

Deel 1: Multiple choice vragen

1. Wat heb je fout gedaan als je de "Headers already sent"-foutmelding krijgt?
 - ☒ A. Al output gegenereerd voor de sessie opgestart is
 - B. De head van de html-pagina niet afgesloten
 - C. Een cookie voor de sessie opgestart
 - D. Een sessie voor de cookie opgestart

2. PHP is een weak-typed language, wat bedoelt men hiermee?
 - A. Variabelen hebben een type maar hun waarde niet
 - B. Variabelen en hun waarde hebben geen type
 - C. Beide hebben een type dat pas bepaald wordt bij het uitvoeren van de app
 - ☒ D. Niet de variabelen maar hun waarde hebben een type

3. Wat is geen standaard datatype in php?
 - A. Integer
 - ☒ B. Number
 - C. Resource
 - D. Float

4. Met welke functie controleer je of een variabele ingevuld is?
 - A. is_set
 - B. String(variable) != ""
 - ☒ C. isset
 - D. empty

5. Hoe start je een sessie op?
 - ☒ A. session_start()
 - B. \$_SESSION.start()
 - C. \$_SESSION_start()
 - D. Session.start()

Opmerking

4. Vage vraag... indien ingevuld betekent dat hij gedeclareerd is, dan isset. Indien ingevuld betekent dat het niet leeg is, dan empty.

6. Met welke tekens voeg je geen commentaren aan je scripts?

- A. //
- B. /* */
- ☒ C. @
- D. Geen van allen

7. Waar worden GET-variabelen ingevuld?

- ☒ A. In de querystring van een URI
- B. In de querystring van de http-request
- C. In de body van de http-request
- D. Geen van allen

8. Hoe kunnen POST-variabelen van de cliënt naar de server gestuurd worden?

- A. Link
- ☒ B. Formulier
- C. Javascript
- D. Sessie

9. Welke methode gebruik je voor data uit formulieren en andere privacygevoelige data?

- ☒ A. POST
- B. GET

10. Hoe vragen we ~~GET~~ ^{POST} variabelen op server-side?

- A. \$POST["name"]
- B. POST["name"]
- C. POST[name]
- ☒ D. \$_POST["name"]

11. Wat heb je niet nodig om met een database te connecteren?

- A. username
- B. host
- ☒ C. query
- D. database

12. Waar kunnen sessies worden opgeslagen?

- A. In een database op de server
- B. In een tekstbestand op client
- C. In de headers van de request en response
- ☒ D. In de localStorage op de server

13. Met welke functie output je tekst naar je html-pagina?

- A. echo
- B. print
- ☒ C. allebei
- D. geen van beide

14. Met onderstaand voorbeeld:

```
if ($connect){  
  
require("db.php");  
  
}
```

Wanneer kan je de functies en variabelen uit db.php aanspreken?

- ☐ A. Altijd.
- ☒ B. Enkel als connection = true; is
- ☐ C. Nooit, require werkt enkel voor variabelen.
- ☐ D. Nooit, require werkt enkel voor functies.

Opmerkingen

12. In een database is ook mogelijk, afhankelijk van de configuratie van de server.

14.

- De docent heeft een fout gemaakt in haar slides. Require wordt NIET altijd aangeroepen... net als include gebeurt dit enkel wanneer de code bereikt wordt. Door meerdere mensen getest en bevestigd.
- Indien `$connect == true;` (i.p.v. `connection = true;`)

15. Hoe vangen we op wanneer de DOM is ingeladen met JQuery?

- A. `$(document).ready("init");`
- B. `$(document).init(ready);`
- ☒ C. `$(document).ready(init);`
- D. `$(window).init(ready);`

16. Met welke selector selecteer je alle paragrafen in een pagina?

- ☒ A. `$("p")`
- B. `$('-p')`
- C. `$("p:all");`
- D. `$(':p');`

17. Waarom laad je JQuery best in vanop een centrale server (bv `googleapis.com`)?

- ☒ A. Caching
- B. Security
- C. XSS
- D. Meer latency

18. Voor welke reden gebruiken we JQuery niet?

- A. Write less, do more
- B. Cross-browser
- C. Vereenvoudigt code
- ☒ D. Performanter

19. Met welk blok gaan we data met speciale tekens zoals accenten beschrijven in xml?

- ☒ A. `<![CDATA[inhoud]]>`
- B. `<![DATA[inhoud]]>`
- C. `<![DATA[inhoud]>`
- D. `<!--DATA[inhoud]-->`

Opmerking

15: init als function pointer?

20. Met welke JQuery functie overlopen we alle elementen in een lijst?

- A. `jQuery.each()`
- B. `jQuery.in()`
- C. `jQuery.forin()`
- D. `jQuery.forEach()`

Open vragen:

1. Beschrijf de verschillen tussen een sessie en cookie?
 - a. Sessies opgeslagen op server, cookies opgeslagen op client.
 - b. Sessievariabelen worden afgebroken op einde van de sessie; cookies kunnen afgebroken worden op einde van de sessie, maar kunnen ook voor langere tijd bewaard worden.
 - c. Sessies zijn veiliger omdat ze op de server staan en dus niet door gebruikers kunnen aangepast worden. Cookies kunnen wel door gebruikers aangepast worden.
2. Waarom valideer je formulier-data client-side en server-side?
 - a. Client-side:
 - i. Bezoeker helpen met (juiste) data juist in te voeren
 - ii. Load op de server verminderen door te verhinderen dat slecht ingevulde formulieren doorgestuurd worden.
 - iii. Nadeel: kan omzeild/uitgeschakeld worden door de bezoeker
 - b. Server-side:
 - i. Gegevens controleren op een plaats waar de bezoeker de controle niet kan uitzetten.
 - ii. Nadeel: verbruikt server resources
 - iii. Belangrijker dan client-side validatie!
3. Waarom zijn reguliere expressies beter voor validatie van gebruikersinput?
 - a. Reguliere expressies zijn een programmeerstandaard die vaak gebruikt wordt.
 - b. Ze laten toe om te controleren op patronen, iets wat zonder reguliere expressies heel moeilijk te checken zou zijn.
 - c. Laten niet alleen toe om een expressie te testen, maar om onmiddellijk ook aan te passen.