

Ερωτήσεις για να διακρίνουμε το scope(Εμβέλεια)

1. Πού δηλώθηκε η μεταβλητή;

- **Global Scope:** Αν η μεταβλητή δηλώθηκε εκτός από οποιαδήποτε συνάρτηση ή block (π.χ., εκτός από ένα `if`, `for`, `while`).
- **Function Scope:** Αν η μεταβλητή δηλώθηκε μέσα σε μια συνάρτηση.
- **Block Scope:** Αν η μεταβλητή δηλώθηκε μέσα σε ένα block με `let` ή `const`.

2. Πώς έχει δηλωθεί η μεταβλητή;

- **var:** Δημιουργεί μεταβλητές με function scope, ακόμη κι αν δηλωθούν σε block.
- **let/const:** Δημιουργούν μεταβλητές με block scope. Είναι ορατές μόνο μέσα στο block που δηλώνονται.

3. Μπορώ να έχω πρόσβαση στη μεταβλητή εκτός του block ή της συνάρτησης που την περιλαμβάνει;

- Αν η απάντηση είναι "ναι", τότε πιθανότατα έχει global scope ή ανήκει σε ένα ανώτερο function scope.
- Αν η απάντηση είναι "όχι", τότε έχει block scope ή function scope.

4. Αν η μεταβλητή δηλωθεί δύο φορές, ποια κυριαρχεί;

- **var:** Μπορεί να δηλωθεί δύο φορές στην ίδια λειτουργία και η δεύτερη δήλωση αντικαθιστά την πρώτη.
- **let/const:** Δεν επιτρέπουν πολλαπλές δηλώσεις της ίδιας μεταβλητής στο ίδιο scope.

5. Χρειάζεται η μεταβλητή να είναι διαθέσιμη σε άλλα μέρη του κώδικα;

- Αν ναι, σκέψου αν χρειάζεσαι global ή function scope.
- Αν όχι, προτίμησε block scope για να περιορίσεις τις παρενέργειες.

6. Είναι η μεταβλητή hoisted;

- Οι μεταβλητές δηλωμένες με `var` "ανυψώνονται" στην αρχή της λειτουργίας τους (hoisting), ενώ οι `let` και `const` όχι.

7. Η συνάρτηση που αναζητάς έχει δηλωθεί με `function` ή `arrow function` ;

- Οι παραδοσιακές συναρτήσεις δηλώνονται σε function scope, ενώ οι arrow functions ακολουθούν τα block scopes των `let` και `const`.

Με αυτές τις ερωτήσεις, μπορείς να εξετάζεις τον κώδικα και να εντοπίζεις εύκολα τι scope έχει κάθε μεταβλητή ή συνάρτηση σε ένα πρόγραμμα JavaScript.