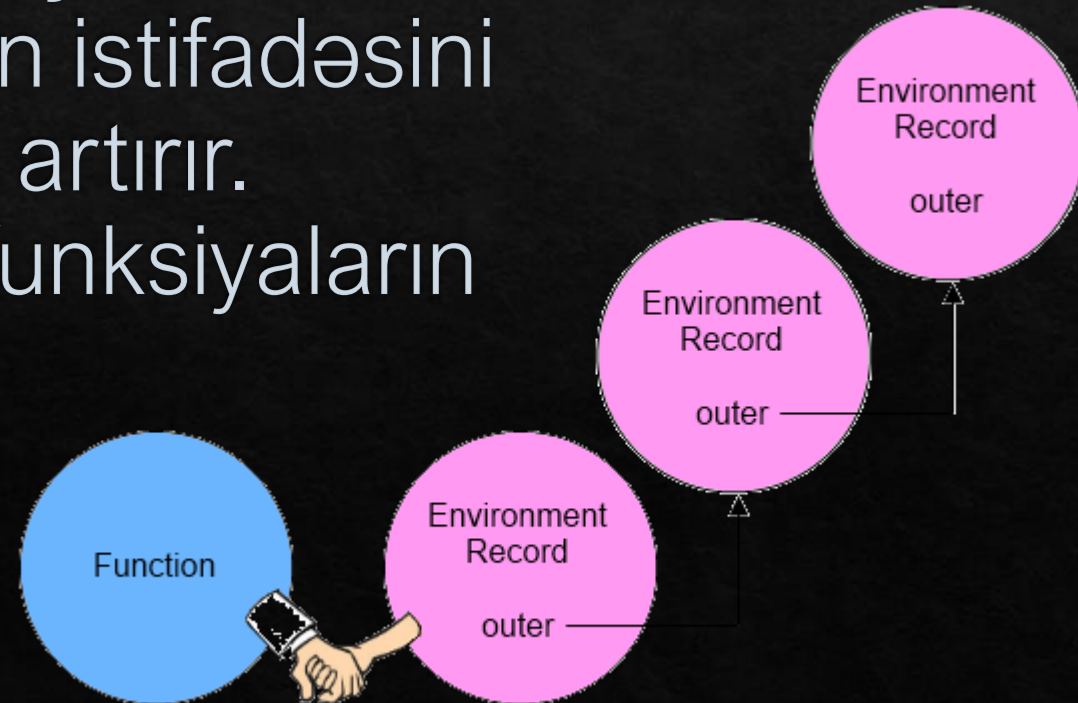


**CLOSURE
NƏDİR?**

JavaScript-də hər dəfə bir funksiya digər bir funksiyanın daxilində yaradıldıqda, CLOSURE adlanan bağlılıq yaranır. Closure elə bir funksiya ki, xarici funksiya çağrıldıqdan sonra belə onun daxilindəki dəyişənlərə istinadını saxlayır. Başqa sözlə, daxili funksiya üçün xarici funksiyanın mühitinə çıxış imkanı verir.

Closure- nin əhəmiyyəti

- ✓ Daxili məlumatların qorunması: Xarici mühitdən dəyişməyə imkan vermədən iç funksiyaların məlumatlarına giriş təmin edir.
- ✓ Təmizləmə: Qlobal dəyişənlərin istifadəsini azaldır və kodun oxunaqlılığını artırır.
- ✓ Funksiyalar: Xüsusi və unikal funksiyaların yaradılmasına imkan verir.



```
function myCustomFunction() {  
  let myAge = 27;  
  return function () {  
    myAge += 1;  
    return myAge;  
  };  
}  
  
const myAgeCounter = myCustomFunction();  
  
console.log(`Bu il mənim ${myAgeCounter()} yaşım olacaq`);  
  
// Çıxış: Bu il mənim 28 yaşım olacaq  
  
console.log(`Gələn il mənim ${myAgeCounter()} yaşım olacaq`);  
  
// Çıxış: Gələn il mənim 29 yaşım olacaq
```

Closure məntiqi ilə hər dəfə **myAgeCounter** adlı funksiya çağırıldıqda onun istinad etdiyi **myCustomFunction** funksiyaının daxilindəki **myAge** dəyişəninə əlçatanlıq təmin olunur. Beləliklə hər dəfə funksiyanın nəticəsi geri qaytarılır