

PowerShell - Klausurvorbereitung (Übersicht + Test)

PowerShell - Klausurvorbereitung (Übersicht + Test)

Grundlagen

PowerShell ist eine lose typisierte Skriptsprache und arbeitet mit automatischer Typkonvertierung.

Das bedeutet:

- Datentypen werden automatisch angepasst
- keine feste Typdefinition notwendig

Begriffe:

- Deklaration → Variable erstellen
 - Initialisierung → Wert zuweisen
 - Konvertierung → Datentyp ändern
-

Variablen

- beginnen immer mit `$`
- erlaubte Zeichen: Buchstaben, Zahlen, `_`
- Leerzeichen nur mit `{ }` möglich

Beispiele:

```
$name
```

```
$_abc
```

```
$123
```

```
${mein Wert}
```

Ungültig:

```
$abc-12
```

```
$mein Name
```

Geschützte Variablen:

```
$true, $false, $_
```

Datentypen

- String → "Text"
- Int → 12
- Double → 4.5
- Boolean → \$true / \$false
- DateTime → Get-Date

Wichtig für IHK:

- Read-Host liefert IMMER einen String → oft Konvertierung nötig

Typ anzeigen:

```
$variable.GetType().Name
```

Beispiel:

```
[int]$zahl = Read-Host
```

Cmdlets

- Ausgabe → Write-Host
 - Eingabe → Read-Host
 - Datum → Get-Date
 - Konsole löschen → Clear-Host (Alias: cls)
-

Rechnen

```
(4 + 4) * 12
```

```
12 - 3
```

```
16 * 4
```

Wichtig:

- Punktrechnung vor Strichrechnung

Im String (Unterausdruck):

```
"Ergebnis: $(4*12)"
```

Datum:

```
Get-Date("2026-04-14")
```

Operatoren

Vergleich

```
-eq -ne -gt -lt -ge -le
```

```
-ceq → case sensitive
```

Beispiel:

```
12 -gt 9 → true
```

Rechenoperatoren

```
+ - * / %
```

```
% = Modulo (Rest)
```

Zuweisung

```
= → Wert setzen
```

```
+= → addieren und speichern
```

Beispiel:

```
$x = 5
```

```
$x += 3 → Ergebnis: 8
```

Stringoperatoren

```
-like → Wildcards
```

```
* → beliebig viele Zeichen
```

```
? → genau ein Zeichen
```

Beispiel:

```
"Apfel" -like "A*"
```

```
-match → Teilstring / Regex
```

Beispiel:

```
"Apfel" -match "pf" → true
```

Listenoperatoren

`-contains` → Liste links

```
"A","B","C" -contains "A"
```

`-in` → Liste rechts

```
"A" -in "A","B","C"
```

Wichtig:

- nur ganze Werte (keine Teilstrings!)
-

Logische Operatoren

`-and` → UND

`-or` → ODER

`-not` → NICHT

Unäre Operatoren

`++` → erhöhen

`--` → verringern

Verzweigungen

if / else

```
if($zahl -lt 12){...} elseif(...) {...} else {...}
```

Wichtig:

- Bedingungen müssen TRUE oder FALSE ergeben
-

switch

```
switch($wert){ {$_ -lt 12}{"kleiner 12"; break} default{"größer"} }
```

Wichtig:

- `$_` steht für aktuellen Wert
- `break` beendet den Block

TEST

1. Was bedeutet lose Typisierung?

Antwort anzeigen

Variablen haben keinen festen Datentyp und werden automatisch konvertiert.

2. Unterschied zwischen Deklaration und Initialisierung?

Antwort anzeigen

Deklaration = Variable erstellen
Initialisierung = Wert zuweisen

3. Woran erkennt man Variablen?

Antwort anzeigen

Am `$` Zeichen.

4. Was ist bei `Read-Host` wichtig?

Antwort anzeigen

Es liefert immer einen String.

5. Wofür wird `Get-Date` verwendet?

Antwort anzeigen

Für Datum und Uhrzeit.

6. Was macht `Clear-Host`?

Antwort anzeigen

Löscht die Konsole.

7. Ergebnis von $(4+4)*12$?

Antwort anzeigen

96

8. Wofür steht `-eq`?

Antwort anzeigen

Vergleich auf Gleichheit.

9. Unterschied `-eq` und `-ceq`?

Antwort anzeigen

`-ceq` beachtet Groß-/Kleinschreibung.

10. Was macht `-like`?

Antwort anzeigen

Vergleich mit Wildcards.

11. Bedeutung von `*` bei `-like`?

Antwort anzeigen

Beliebig viele Zeichen.

12. Bedeutung von `?` bei `-like`?

Antwort anzeigen

Genau ein Zeichen.

13. Was macht `-match`?

Antwort anzeigen

Teilstring-Suche.

14. Unterschied `-contains` und `-in`?

Antwort anzeigen

`-contains` → Liste links

`-in` → Liste rechts

15. Was macht `+=`?

Antwort anzeigen

Addiert und speichert den Wert.

16. Was machen `++` und `--`?

Antwort anzeigen

Erhöhen bzw. verringern um 1.

17. Wofür wird `if` verwendet?

Antwort anzeigen

Für Bedingungen.

18. Wann wird `else` ausgeführt?

Antwort anzeigen

Wenn keine Bedingung zutrifft.

19. Wofür wird `switch` verwendet?

Antwort anzeigen

Für mehrere Bedingungen.

20. Was macht `break` im `switch`?

Antwort anzeigen

Beendet den Block.

Selbstbewertung

0-7 richtig → wiederholen

8-14 richtig → unsicher

15-18 richtig → gut

19-20 richtig → bereit für die Klausur

Revision #2

Created 15 April 2026 12:12:22 by Admin

Updated 19 May 2026 07:15:51 by Admin