

Hoorcollege 3

JavaScript & PHP

Informatie-uitwisseling: Web
Jeroen Ooge

JAVASCRIPT

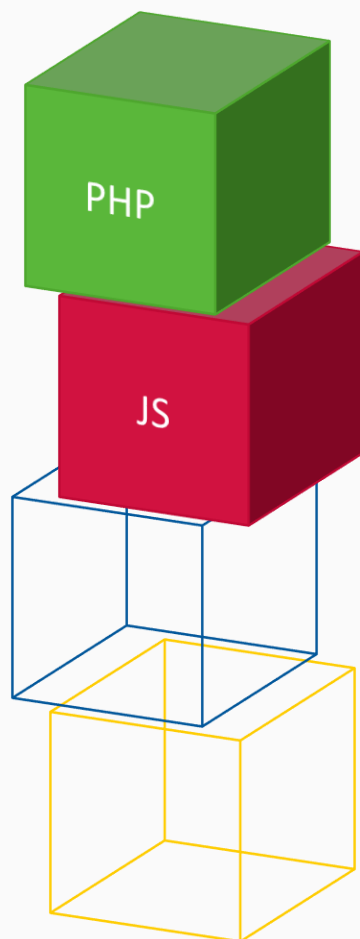
```
function  
greet(name){  
  alert('Hello,  
    + name);  
}
```

PHP

```
<?php  
$username  
= 'admin';
```



Bouwblokken voor webdesign



PHP (*Hypertext Preprocessor*)
genereert websites dynamisch

JavaScript
programmeert het gedrag van websites

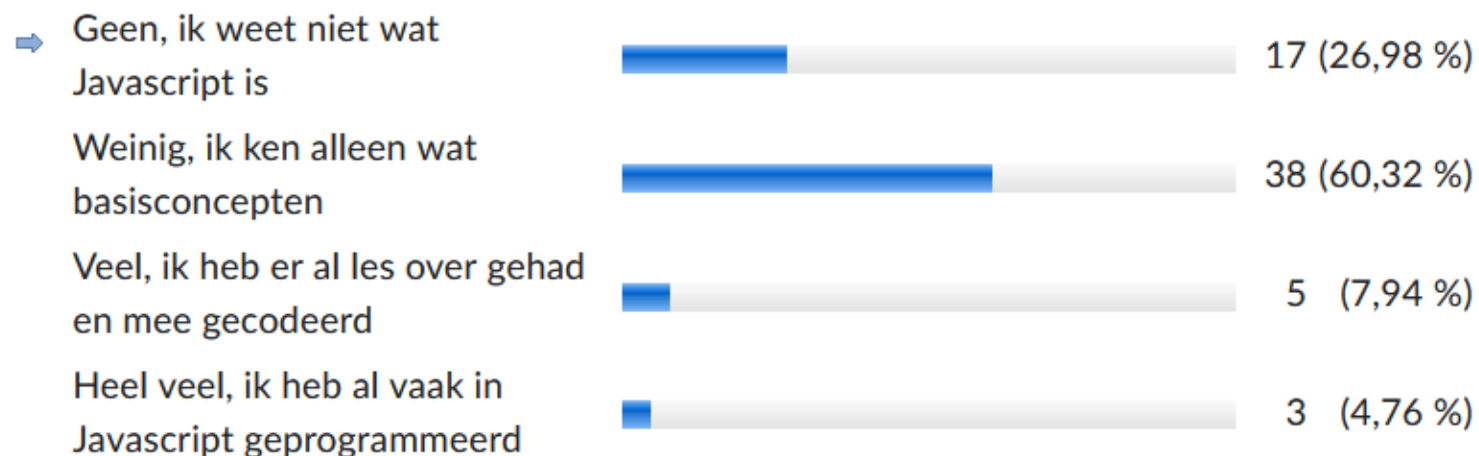
CSS (*Cascading Style Sheets*)
specifieert de opmaak van websites

HTML (*HyperText Markup Language*)
definieert de inhoud van websites

SVG (*Scalable Vector Graphics*) afbeeldingen

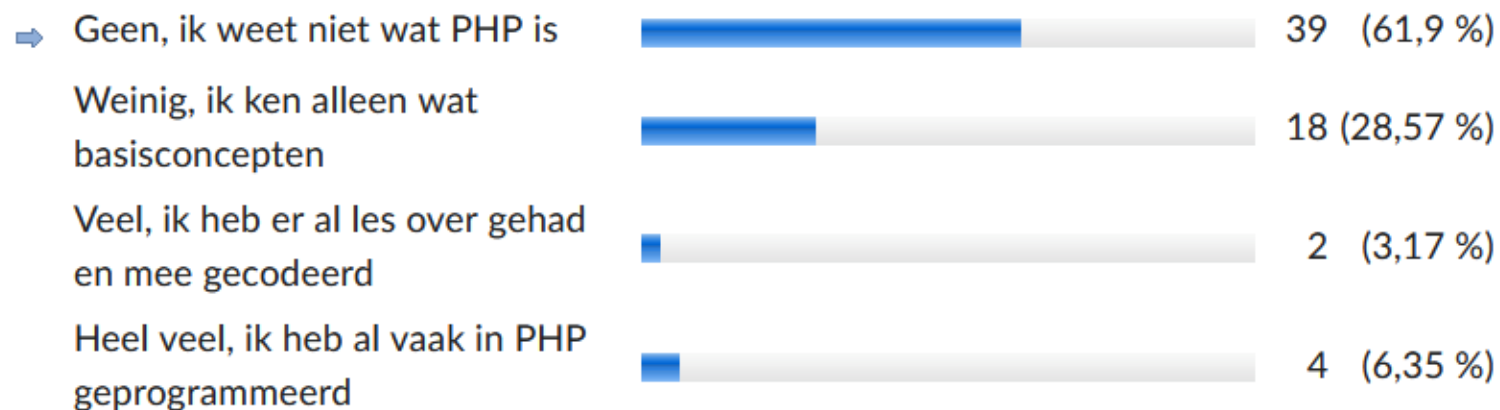


Hoeveel ervaring heb je met Javascript?



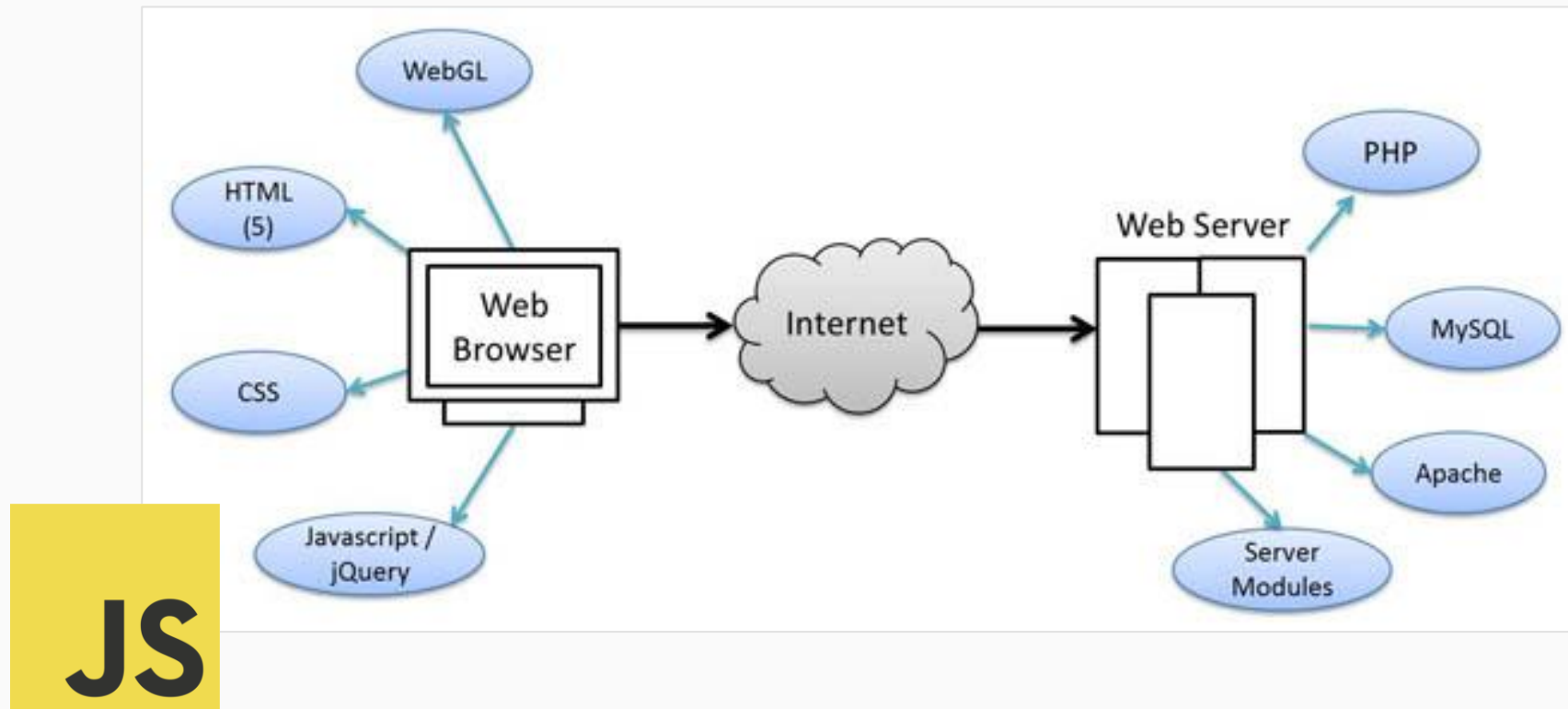
Average Grade: 0,27 / 1 (26,98 %)
 Standard Deviation: 44,39 %
 Point Biserial: 0,77
 Discrimination Index: 70,59 %

Hoeveel ervaring heb je met PHP?



Average Grade: 0,62 / 1 (61,9 %)
 Standard Deviation: 48,56 %
 Point Biserial: 0,77
 Discrimination Index: 100,00 %

Client side vs server side



Hoorcollege 3

JavaScript & PHP

1. Introductie JavaScript
2. Introductie PHP
3. Lang leve het Web

Stel gerust vragen



100 *SECONDS OF*

JS

JavaScript maakt websites interactief

JavaScript (JS) is een scripting-taal voor het Web:

- Data manipuleren en berekenen
- Reageren op events (bv. clicks, mouseovers, mousedown, submit...)
- HTML en CSS aanpassen, toevoegen en verwijderen
- ...

Basismogelijkheden van JS

Variabelen en expressies

- Types: number, string, array, object, booleans
- Operaties voor variabelen
- Assignment: waarde toekennen aan variabele (`let` en `const`)

Statements

- Conditional statement (`if`)
- Loop (`while`, `for`)

Functies en objecten

- Ingebouwd in JavaScript, bv. `console.log()`, `alert()` en `Number()`
- Zelfgedefinieerd: `function myFunction(p1, p2) { ... }`

JS toevoegen aan websites

In principe overal in HTML toegestaan,
maar best in een apart bestand

Script wordt uitgevoerd zodra de
browser het script-element tegenkomt

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>alert demo</title>
</head>
  <body>
    <script type="text/javascript">
      tekst = "alert demo";
      alert(tekst);
    </script>
  </body>
</html>
```

Niet zo elegant

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>alert demo</title>
    <script type="text/javascript">
      tekst = "alert demo";
      alert(tekst);
    </script>
  </head>
  <body/>
</html>
```

Als het moet...

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>alert demo</title>
    <script type="text/javascript" src="alert.js">
    </script>
  </head>
  <body/>
</html>
```

Beter

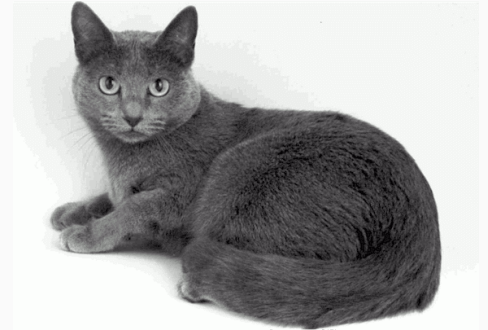
Tijd voor een demo!

Hoe kan JS interageren met HTML en CSS?

HTML DOM (Document Object Model) definieert:

- alle HTML-elementen als objecten met properties
- methodes om toegang te krijgen tot deze objecten
- events voor de objecten

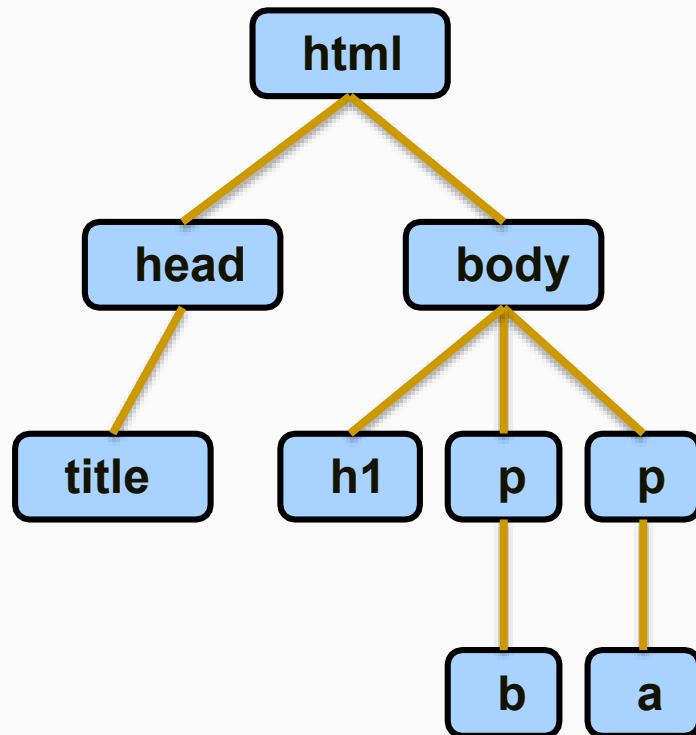
Hierdoor kun je dynamisch HTML-elementen lezen, toevoegen, verwijderen en aanpassen



Een kat is een object `o` met
eigenschap `kleur` en
methode `eten`

```
o.kleur geeft grijs  
o.eten('muis') zorgt  
dat de kat minder honger is
```

DOM-boom: geneste HTML-elementen

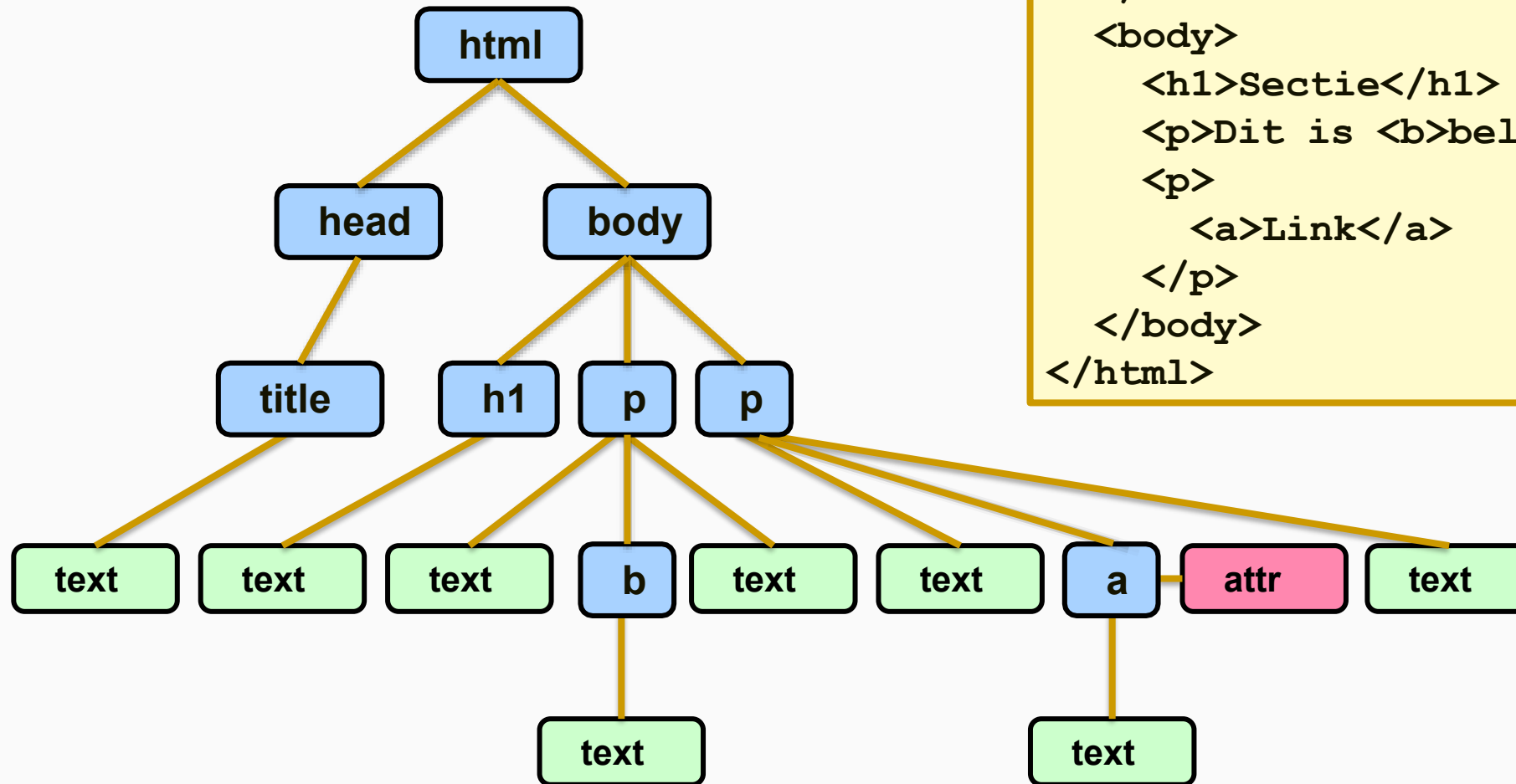


```
<html>
  <head>
    <title>Mijn titel</title>
  </head>
  <body>
    <h1>Sectie</h1>
    <p>Dit is <b>belangrijk</b>!</p>
    <p>
      <a>Link</a>
    </p>
  </body>
</html>
```



Jargon voor boomstructuren: ancestor, descendant, parent, child, sibling

De volledige DOM-boom



```
<html>
  <head>
    <title>Mijn titel</title>
  </head>
  <body>
    <h1>Sectie</h1>
    <p>Dit is <b>belangrijk</b>!</p>
    <p>
      <a>Link</a>
    </p>
  </body>
</html>
```

Werken met de DOM

HTML-element zoeken

- Zoeken op tag name
`document.getElementsByTagName("p");`
- Zoeken op ID
`document.getElementById("intro");`
- Zoeken op klasse
`document.getElementsByClassName("intro");`

Werken met de DOM

HTML-element aanpassen

- Inhoud wijzigen:
`document.getElementById(id).innerHTML = new HTML`
- Attribuut aanpassen:
`document.getElementById(id).attribute = new value`

HTML-elementen maken

Maak een nieuw element en nest het in een ander element:

```
const parent = document.getElementById("id");  
const new = document.createElement("p");  
parent.appendChild(new);
```

Events toevoegen aan HTML-elementen

Voeg een attribute toe aan een HTML-element

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

HTML

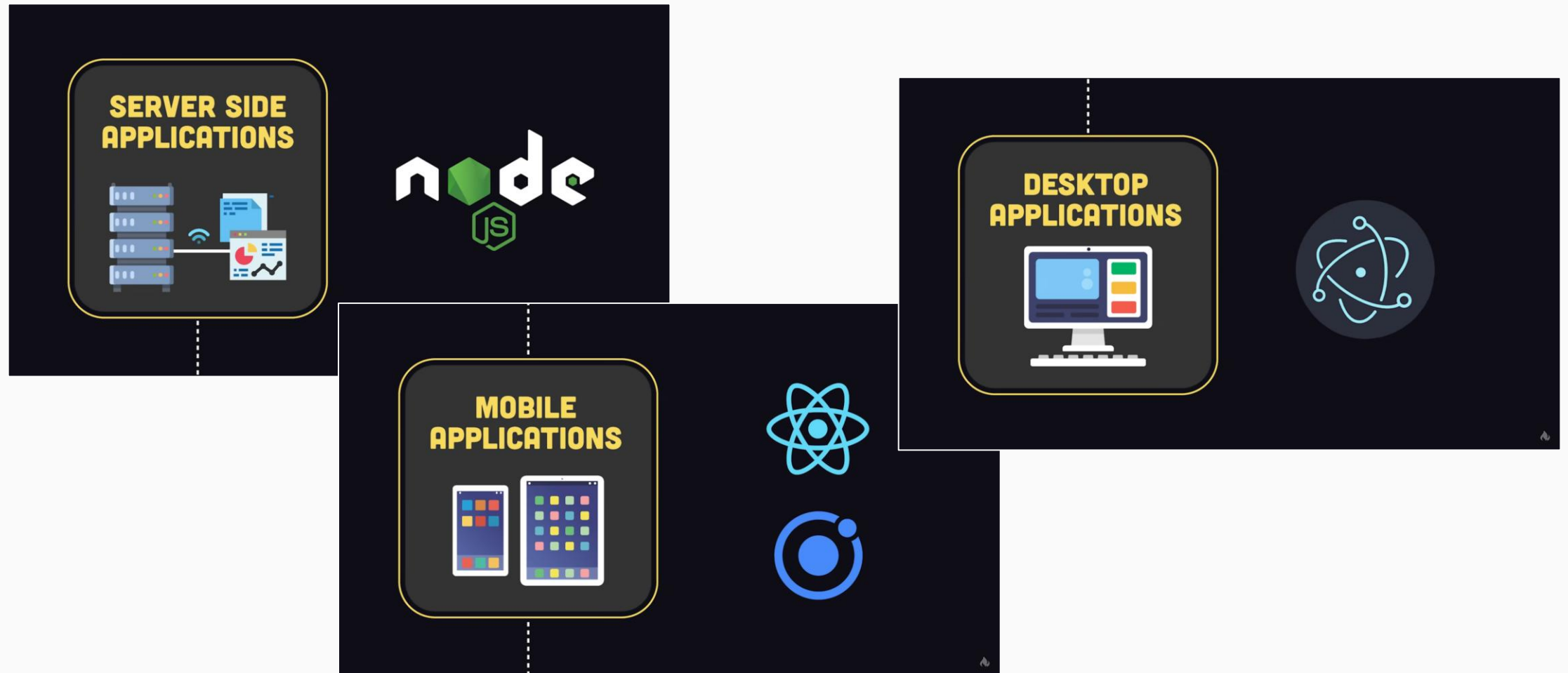
```
<h1 onclick=
  "myFunction(this)">
  Click here!
</h1>
```

Voeg een event toe via de DOM

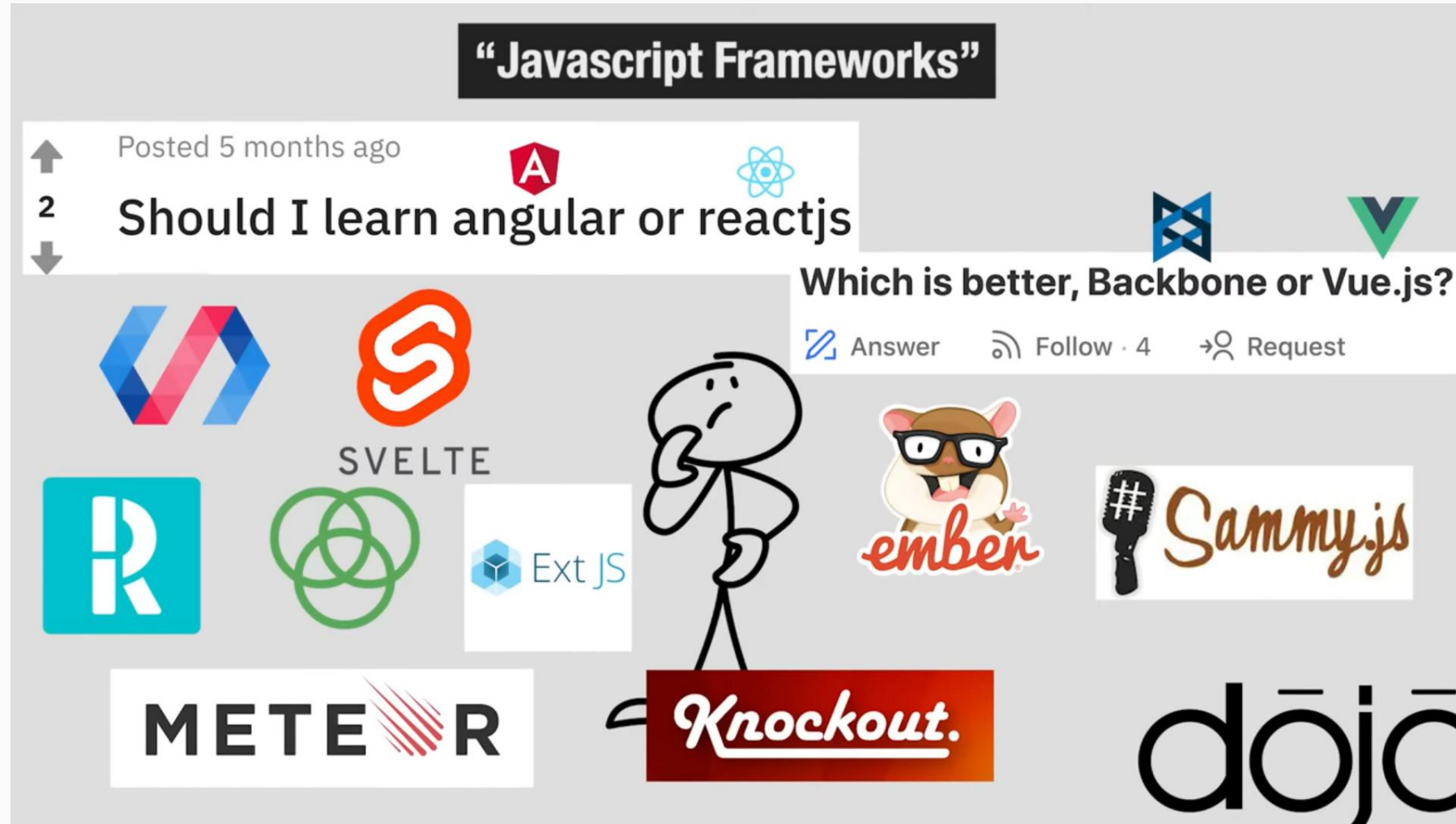
```
document.getElementById("myBtn").onclick = ... ; JS
```

Tijd voor een demo!

JS vandaag: meer dan front-end



JS vandaag: frameworks



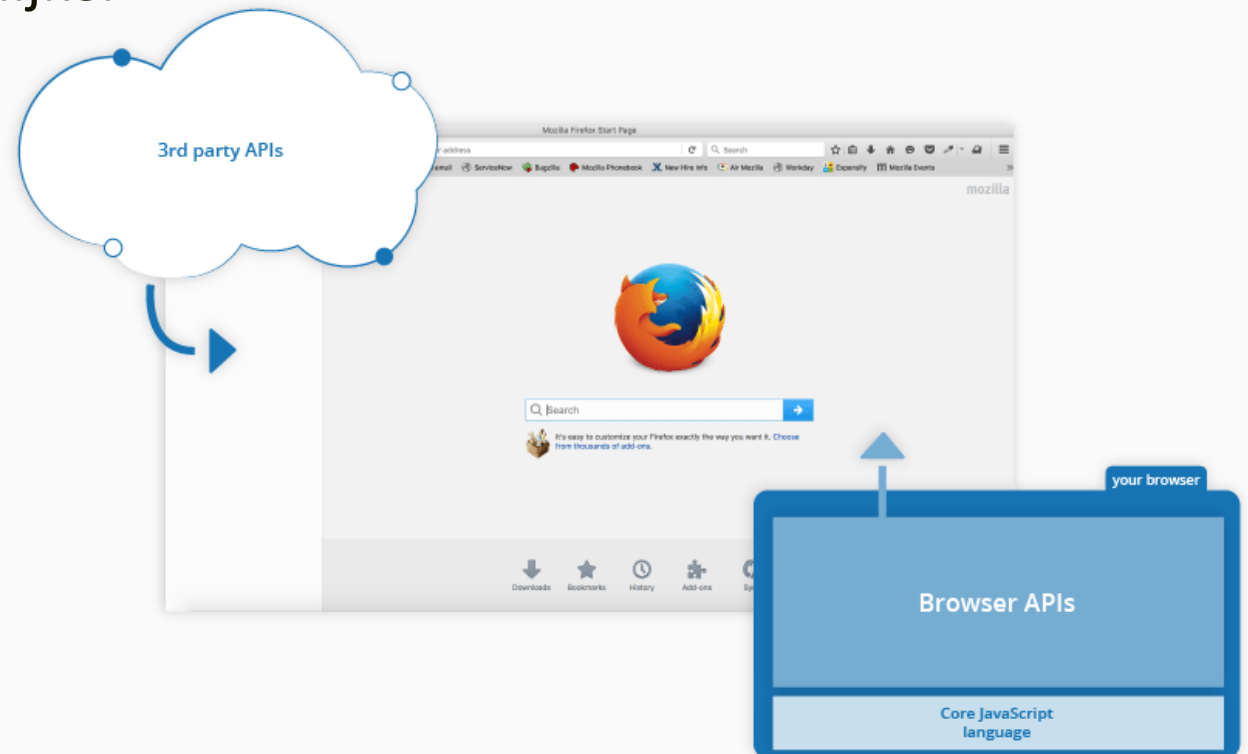
JS vandaag: Web APIs

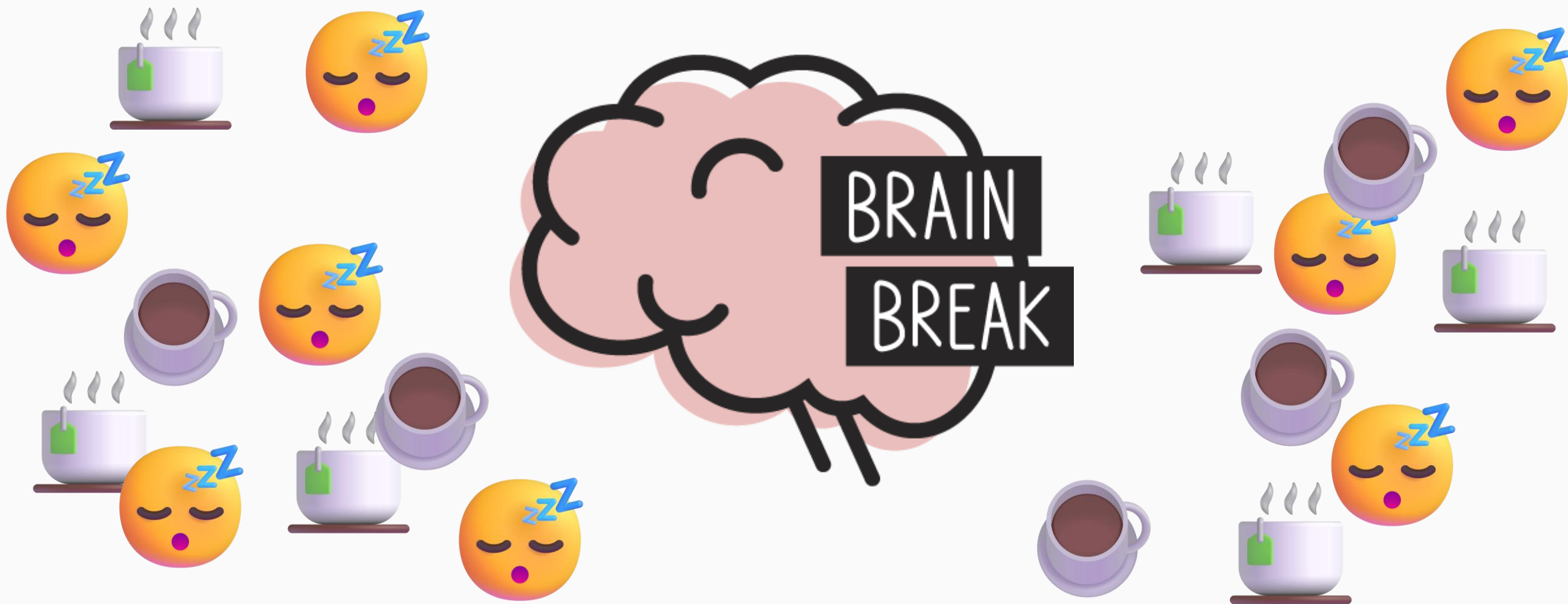
Een Web **API** (Application Programming Interface) ...

- breidt de functionaliteiten van browsers uit
- maakt complexe functies gemakkelijker

HTML DOM is een Browser API,
ingebouwd in alle browsers!

Veel derde partijen bieden APIs aan,
bv. Google Maps, YouTube,
Facebook...





Hoorcollege 3

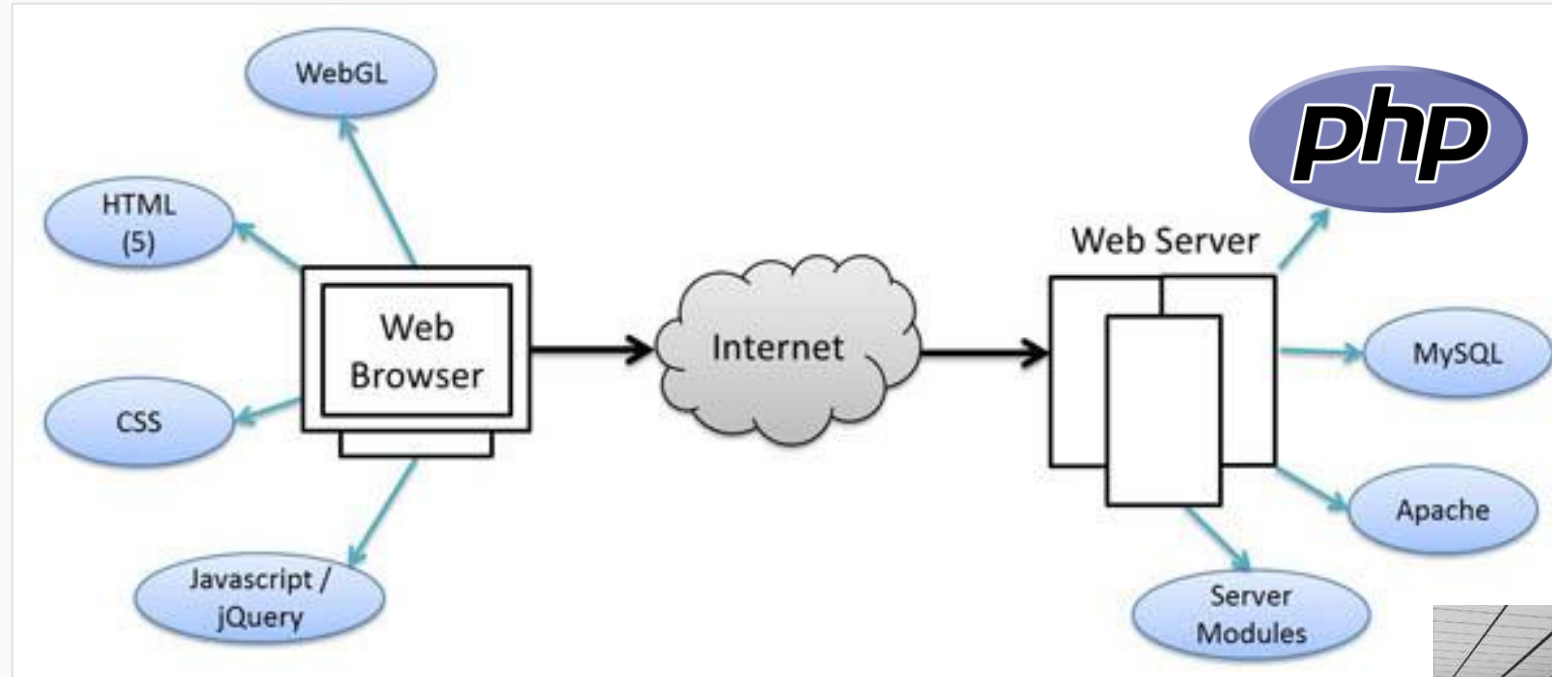
JavaScript & PHP

1. Introductie JavaScript
2. Introductie PHP
3. Lang leve het Web

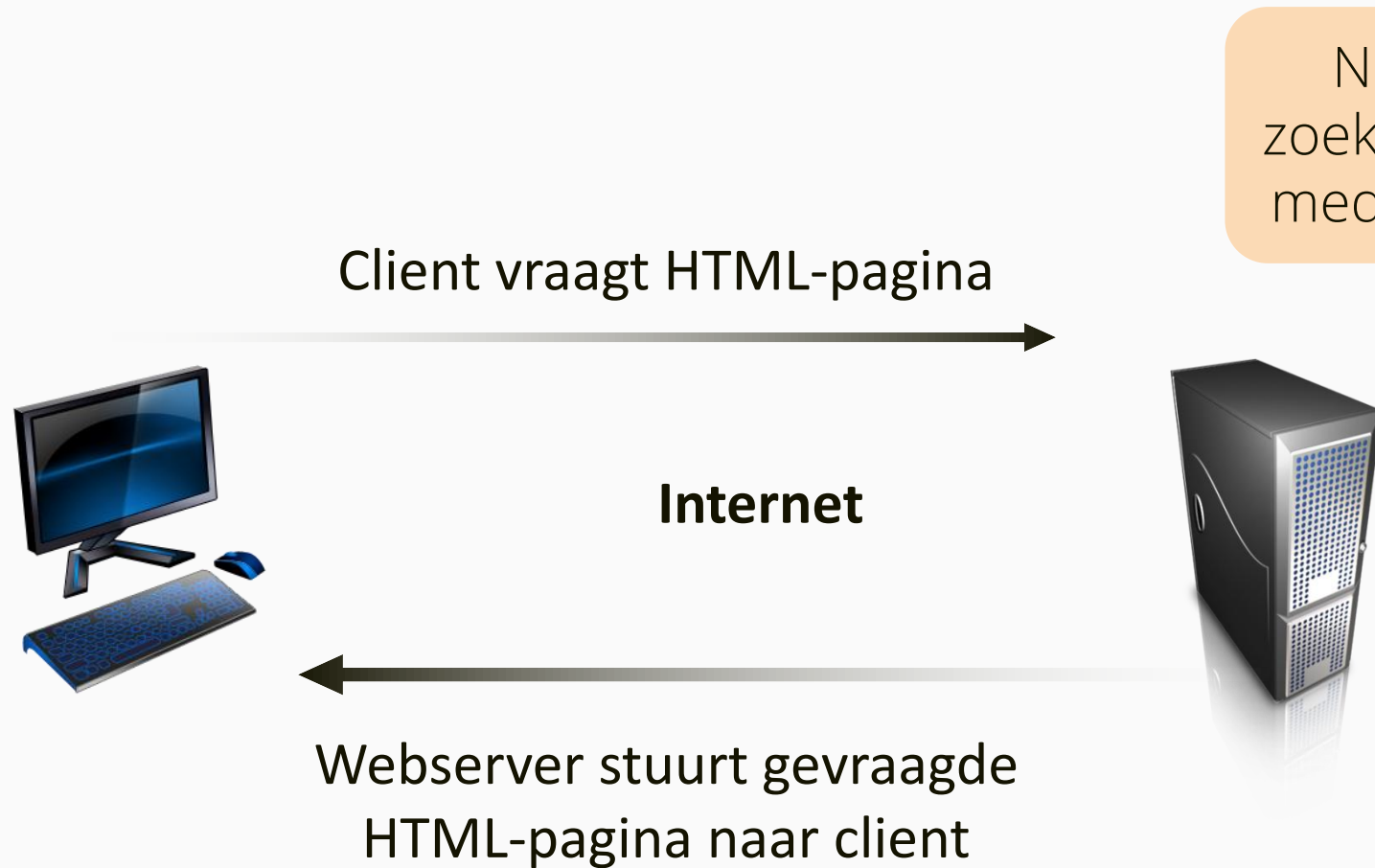
Stel gerust vragen



Client side vs server side



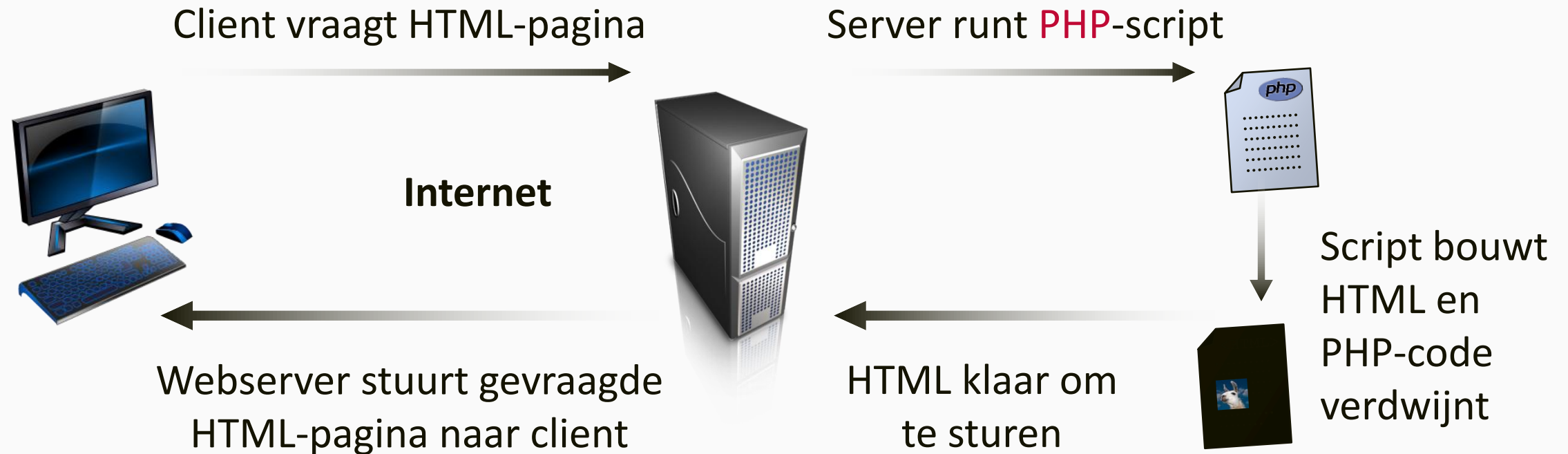
Tot 1995: statische webpagina's



Niet handig voor zoekmachines, sociale media, e-commerce...



Vanaf 1995: dynamische webpagina's mogelijk



Server side scripting

PHP (PHP: Hypertext Preprocessor):

- Open source en specifiek ontwikkeld voor web development
- Moet geïnstalleerd worden op de webserver, maar vaak ondersteund
- Zet HTML- en PHP-code om naar HTML (PHP-code wordt verwijderd)
- Nuttig voor communicatie met databases, formulieren, file management...
- Virtual server nodig!



Alternatieven: Ruby, ASP, .NET ...

Opbouw van een PHP-bestand

.php bestand op server

```
<!DOCTYPE html>
<html>
<body>

<h1>
<?php
echo "Hello World!";
?>
</h1>

</body>
</html>
```



.html bestand op client

```
<!DOCTYPE html>
<html>
<body>

<h1>Hello World!</h1>

</body>
</html>
```

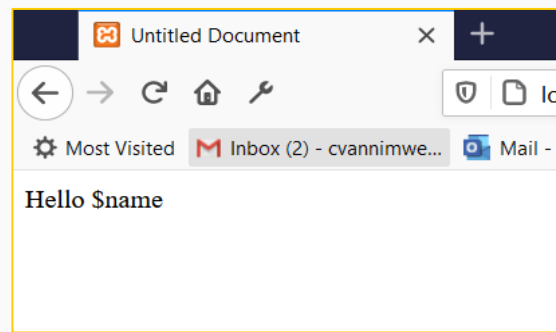
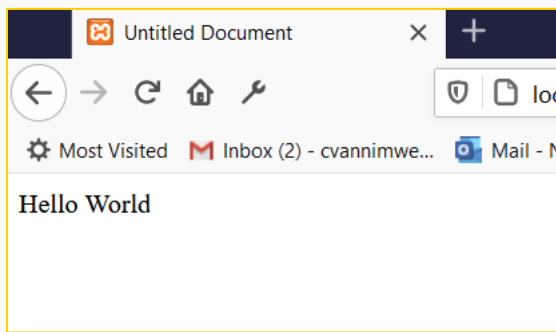


Syntax van PHP is herkenbaar

- Variabelen beginnen met \$
- Statements sluiten af met ;
- Veelgebruikelijke operatoren, functies, if

```
<?php  
$name = 'World';  
echo "Hello $name";  
?>
```

```
<?php  
$name = 'World';  
echo 'Hello $name';  
?>
```



Invoer doorgeven vanaf client

- Via HTML `<form>` element
 - action: naam PHP-pagina
 - je kunt ook hidden input fields gebruiken
 - method: POST of GET
- GET: parameters zichtbaar in URL request (data halen)
 - handig voor bookmarking, linking
- POST: onzichtbaar, meer data
 - is (wat) veiliger dan GET

POST: Invoer doorgeven vanaf client

HTML

```
<!DOCTYPE HTML>
<html>
<body>

<form action="hallo.php" method="post">
Naam: <input type="text" name="naam"><br>
Stad: <input type="text" name="stad"><br>
<input name="" type="submit" value="Ga!">
</form>

</body>
</html>
```

Naam:

Stad:

IN BROWSER



PHP

```
<?php
/* Bepaal wat de statische text wordt
en waar de dynamische data komt */

echo 'Hallo ';
echo $_POST["naam"];
echo ', woon jij echt in ';
echo $_POST["stad"];
echo '?';
?>
```

Hallo Christof , woon jij echt in Utrecht?

Files schrijven op de server

- Kan in een database
 - Veelgeziene combinatie is PHP en MySQL (ook gratis)
- Maken, lezen en schrijven in files op server
 - `fopen()`, `fclose()`, `fwrite()`
 - bv.: `fopen("test.txt", "a")` opent file om data aan einde toe te voegen ('append')
 - `r`, `r+`, `w`, `a`, `a+`
- Let bij schrijfacties op toegangsrechten op server

Files schrijven op de server

HTML

```
<!DOCTYPE HTML>
<html>
<body>

<form action="hallo.php" method="post">
Naam: <input type="text" name="naam" <br>
Stad: <input type="text" name="stad" <br>
<input name="" type="submit" value="Ga!">
</form>

</body>
</html>
```

PHP

```
<?php
$info = fopen($_POST["naam"].".txt", "a");

fwrite($info, $_POST["stad"]);
fclose($info);
?>

</body>
</html>
```

Naam:
Stad:

IN BROWSER

Christof.txt - Notepad
File Edit Format View Help
Utrecht

Christof.txt 7 Text Docu... 16-11-2014 19:54:00 -rw-r--r-- apache ap...

Hoorcollege 3

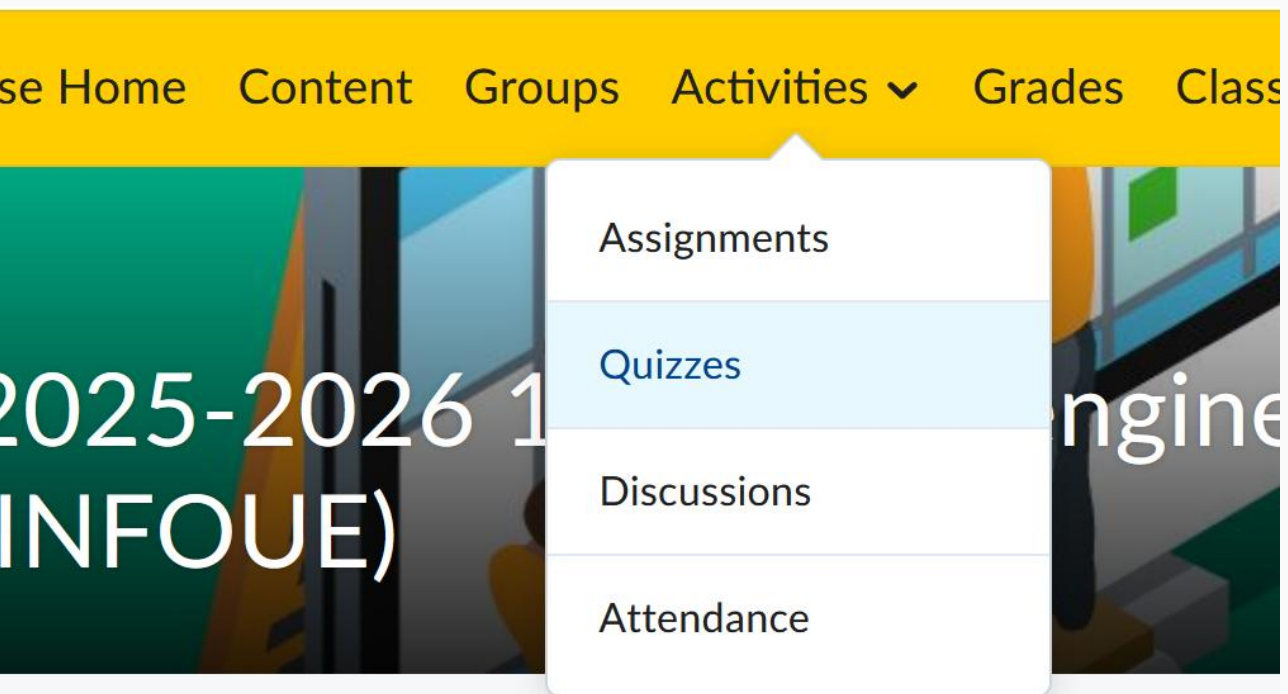
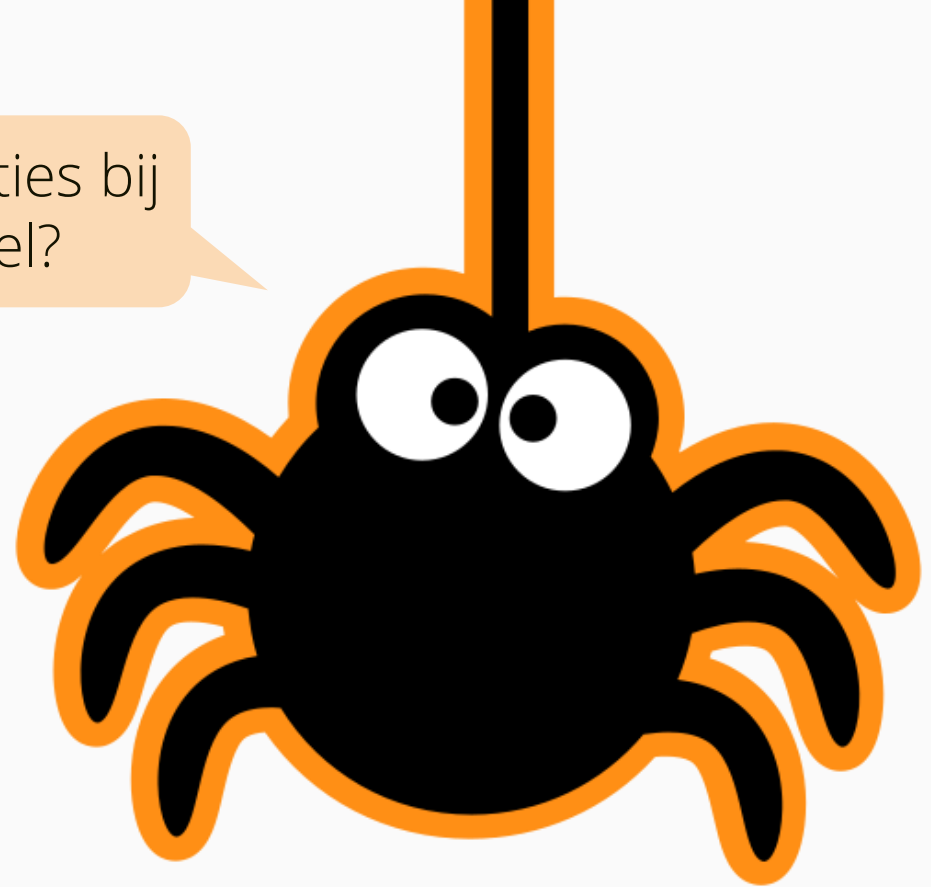
JavaScript & PHP

1. Introductie JavaScript
2. Introductie PHP
3. Lang leve het Web

Is het Web kapot?



Eerste reacties bij
de artikel?



Discussiepunten

- Hoe problematisch is een gecentraliseerd Web?
- Hoe haalbaar is een universeel Web in tijden van “AI slop”?
- Hoe veilig is grootschalige data-integratie?



Is het Web kapot?

Nieuws - 17 juli 2021 - 10:02

'Het web is kapot, hoe kunnen we het repareren?'



Het wereldwijd web is kapot. De geïnteresseerde ouders van kinderen hadden een andere insteek, maar het internet is overwoekerd door commerciële en politieke belangen, en de 'move fast and break stuff'-mentaliteit. Met negatieve gevolgen voor zaken als privacy en autonomie als gevolg. Signalen hierover

TECHZINE

Tim Berners-Lee publiceert plan om internet te repareren



NOS

Nieuws Sport



nieuwsuur

Vrijdag 8 maart 2019, 14:38

Het www is kapot, hoe gaan we het repareren?

Het internet is stuk

Maar we kunnen het repareren

Marleen Stikker

DE GEUS

Is het Web kapot?



https://www.youtube.com/watch?v=YuI6D7t_LZE

Algemene samenvatting

Bouw websites stapsgewijs

- Content met **HTML**: welke `id` en `class` waarden heb je nodig?
- Opmaak met **CSS**: maak het mooi
- Interactie met **JavaScript**: plan noodzakelijke functie(s) en zorg altijd dat je code werkt voor je verder gaat (JS is gevoelig voor bugs en errors)

Web development en scripten zijn een vak op zich

- verschillende browsers ondersteunen
- rekening houden met verschillende versies van programmeertalen
- kennis hebben over libraries, toolboxes, frameworks...

Het Web is voortdurend in ontwikkeling!

Credits

Inhoud geïnspireerd op slides van Christof van Nimwegen

<https://www.w3schools.com/js>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://www.w3schools.com/php>