen schoolbestanden, externe netwerkaanroepen of blijvende leerlingbestanden beschikbaar. De voorvertoning van PHP-HTML voert geen scripts uit. De basisreeks bouwt geen volledig gekoppelde browser-app met een blijvende database.

Presenteren aan VO-scholen

Vooraf

Controleer de aparte runner en schakel alleen voor de presentatie de vrije demo in als dat gewenst is. Probeer de voorbeelden zelf. Houd een blanco demo klaar.

0:00 - 1:00 | Het uitgangspunt

Open de centrale introductie. Vertel: leerlingen gaan van maken naar begrijpen. Toon kort het onderscheid tussen PO-blokken en de VO-codewerkplaats.

1:00 - 3:00 | JavaScript zichtbaar maken

Open de VO-ruimtequiz. Voer die uit, beantwoord de vraag en toon de score. Verander één knoptekst of kleur. Wijs de klikfunctie en een comment aan.

3:00 - 5:00 | Zelf PHP schrijven

Open Zelf PHP schrijven. Laad de begroeting, verander de invoer en voer uit. Toon tekst en HTML-uitvoer. Laad daarna de score en demonstreer 3 van 5: 60%.

5:00 - 6:00 | Een fout als leermoment

Verwijder bewust een puntkomma en lees de foutmelding. Herstel de fout. Leg uit dat leerlingcode in een aparte werkplaats draait en geen toegang tot de schoolomgeving krijgt.

6:00 - 7:00 | Van demo naar onderwijs

Toon de blanco template en de vier lesbrieven. Vertel dat elke school eigen groepen, accounts en leerlingprojecten krijgt. Rond af met het centrale aanvraagformulier.

Zeg erbij

Dit is een pilotopzet. De aparte runner moet per installatie worden aangesloten en gecontroleerd. Blijvende gegevensopslag en volledig gekoppelde PHP-webapps zijn vervolgstappen.