Yorumlayıcı ne yapar?



www.yzokul.com.tr

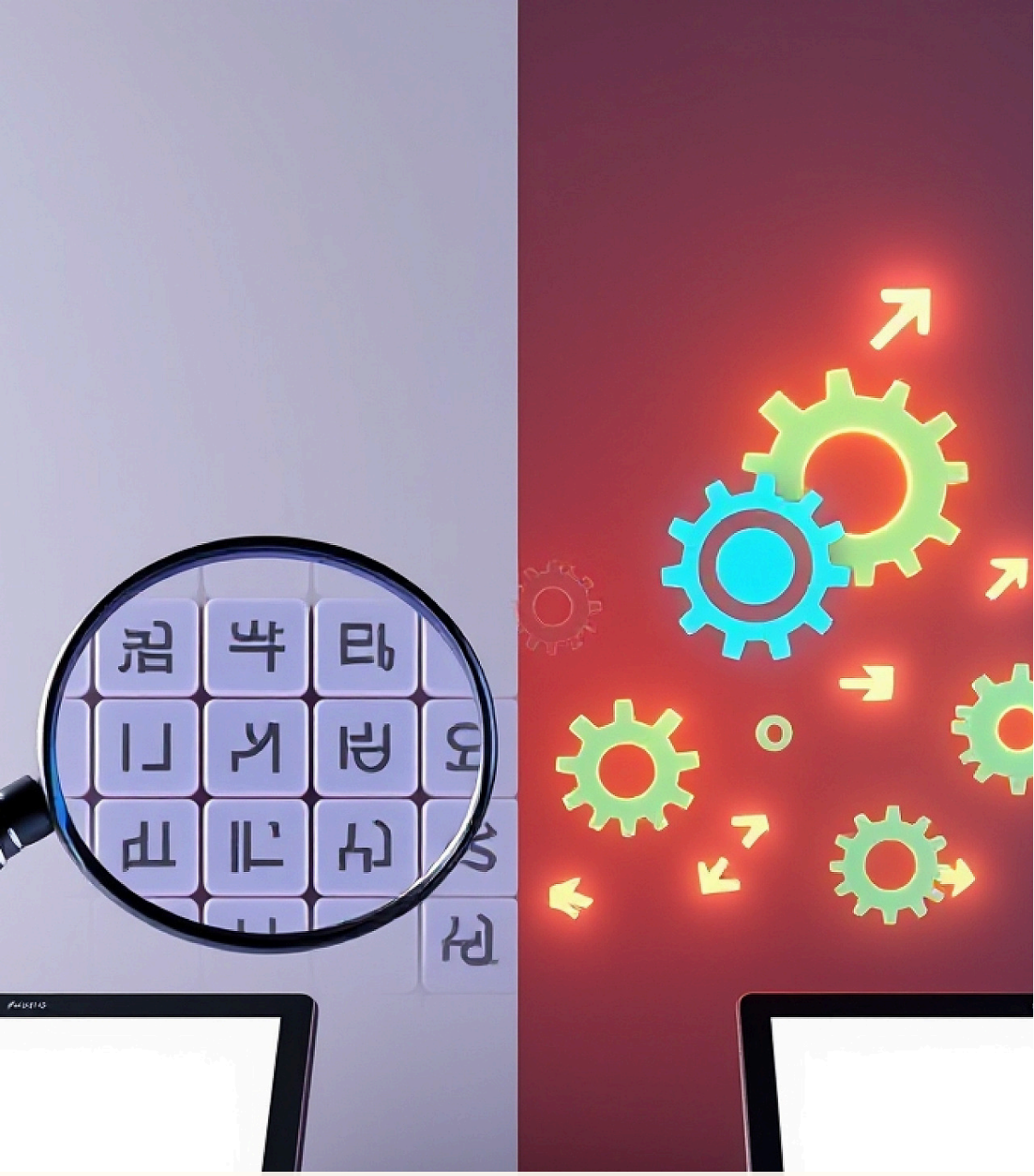
Yorumlayıcı ne yapar?



Yorumlayıcı ne yapar? Yorumlayıcı, bir programın işlemlerini program çalışırken gerçekleştiren bir sistemdir. Talimatları alır ve onları yerine getirir. Örneğin, bir Python betiği çalıştırdığında yorumlayıcı her işlemi yürütür. Kısa yanıt: yorumlayıcı, bir programın işlemlerini çalışma sırasında gerçekleştiren bir sistemdir.



Bir program yalnızca bir talimatlar listesidir, tıpkı bir yemek tarifi gibi. Her talimat bilgisayara bir şey yapmasını söyler; örneğin iki sayıyı toplamasını ya da metin göstermesini. Bir şeylerin gerçekten olması için birinin ya da bir şeyin bu talimatları izlemesi gerekir. Örneğin, bir kek tarifi sana adımları söyler ama adımları sen yapmalısın. Yorumlayıcı, bir program için adımları yapan şeydir.



Talimatları okumak, onları yerine getirmekle aynı şey değildir. Yemek yapmadan bir tarifi okuyabilirsin. Yorumlayıcı yalnızca okumayı değil, yerine getirmeyi yapar. Örneğin, bir programın kodunu okumak bir tarifi okumaya benzer; yorumlayıcı ise yemeği gerçekten yapan aşçı gibidir. Programın tanımladığı eylemleri gerçekleştirir.



Yürütme, programın tanımladığı her işlemi program çalışırken birbiri ardına gerçekleştirmek demektir. Yorumlayıcı bu yürütmeyi yapar. Örneğin, program bir sözcük yazdırmayı söylüyorsa yorumlayıcı sözcüğün ekranda görünmesini sağlar. Program sayıları toplamayı söylüyorsa yorumlayıcı toplamayı yapar. Yürütme, önceden değil, çalışma sırasında gerçekleşir.



Yorumlayıcıyı, programın talimatlarını alıp yerine getiren bir işçi olarak düşün. İşçi, çalışma sırasında harekete geçer, önceden değil. Örneğin, bir planı izleyen bir inşaatçıyı düşün: inşaatçı işi inşa ederken yapar. Yorumlayıcı da o inşaatçı gibidir; program çalışırken her işlemi yapar. Yalnızca plan yapmaz; yapar.



Bir tür yorumlayıcı, programcının yazdığı özgün kaynak kodu alır. Kaynak kod, programın bir programlama dilindeki metnidir. Yorumlayıcı doğrudan bu metinden çalışır. Örneğin, Python gibi bir dil için basit bir yorumlayıcı kaynak kodu okur ve yürütür. Önce ayrı bir dönüştürme adımına ihtiyaç duymaz.



Başka bir tür yorumlayıcı ise önce daha basit bir ara forma dönüştürülmüş kodu alır. Ara kod, programın yorumlayıcının yürütmesi için daha kolay olan bir sürümüdür. Yorumlayıcı daha sonra bu ara kodu yürütür. Örneğin, bazı sistemler kaynak kodu bir tür ara kod olan bayt koduna dönüştürür ve sonra bayt kodunu yorumlar.



Yaygın bir inanç, yorumlayıcıların her zaman tek seferde bir satır okuduğudur. Bu tanım tüm yorumlayıcılar için doğru değildir. Bazı yorumlayıcılar daha büyük parçaları ya da tüm programı bir seferde işler. Örneğin, bir yorumlayıcı yürütmeden önce bütün bir işlevi okuyabilir. Yani satır satır okumanın yorumlayıcının tanımı olduğunu varsayma.



İnsanlar satır satır fikrine inanır çünkü bazı basit yorumlayıcılar bu şekilde çalışır. Örneğin, bir öğretim dili için temel bir yorumlayıcı tek seferde bir satır okuyup yürütebilir. Ancak bu yalnızca olası bir tasarımdır, tanım değildir. Birçok modern yorumlayıcı farklı çalışır. Önemli olan, kodu nasıl okursa okusun, işlemleri çalışma sırasında gerçekleştirmeleridir.



Bazı sistemler önce kodu ara koda dönüştürür ve sonra yürütür. Dönüştürme, işlemler yerine getirilmeden önce gerçekleşir. Örneğin, bir sistem kaynak kodu alıp daha basit bir forma çevirebilir, sonra bir yorumlayıcı o formu yürütür. Bu iki adımlı yaklaşım yürütmeyi daha hızlı ya da daha taşınabilir kılabilir. Yürütmeyi yine yorumlayıcı yapar.



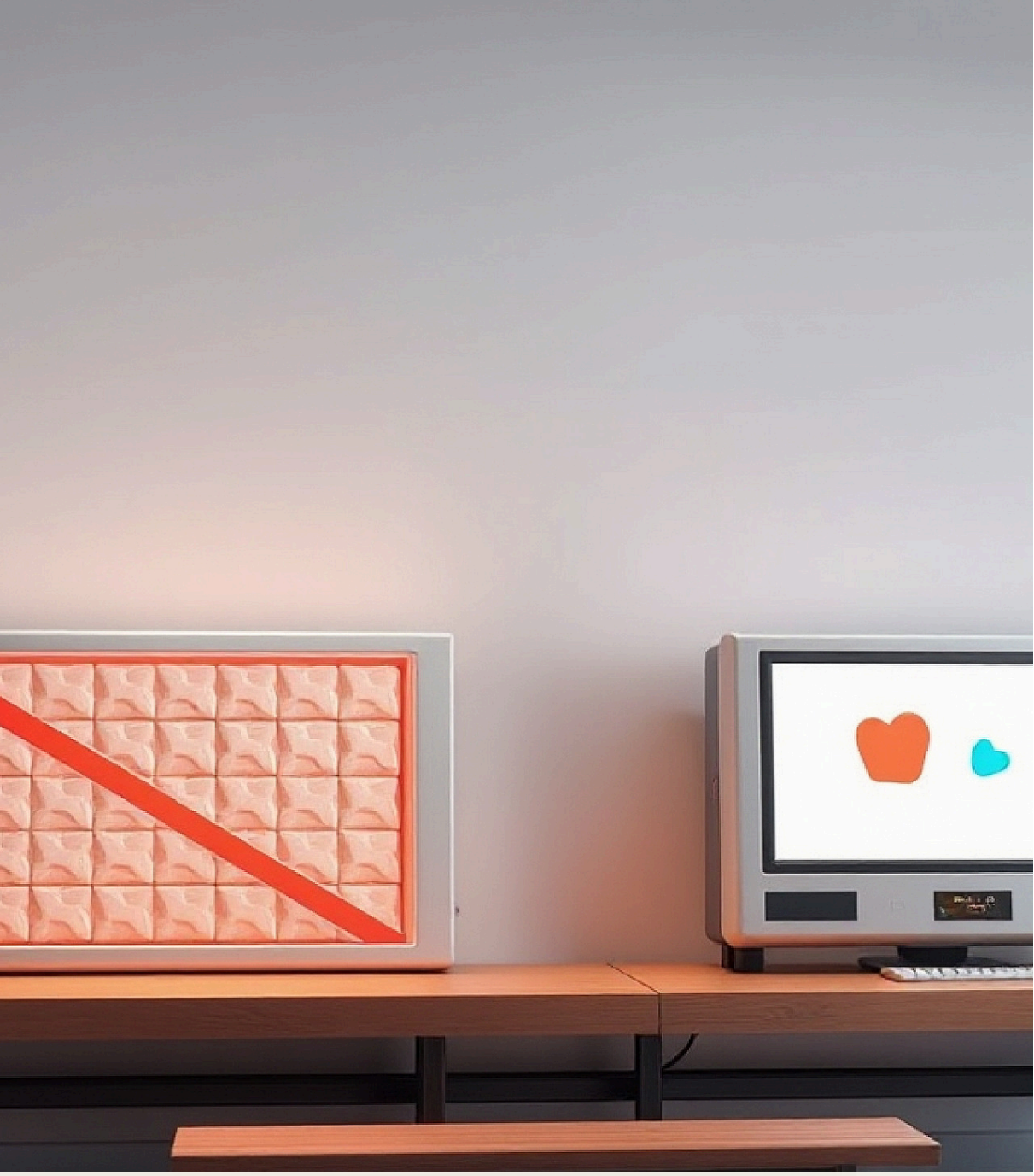
Derleme, kodu önceden başka bir forma dönüştürmek demektir. Bu, işlemleri hemen yerine getirmekten farklıdır. Örneğin, bir derleyici kaynak kodu bilgisayarın daha sonra doğrudan çalıştırabileceği makine koduna çevirebilir. Derleyici programı yürütmez; onu hazırlar. Yorumlayıcı ise bunun aksine programın işlemlerini yürütür.



Modern dil uygulamalarında yorumlama ve derleme birlikte kullanılabilir. Bir sistem kaynak kodu ara koda (bir tür derleme) dönüştürebilir ve sonra bu ara kodu yorumlayabilir. Örneğin, birçok dil çalışma ortamı bayt koduna derler ve sonra bayt kodunu yorumlar. Yani iki iş tek bir sistemde birleştirilebilir.



Girdi nasıl hazırlanırsa hazırlansın, yorumlayıcının işi aynı kalır: programın işlemlerini çalışma sırasında gerçekleştirmek. İster kaynak koddan ister ara koddan çalışsın, eylemleri yerine getirir. Örneğin, ona özgün metni de dönüştürülmüş bir formu da versen, yorumlayıcı adımları yürütür. Amaç her zaman programı çalıştırmaktır.



En büyük yanlış anlama, her yorumlayıcının satır satır okuduğunu düşünmektir. Doğru görüş, bunun uygulamaya bağlı olduğudur. Bazı yorumlayıcılar satır satır okur, ama birçoğu okumaz. Örneğin, bir yorumlayıcı yürütmeden önce bütün bir ifadeyi ya da bloğu okuyabilir. Yani yorumlayıcıların her zaman satır satır çalıştığını söylemekten kaçın.



İkinci bir yanlış anlama, yorumlama ile derlemenin asla karışmadığını düşünmektir. Modern uygulamalarda sıklıkla birlikte çalışırlar. Örneğin, bir sistem kaynak kodu ara koda derleyip sonra yorumlayabilir. Yani yorumlama ve derleme birbirinin zıddı değildir; aynı sürecin parçaları olabilirler. Birçok dil ikisini de kullanır.



Yorumlayıcı, bir programın işlemlerini çalışma sırasında, kaynak ya da ara koddan gerçekleştirir ve her zaman satır satır değildir. Yorumlama ile derlemenin birleştirilebileceğini unutma. Örneğin, Python hem bayt koduna derlemeyi hem de o bayt kodunun yorumlanmasını kullanır. Temel fikir: yorumlayıcı, program çalışırken programın talimatlarını yürütür.