**Function Closure**

JavaScript xarici funksiya daxilində funksiya yazmağa imkan verir. İstədiyimiz qədər daxili funksiyaları yaza bilərik. Əgər daxili funksiya xarici funksiyanın dəyişənlərinə daxil olursa, buna closure deyilir.

**Closure necə işləyir?**

1. **İç içə funksiyalar:** Bir funksiyanın daxilində başqa bir funksiyanın yaradılması.
2. **Daxili funksiyanın mühitə daxil olması:** Daxili funksiya, xarici funksiyanın mühitindəki dəyişənlərə daxil ola bilir, hətta xarici funksiyanın icra olunduğu mühitdən çıxsa belə.

**Closure-un Xüsusiyyətləri:**

1. **Davamlılıq:** Daxili funksiya yaradıldığı mühitdəki dəyişənlərə daxil olma qabiliyyətini saxlayır.
2. **Məlumatın gizlədilməsi:** Xarici funksiyanın dəyişənləri birbaşa əlçatmazdır, yalnız daxili funksiyalar tərəfindən istifadə edilə bilər.
3. **Funksional vəziyyət:** Funksiya daxilindəki məlumatların davamlı olaraq qorunmasını təmin edir.

**Closure-un Faydaları:**

1. **Məlumatların qorunması:** Dəyişənlər birbaşa əlçatmazdır və yalnız müəyyən metodlar vasitəsilə çıxış təmin edilir.

**Closure-un Əsas Prinsipləri**

1. **Lexical Environment (Leksik Mühit):** Closure, funksiyanın yaradıldığı mühitdəki dəyişənlərə daxil olmağa imkan verir. Hər dəfə bir funksiya yaradıldıqda, onun öz leksik mühiti yaradılır. Bu mühitdəki dəyişənlər funksiya ilə bağlı olur və həmin funksiya tərəfindən istifadə edilə bilər.
2. **Daxili Funksiyalar:** Closure yalnız bir funksiyanın daxilində yaradılmış daxili funksiyalar üçün tətbiq olunur. Xarici funksiyanın mühitindəki dəyişənlərə daxil olmaq üçün daxili funksiyadan istifadə edilir.
3. **Əlçatmazlıq:** Xarici funksiyanın dəyişənləri birbaşa əlçatmazdır, amma daxili funksiyalar bu dəyişənlərə daxil olmaq və onlarla işləmək qabiliyyətinə malikdir.

**Closure-un İstifadə Sahələri**

1. **Məlumatların Gizlədilməsi:** Closure-lar, bir funksiyanın daxilində yaradılan dəyişənləri xarici koddan gizlətmək üçün istifadə olunur. Bu, məlumatların qorunmasına və təhlükəsizlik məsələlərinə kömək edir.
2. **Funksional Vəziyyətin Yaradılması:** Closure-lar, bir funksiyanın daxilində yaradılan vəziyyəti qorumağa kömək edir. Məsələn, bir sayğacın vəziyyətini izləmək üçün closure istifadə edilə bilər.
3. **Factory Functions (Fabrika Funksiyaları):** Closure-lar funksiyaların və obyektlərin yaradılması üçün faydalıdır. Belə funksiyalar öz mühitlərini saxlayaraq müxtəlif instansiyalar yarada bilər.
4. **Event Handlers və Asinxron Proqramlaşdırma:** Closure-lar event handlers (tədbir idarəediciləri) və asinxron əməliyyatlarda vəziyyətin qorunmasına kömək edir.