Aydın Adnan Menderes Üniversitesi

Ziraat Fakültesi

Bitki Koruma Bölümü

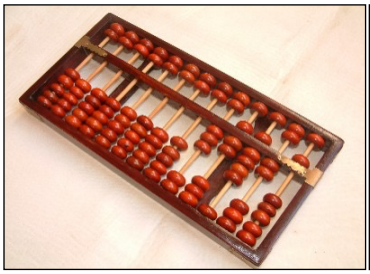
Bilgisayar ve Bilgi Teknolojileri

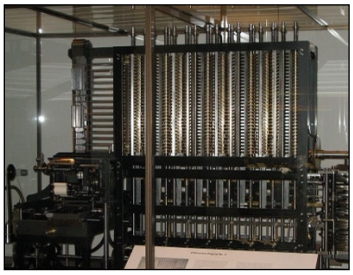
Ders Notları

Doç. Dr. Ümit ÖZYILMAZ

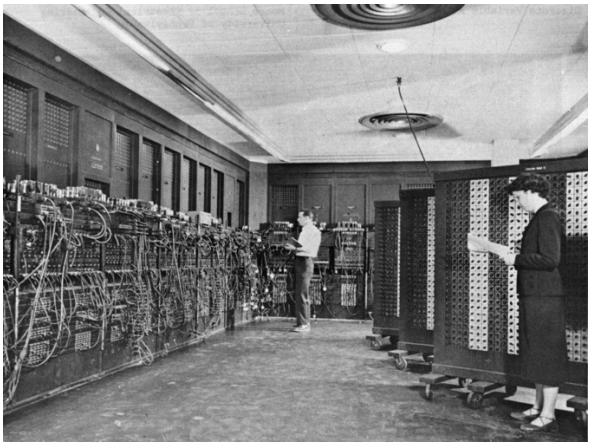
2025

Bilgisayarlar bir çok alanda kullanılan, hızlı ve doğru hesaplama yapabilen gelişmiş elektronik aygıtlardır. Günümüz teknolojisindeki gelişme ile birlikte bilgisayarların hızı artarken boyutu küçülmeye başlamıştır. Ayrıca işlem gücünün yanında giriş ve çıkış birimlerindeki bazı yenilikler ile kullanıcılar bilgileri ve istekleri bilgisayara daha rahat iletebilmekte, daha görsel ve farklı duyulara hitap eden çıktılar alabilmektedir. Çoğu uzman tarafından MÖ 2700-2300 yıllarında kullanılan *Abaküs* (solda) ilk bilgisayar olarak işaret edilmektedir. 1900 lü yıllarda Yunanistan açıklarında bulunan ve MÖ 27 de kullanıldığı düşünülen *Antikythera Mekanizması* (ortada) o dönemlerde gezegenlerin pozisyonlarını hesaplamakta kullanıldığı belirlenmiştir. Mekanik parçalar içeren bu tür cihazlar belirli bir amaca hizmet etmeleri için tasarlanmışlardır. Daha sonradan kullanıcılar tarafından programlanarak hesaplama yapan *Babbage'nin Fark Makinesi* (1833) (sağda) bilgisayarların atası olarak kabul edilmiştir.



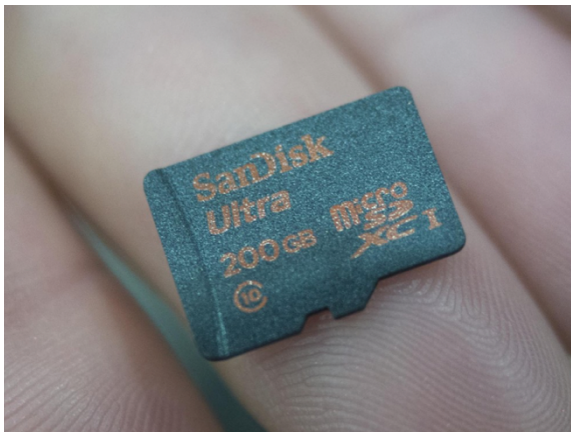


Mekanik parçalardan oluşan bu makineyi yaklaşık 100 yıl sonra üretilen ve hem mekanik hem de elektronik parçalardan oluşan *Z3* (1938) izlemiştir. 1930 larda ise ilk programlanabilir dijital bilgisayar *Colossus* tanıtılmıştır. Bu bilgisayarlarda günümüz bilgisayarların temelini oluşturan transistörlerin atası *vakum tüpleri* kullanılmıştır.



Colossus'u takiben 1946 daha hızlı ve daha esnek olan *ENIAC* (yukarıda) duyurulmuştur. Amerikan ordusu tarafından balistik hesaplamalarda kullanılan bu bilgisayar büyük bir salonu kaplayacak büyüklükteydi. 1980 lerde kişisel bilgisayar (Personel Computer [PC])

kavramı ile bilgisayar ticari kurumlara, okullara ve evlere girmeye başlamış ve bu süreç hızlı bir şekilde devam etmiştir. Bu yıllarda 10 MHz olan bilgisayarların hızları oldukça artmış ve çok hızlı işlem yapar duruma gelmiştir. Bilgisayarların hızlanması ve gelişmesiyle birlikte bilgilerin kaydedilmesine olanak sağlayan depolama alanlarına ihtiyaç duyulmuş ve bu amaca hizmet eden *Hard Disk Drive* (Hard Disk Sürücüsü) adında depolama ürünleri geliştirilmiştir. Gerek duyulan saklama alanının azlığı ve eldeki mevcut teknoloji nedeniyle ilk *HDD* (Hard Diskler) oldukça büyük ve az kapasiteye sahiplerdi. Her geçen gün daha fazla alana ihtiyaç duyulması nedeniyle hızlı bir şekilde kapasiteleri artmaya başlamıştır. Bunun aksine ise boyutları giderek ufalmıştır. 1950'li yıllarda bir dolap büyüklüğünde olan depolama birimleri (aşağıda solda) günümüzde bir pulun 4 te 1 büyüklüğüne (aşağıda sağda) gelmiştir.



Boyu ufalırken kapasiteleri ise binlerce kat artmıştır. Günümüzde media adı altında yer alan resim, müzik ve videolar depolama birimlerinde en çok yer kaplayan kaynaklardır. Bilgisayarda bir harfin kapladığı büyüklük 1 byte'tır ve bu bilginin büyüklüğü:

1024 Byte = 1 Kilobyte (kb)
1024 kb = 1 Megabyte (Mb)
1024 Mb = 1 Gigabyte (Gb)
1024 Gb = 1 Terabyte (tb)

şeklinde birbirinin 1024 katı şeklinde birimlerden oluşur.

Kişisel bilgisayar (*PC - Personel Computer*) çeşitli kısımlardan oluşmaktadır. Bu kısımlar hesaplamaların, işlemlerin yapıldığı ana işlemci (CPU), bilgisayarın ürettiği ya da

başkalarının ürettiği bilgilerin saklandığı depolama birimleri (Sabit veya harici diskler), kullanıcıların bilgisayara isteklerini ilettiği veri gerişini yaptıkları giriş birimleri (Klavye ve fare) ve işlenen/bulunan bilgilerin kullanıcaya gösterildiği çıkış birimleri (Ekran, printer, hoparlör vs) kısımlarından oluşmaktadır. Bilgisayar donanım (*Hardware*) ve yazılım (*Software*) kısımlarından oluşmaktadır. Donanım elektronik düzeneklerin her birine ya da tümüne denilmektedir. Yazılım ise bu donanımın çalışması için programcılar tarafından yazılan ve bilgisayarın belli bir amaca hizmet edecek şekilde çalışmasına yarayan komutlar dizisidir. Günümüz bilgisayarları ilk alındıklarında sadece donanım olarak gelmezler. Üzerinde donanımın çalışması, kullanıcıların program yükleyebilmesi için bir yazılımla birlikte gelirler. Bilgisayar açıldığında otomatik olarak yüklenen ve insanların bilgisayarı kullanmasına yarayan bu yazılımlara işletim sistemi adı verilmektedir. İşletim sistemi olmayan bir bilgisayar hiç bir işe yaramamaktadır. İşletim sistemleri bilgisayarı çalışır vaziyete getirmekle birlikte içlerinde basit işlerin yapılabileceği bazı programları da içerebilmektedir (Hesap makinesi, not defter, takvim vb.). Ancak bu programlar çok karmaşık işlemler için tasarlanmadıkları için yetersiz kalırlar. Bu nedenle kullanıcılar bilgisayarlarına hangi amaçla kullanılacaklar ise ona yönelik yazılmış özel programları (Resim işleme, mimari çizim yapma, oyun vb.) bilgisayarlarına kurmak zorundadırlar. Bu özel programlar bilgisayar alındıklarında üzerinde kurulu gelmezler. Bilgisayar için belli bir ücret ödeyen kişi bu tür programalar için de belli bir bedel ödemesi hatta bazı durumlarda işletim sistemi için bile ödeme yapması gerekmektedir. Yazılımların fikir ve sanat eseri çerçevesinde incelendiğinde belli bir emeğin karşılığı olduğu her ne kadar gözle görülür elle tutulur olmasalar da ciddi bir çabanın ürünü olduğu akıldan çıkarılmamalıdır. Bu nedenle bu yazılımların korsan olarak yüklenmesinden ziyade ücretinin ödenerek lisanslı olarak bilgisayara kurulması gerekmektedir. Ancak bazı programlar üreticileri tarafından kullanıcıların para ödemeden kullanabilmesi için yazılmışlardır. Bu tür programlar internetten rahatlıkla bulunabilir ve bilgisayara kurulabilirler. Bu programlara *Freeware* adı verilmektedir. Genellikle belli bir amaca hizmet eden ve para ödenerek bilgisayara kurulabilen programaların *Freeware* alternatifleri vardır.

İnternet ve Web Tasarımı

Günümüzde iletişim en parlak dönemini yaşamaktadır. Bunun en başta gelen araçlarından bir tanesi medya, diğeri ise internettir. Medya kaynaklı iletişimin de alt yapısını internet oluşturduğu düşünüldüğünde İnternet olgusunun günümüz temel iletişim aracı olduğu rahatlıkla söylenebilir. Bilgisayarları birbirine bağlayan bu büyük ağ (**network**) sayesinde telefon görüşmeleri, TV yayınları yapılmakta, anlık olarak haberleşebilmekte, paylaşılan haberler ile dünyada olan biten anında öğrenilebilmektedir. İnternet sayesinde dünya anlık olarak bilgiye erişme açısından küçük bir köye dönüşmüştür. Ancak internette oluşan bu bilgi bağımsız kurum ya da kuruluşlar tarafından oluşturabileceği gibi taraflı kurum kuruluş veya kişiler tarafından da oluşturulabileceğinden elde edilen bilginin doğruluğu veya gerçekliği her zaman için sorgulanması gereken bir olaydır. Bu nedenle internetten elde edilen bilgilerin güvenilir kaynaklardan elde edilmesi büyük önem taşımaktadır. Bir şekilde insanların internetten faydalanması beraberinde suçları da beraberinde getirmektedir. Neredeyse kontrolsüz bir bilgi yığını ve iletişimi olan internette dolandırıcılık olayları da çok sık görülmeye başlanmıştır. Kurumlar gerekli önlemleri almasına rağmen kullanıcılardan kaynaklanan hatalardan dolayı her gün yüzlerce kişi dolandırılmaktadır.

Bilgi alış verişi yapılan internet ağına sadece bilgisayarlar mı bağlanmaktadır? Bu sorunun cevabı kesinlikle hayırdır. Ev otomasyonları, müzik sistemleri, televizyonlar, cep telefonları, hatta artık arabalar bile internete bağlanma noktasına gelmiştir. Bilgisayarlar dâhil internete bağlanan her cihazın o cihaza özel, onu bulunduğu ağda tanımlayan ve sayılardan oluşan bir **IP** (Internet Protocol) adresi vardır. IP adresinin asıl görevi iletişime geçecek olan cihazların birbirlerini milyonlarca cihaz içinde bulmasıdır. Ayrıca internet üzerinde işlenen suçlarda IP adresinden suçu işleyen cihaz dolayısı ile kişi rahatlıkla bulunabilir. İnternete bağlanan her cihazın -ki bu bir televizyon, müzik sistemi veya telefon da olabilir- zorunlu olarak bir IP adresi vardır. Bu cihazlara IP adresini bağlı oldukları bir üst ağ cihazı belirlemektedir. Örneğin evinizde kullandığınız bilgisayar ya da akıllı internet televizyonuna IP adresini internete çıkmanıza yarayan modem vermekte, modemin IP adresini ise belli bir ücret karşılığı hizmet aldığınız internet servis sağlayıcı (ISP) şirketi belirlemektedir. Bu hiyerarşik bir sıralamadır.

İnternet dünya çapında bir ağdır. Dünyanın herhangi bir yerindeki bilgisayar dünyanın başka bir ucundaki bilgisayara bağlanabilir. Ancak bazı ağlar vardır ki tüm dünyaya açık değildir. Örneğin bankaların kendi içinde kullandığı ve tüm hesap işlerinin tutulduğu bilgisayarlar tüm dünyaya açık değildir, sadece bu kurum içinde bilgisayarlar birbirine bağlıdır, yani kapalı bir ağdır. Bu bankanın kullanıcılarına sağladığı sınırlı bankacılık işlemlerinin haricinde asıl hesapların tutulduğu bilgisayarlara dışarıdaki bilgisayarların bağlantısı yoktur. Bu kapalı ağlara **intranet** adı verilir. Bu ağlara örnek olarak askeri kurumların kullandığı birbirine bağlı bilgisayarlar da örnek verilebilir. Bilgisayarların bulunduğu fiziki konuma göre yan yana veya aynı binada bulunan bilgisayarlar aynı

bağlantıyı paylaşıyorsa lokal ağ içindedir ve buna lokal ağlar **LAN** (Local Area Network) adı verilir. Eğer bağlı olan bilgisayarlar fiziki olarak birbirlerinden uzak ise (örneğin farklı bina, mahalle, şehir veya ülke) ve birbirine bağlanması için özel cihazlar ve abonelikler gerektiriyorsa bu ağlara ise uzak ağlar **WAN** (Wide Area Network) adı verilir. Daha iyi anlaşılması açısından şu örnek verilebilir. Bir internet kafede birbirine bağlı bilgisayarlarda oyun oynayan insanlar lokal ağ (LAN) içindedir. İki ayrı internet kafede birbiri ile oyun oynayan insanlar ise uzak ağ (WAN) içindedir. İki ayrı kafeyi birbirine bağlayan internettir. Eğer internet kesilirse yani WAN bağlantısı koparsa (örneğin modem bozulursa) iki ayrı kafenin bağlantısı da doğal olarak kopar ancak insanlar aynı kafe içinde dahil oldukları lokal ağda (LAN) bulunan arkadaşları ile oyun oynamaya devam edebilirler. İnternette yapılan farklı işler için farklı iletişim protokolleri vardır. Milyonlarca cihazın bağlı olduğu internet çok büyük bir ağdır. İnternet içinde akan veri aynı yapıda değildir. Örneğin elektronik posta gönderme ve alma ayrı bir protokol kullanırken, dosya paylaşma ayrı, sesli ve videolu iletişim ayrı, web sayfalarına göz atma ayrı bir protokol gerektirmektedir. Bu protokoller bağlanan iki bilgisayar arasında nasıl bir haberleşme olacağının kurallarını içerir ve çok karmaşık bir yapıdadır. Ancak kullanıcıların kullandıkları bilgisayar programları bu protokol bağlantılarını otomatik yapmakta ve kullanıcıya hiç bir zaman bu zorluk yansıtılmamaktadır. Örneğin bir mail programı kullanılarak e-posta gönderilirken SMTP (Simple Mail Transfer Protocol), alınırken ise POP (Post Office Protocol) protokolleri otomatik olarak devreye sokulmakta ve haberleşme sağlanmaktadır. Kullanıcı bu karmaşık haberleşmenin ürünü olan sadece gelen maili görmektedir. Kullanıcı uzaktaki bir bilgisayara dosya göndermek istediğinde ise FTP (File Transfer Protocol) protokolü ile bağlantı kurmakta ve dosya göndermektedir. Web sayfalarına göz atmak için kullanılan Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari gibi **internet tarayıcısı programları** (Internet Browser) ise **HTTP** (Hyper Text Transfer Protocol) protokolünü kullanırlar. Bu yüzden web sayfaları `http://` ile başlamaktadır. Kullanıcılar iletişimde kullanılan karmaşık protokol komutlarını hiçbir zaman görmezler, onların gördüğü sadece görselliği yüksek, faydalı bilgiler içeren web sayfalarıdır. Ancak bu ders kapsamında tasarımcı olarak biz web sayfası yaparken **HTML** (Hyper Text Markup Language) komutlarını kullanacağız. HTML kodları HTTP protokolünün ana haberleşme dilidir ve web sayfalarının oluşturulmasında kullanılır.

İnternette bilgi hiç bir zaman kablolar üzerinde kalmaz muhakkak bir bilgisayar üzerinde barındırılmaktadır. Örneğin arkadaşınıza bir e-posta attınız, o anda arkadaşınızın bilgisayarı kapalı olabilir, arkadaşınıza e-posta ulaşmadan siz de bilgisayarınızı kapattınız, şu anda her iki bilgisayarda kapalı olduğunu düşünelim. Bir süre sonra arkadaşınız bilgisayarını açtı ve e-posta ona ulaştı. Peki, bu nasıl olmaktadır? Arkadaşınızın e-posta adresi `arkadas@hotmail.com` olsun, aslında siz e-postayı hotmail şirketindeki bilgisayara atmaktasınız, postanız bu şirketteki bilgisayarda durmakta, arkadaşınız ne zaman kendi bilgisayarını açarsa ona buradan ulaştırılmaktadır. Bu size hizmet eden bilgisayarların

genel olarak **sunucu (server)** adı verilmektedir. Sunucular hizmet verebilmeleri için 24 saat açık tutulurlar ve hiç bir şekilde kapatılmazlar, elektrik kesintilerine karşı korunurlar. Başka bir örnek daha verelim; üniversitemizin web sayfa adresi `http://www.adu.edu.tr` dir. Bu adresi internet tarayıcınıza yazdığınızda üniversitemize ait bilgilerin yer aldığı web sayfası bilgisayarınızda görüntülenir. Ekrana gelen bilgilerin kaynağı üniversitedeki 24 saat açık bulunan sunucudur. Eğer bu sunucu kapatılır ya da bir şekilde kendi kapanırsa web sayfasına girmek istediğinizde sunucu hatası ile karşılaşır ve sayfayı görüntüleyemezsiniz. Sunuculara bilgi almak için bağlanan bilgisayarların genel ismine ise **istemci (client)** adı verilmektedir. Yukarıdaki örnekte üniversitedeki 24 saat açık olan bilgisayar sunucu, sunucuya bağlanıp oradaki web sayfasını görüntüleyen sizin bilgisayar ise istemci rollerindedir. Günlük yaşamda kullandığımız ve internete bağlanan bilgisayarlar, cep telefonları gibi aygıtların hepsi istemcidir ve 24 saat açık olan sunuculardan bilgi almak için kullanılmaktadır. Web sayfalarını içinde tutarak bu sayfanın bilgilerini barındıran sunucuların ismine **host** adı verilir. Hazırlanan web sayfaları devamlı ulaşılabilir olması için hiç kapatılmayan host sunuculara yüklenmek zorundadır. Dosyaların host sunucuya yüklenmesi FTP protokolü ile gerçekleştirilmektedir. Bunun için bir FTP programına ve host şirketi tarafından size verilen kullanıcı ismi ve şifreye gereksinim vardır (Bu konuda host şirketinden yardım alabilirsiniz). Bu host sunucular ticaridir, web sayfanızı belli bir süre yayınlamak için sizden ücret talep ederler ve sözleşmenizin belirli sürelerde yenilenmesi gerekmektedir. Aksi halde dosyalarınız host tan silinmekte ve siteniz artık ulaşılamaz hale gelmektedir (bazı sunucular reklam karşılığı ücretsiz hizmet verebilmektedir). Her web sayfasının ona ulaşılabilmemizi sağlayan **URL** adında bir bağlantı adresi vardır. Üniversitemiz web sayfası URL si `http://www.adu.edu.tr` dir. Nasıl web sayfalarının barındırılması için host sunucusu şirketine bir ücret ödenmekteyse, aynı şekilde URL isminin tescillenmesi ve alınması içinde belli bir ücret ödenmektedir. Alınan bu URL size özeldir, bir kopyası daha yoktur ve belli periyotlarda sözleşmenin yenilenmesi gerekmektedir. Sözleşme yenilenmezse URL hizmet aldığınız şirket tarafından kapatılmakta ve talep olursa isim bir başkasına satılabilmektedir. Dikkat edilecek olursa URL belli kısımlardan oluşmaktadır. Baş kısımdaki `http://` bu URL nin bir web sayfası protokolü olduğunu göstermektedir. Daha sonra gelen `www` alt alan adı (subdomain) dır. Alt alan adı istenirse kullanılmayabilmekte, kullanıldığı durumda web sayfası sunucu yöneticisi tarafından istenildiği gibi isimlendirilmektedir. Örneğin bir şirketin farklı departmanları için `http://www.sirketim.com` `http://satis.sirketim.com` `http://yonetim.sirketim.com` alt alan isimleri kullanılabilmektedir. Verilen bu örnekte asıl alan adı (domain) sirketim dir (isim alınırken verilmesi zorunludur). Bu kısım web sayfası sahibini tanımlayan asıl kısımdır. Bunu takiben ticari kuruluş ise `com`, eğitim kurumu ise `edu`, hükümet kuruluşu ise `gov`, dernek organizasyon ise `org`, askeri kuruluş ise `mil` eki gelmektedir (bu kısım da isim tescillenirken verilmek zorunludur). En son ek ise ülke kodudur, kullanılma zorunluluğu yoktur. Örnek olarak Türkiye için `tr`, İtalya için

`it`, Fransa için `fr` ve benzeri şekilde çoğaltılabilir. URL tescil edilirken sadece bir kısmı değil bir bütün olarak tescillenmektedir. `http://webtasarim.com` `http://webtasarim.com.tr` `http://webtasarim.org` isimleri ayrı ayrı tescil edilmek zorundadır. Bazı web sayfalarının http:// yerine https:// ile başladığını dikkat etmiş olabilirsiniz. Genelde banka, sosyal medya, özel haberleşme siteleri https:// protokolünü kullanmaktadırlar. Bu protokolde web sayfasına yazılan kredi kartı numaraları, yazışmalar, metinler ve sayfa içeriklerinin tamamı kullanıcı ve web sayfası sunucusu arasında gizli tutulmakta ve şifrelenmektedir. Hiçbir şekilde üçüncü şahıslar tarafından görüntülenememektedir. Çok kuvvetli olan şifreleme tekniği sayesinde çözülmesi neredeyse imkânsızdır. Bu ne anlama gelmektedir? Bir bankanın web sitesi üzerinden bankacılık işlemi yapmak istiyorsunuz ve adresin https:// ile başlamadığını gördünüz. Bu durumda bankanın web sitesine ulaşana kadar aktarıldığınız tüm ara bilgisayarlar ve sunucularda (en basit bağlantılarda bile en az 3-4 adettir) ya da araya giren kötü niyetli bilgisayarlarda istenildiği taktirde yaptığınız tüm işlemler görülebilmekte, şifreler okunabilmekte, kredi kartı numaraları alınabilmektedir. Eğer gizlilik sizin için önemli ise protokolün https:// olmasına dikkat etmeniz gerekmektedir.
Bir web sayfasının URL adresinin internet tarayıcısına girilmesi ile nasıl oluyor da ilgili host sunucusuna bağlanılmaktadır. Her bilgisayarın bir IP adresi olduğundan daha önce bahsedilmişti. URL tescil edilirken web sunucusunun IP adresi ile URL adı eşleştirilmekte ve **DNS** (Domain Name Server) adı verilen büyük sunuculara bu bilgi kayıt edilmektedir. İnternet tarayıcısına URL yazıldığı zaman DNS sunucusundan URL'nin IP adresi bulunmakta ve bu IP adresine ait bilgisayara otomatik olarak yönlendirme yapılmaktadır. Böylece bir web sitesine gitmek için IP adresinin ezberlenmesine gerek kalmamakta sadece internet tarayıcısına URL girilmesi yetmektedir. Örneğin `http://www.adu.edu.tr` web sayfası sunucusunun IP adresi `194.27.38.14` tür. İnternet tarayıcısına URL yazılıp entere basıldığında otomatikman DNS sunucusundan bu IP adresi hemen bulunur ve bilgisayara erişilir. Adres kısmına IP adresini yazarak da yine aynı sayfaya erişilebilmektedir.

HTML Dili

HTML dili ile yazılmış bir web sayfası `<html>` ile başlar ve `</html>` ile biter. HTML dilinde komutlar "<…>" işareti içerinde kullanılır ve komutun bitimi ise "</…>" şeklindedir. Çoğunlukla her başlama komutunun bir bitiş komutu vardır (istisna olarak sadece <…> şeklinde bitişi olmayan komutlar da mevcuttur). Sayfa içindeki her komutun başlangıcı ve bitişi bir kalemle birleştirildiği varsayılırsa hiçbir çizginin kesişmemesi gerekir. Eğer kesişiyorsa WEB sayfasında hata oluşur ve büyük bir çoğunlukla sayfa hata vermeksizin yanlış görselde gösterilir.

```
<html>
İlk Web sayfamız
</html>
```

HTML dilinin yazıldığı ve sayfa dizaynlarının yapıldığı çok gelişmiş programların olmasına karşın basit bir metin editörü bu dilin yazılması için yeterlidir. Örneğin Windows işletim sisteminin içinde bulunan “Not Defteri” programı bu çalışmalar için rahatlıkla kullanılabilir. Bu program Windows’un “Başlat menüsü” altındaki “Tüm Programlar>Donatılar” altında yer alır. “Not Defteri” ile yazılan komut dizisi kaydedilirken dikkat edilmesi gereken bir-iki nokta vardır. Kaydedilirken “Kayıt türü”’nden “Tüm Dosyalar (*.*)” seçilmeli ve dosya ismi verilirken uzantı olarak “.html” eklenmelidir (sayfam.html örnek olarak verilebilir). Dosya isimlerinde Türkçe karakter (Çç,Ğğ,İı,Öö,Üü) kullanılmaması gerekmektedir. Kaydedilen dosya üzerine çift tıklandığı zaman otomatik olarak varsayılan web tarayıcı tarafından gösterilir. Eğer dosya üzerinde bir değişiklik yapılmak isteniyorsa dosya üzerinde farenin sağ tuşuna basılır ve “Birlikte Aç…” menüsünden “Not Defteri” seçilir ya da doğrudan “Not Defteri” programı içinden de bu dosya açılabilir (Dosya türü “Tüm Dosyalar (*.*)” seçilmelidir aksi halde dosya görünmez). Host sunuculara web sayfasının görüntülenmesi için bağlanıldığında otomatik olarak gösterilen ilk ana sayfa index.html index.htm ya da default.html dosyalarından biridir. Bu nedenle ana sayfanızın ismi bu dosya isimlerinden biri olmak zorundadır.

Basit HTML komutları

<font> yazı karakteri ayarlama komutu

Bu komut yardımıyla sayfanın herhangi bir alanına yazılan yazıya boyut, yazı stili ve renk verilebilir. Yazı boyutunu size, sitilini face, rengini ise color parametreleri ayarlar. Bu parametler aynı anda kullanılabileceği gibi sadece biri veya ikisi de kullanılabilir.

```
<html>
<font size=12 face=”Verdane” color=”Red”>deneme1</font>
<font size=8 face=”Courier” color=”Blue”>deneme2</font>
<font size=14 face=”Times New Roman” color=”Yellow”>deneme3</font>
</html>
```

Mevcut yazı tipleri ve renkler için ek1’e bakınız.

<p> paragraf komutu

Web sayfası içinde her paragraf bu komut ile belirtilir. Eğer bu komut verilmeden metin editör içinde paragraf verilerek yazılsa dahi tarayıcıda tek bir paragrafta görülür. Paragraflar `<p>…</p>` arasında verilerek belirtilir. `<p>` sadece yalın halde kullanılabileceği

gibi paragrafın sola (left), sağa (right), her iki yana yaslı (justify) veya ortalı (center) yazılmasına yarayan align parametresi ile de kullanılabilir. Örneği ortalı bir paragraf için `<p align=center>...</p>` kullanılmalıdır. Eğer align parametresi kullanılmazsa varsayılan olarak sola yaslı kabul edilir.

```
<html>
<p>ilk paragraf</p>
<p>ikinci paragraf</p>
</html>
```


 satır başı komutu

Bir paragrafta bir alt satıra inilmek için kullanılır ve sadece
 şeklinde kullanılır. </br> gibi komutu sonlandıran bir kullanımı yoktur.

```
<html>
<body>
<p>bu birinci satır<br>bu ikinci satır</p>
</body>
</html>
```

<h1> başlık atma komutu

Bu komut metin karakterlerinden büyük ve kalın bir başlık atılması gerektiğinde kullanılır. `<h1><h2><h3><h4><h5><h6>` olmak üzere 6 adet başlık büyüklüğü vardır (En büyük başlık `<h1>` dir ve büyüklüğü `<h6>` doğru azalır).

```
<html>
<h1>bu bir başlık satırıdır</h1>
<p>bu da normal bir metindir</p>
<h2>bu ilk başlığa göre daha küçük bie başlıktır</h2>
<p>bu da ikinci başlığın altında yazan bir metindir</p>
</html>
```

<b> kalın karakterde yazma

Bu komut ile normal metinden daha kalın yazılması sağlanır ve `</b>` komutu ile sonlandırılır. Metin içerisinde `<b> bu yazılan </b>` diğerlerinden daha kalındır.

<i> yatık karakterde yazma

Bu komut ile yatık yazıların yazılması sağlanır ve `</i>` komutu ile sonlandırılır. Metin içerisinde `<i>bu yazılan</i>` yatık karakterdedir.

<u> altıçizili karakterde yazma

Bu komut ile altı çizili yazıların yazılması sağlanır ve `</u>` komutu ile sonlandırılır. Metin içerisinde `<u>bu yazılanın</u>` altı çizilidir.

<sup> üst karakterde yazma

Bu komut ile $2x10^6$ gibi üst karakterli metinlerin yazılması sağlanır ve `</sup>` komutu ile sonlandırılır.

```
2x10<sup>6</sup>
```

<sub> alt karakterde yazma

Bu komut ile H_2SO_4 gibi alt karakterli metinlerin yazılması sağlanır ve `</sub>` komutu ile sonlandırılır.

```
H<sub>2</sub>SO<sub>4</sub>
```

<hr> sayfa boyunca yatay bir çizgi çekmek

Sayfaya yatay olarak bir çizgi çekmek için kullanılır. Aynı `<br>` komutunda olduğu gibi sonlandırıcı bir komutu yoktur.

```
<html>
<p>bu paragraf ile</p>
<hr>
<p>bu paragraf arasında bir çizgi var</p>
</html>
```

<img> resim ekleme komutu

Web sayfası içine sunucuda bulunan bir resmi ya da farklı bir internet adresindeki resmi eklemek için kullanılır. Bu resim dosyası veya adresi src parametresi ile verilir. `<img src=”resim.jpg”>` şeklinde yazılır. `</img>` gibi kapatıcı komut kullanımı olmayan bir komuttur. Eğer resim daha ufak bir boyda gösterilmek istenirse height (yükseklik) ve width (genişlik) değerleri ile piksel cinsinden verilebilir. O zaman komut şu şekilde yazılmalıdır `<img src=”resim.jpg” height=200 width=240>`

<table> tablo oluşturma

Metin içerisinde tablolar oluşturmak için kullanılır. Kullanımı diğer komutlar gibidir, ancak uygulamada biraz dikkat etmek gerekir. Çünkü tabloda yer alacak satır ve sütün sayılarının verildiği alt komutları vardır. Her tablo `<table>` ile başlar ve `</table>` ile biter. Eğer tabloya çerçeve eklenmek istenirse `<table border=1>...</table>` şeklinde border parametresi kullanılır. border'in yanına verilen değer çerçevenin kalınlığını tayin eder. Eğer border değeri verilmezse `<table>...</table>` şeklinde kullanılırsa çerçeve gösterilmez. Tablonun her satırını ve sütununu bir komut belirler. Tablodaki her satır `<tr>` ile başlar ve `</tr>` ile biter. Her satır içinde sütunlar `<td>...</td>` ile belirtilir. Aşağıdaki örneği inceleyerek nasıl kullanıldığını daha net görebilirsiniz. Bu örnekte 3 satırdan ve 2 sütundan oluşan bir tablo yapılmaktadır.

```
<html>
<table border=1>
      <tr>
            <td>birinci satırın birinci sütunu</td>
            <td>birinci satırın ikinci sütunu</td>
      </tr>
      <tr>
            <td>ikinci satırın birinci sütunu</td>
            <td>ikinci satırın ikinci sütunu</td>
      </tr>
      <tr>
            <td>üçüncü satırın birinci sütunu</td>
            <td>üçüncü satırın ikinci sütunu</td>
      </tr>
</table>
</html>
```

Bu tablonun çıktısı şu şekilde olacaktır;

birinci satırın birinci sütunu	birinci satırın ikinci sütunu
ikinci satırın birinci sütunu	ikinci satırın ikinci sütunu
üçüncü satırın birinci sütunu	üçüncü satırın ikinci sütunu

<ul ve ol> Listeleme

Bazı durumlarda hazırlanan web sayfaları içinde listelere ihtiyaç duyulmaktadır. `<ul>` komutu listelenecek olan maddelerin başına bir liste işareti ekleyerek bunların düzgün bir sıra içinde sıralanmasını sağlamaktadır. Listede yer alacak her madde `<li>...</li>` içine alınmalıdır. Listeleme işlemi bittiğinde `</ul>` ile listeleme komutu kapatılmalıdır. Eğer listeleme işlemi liste işareti yerine sıra numarası ile yapılacaksa `<ul>` komutu yerine

 komutu kullanılmalı ve yine komut dizisi ile kapatılmalıdır. Numaralı listelerde komut yazarken numara verilmez, çıktı kendiliğinden numaralandırılır. Örneğin sıra numarası elle girilerek verilen 100 kişilik bir isim listesinde aradan bir kişi çıkarıldığında tüm sıra numaralarının elle tek tek değiştirilmesi gerekir. Ancak bu listeleme <ol> komutu kullanılarak yapılırsa numaralandırma otomatik olacağından aradan kaç isim çıkarılırsa çıkarılsın sıra numaraları hep kendiliğinden güncellenecektir.

```
<ul>
      <li>Ali</li>
      <li>Ahmet</li>
      <li>Veli</li>
</ul>

<ol>
      <li>Elma</li>
      <li>Armut</li>
      <li>Muz</li>
</ol>
```

Bu örneğin çıktısı şöyle olacaktır;
• Ali
• Ahmet
• Veli

1. Elma
2. Armut
3. Muz

<a> bağlantı verme

Bu komut sayfa içinden başka bir internet sayfasına bağlantı vermek için kullanılır. Bağlantı verilen sayfa bizim yazdığımız bir sayfa olabileceği gibi internette bulunan herhangi bir WEB sayfası da olabilir. Yazılan bu komut ile fare ile tıklandığında otomatik olarak verilen sayfanın yüklenmesi sağlanır. Genellikle sayfa içindeki linkler mavi ve altı çizgili olarak kullanıcıya gösterilir ve diğer metinden ayrılır. Gidilecek olan sayfa href değişkenine verilir ve ile komut kapatılır. Aşağıdaki örneği inceleyiniz. Bağlantı aynı pencereye yüklenir.

```
<html>
Adnan Menderes Üniversitesinin WEB sayfasına gitmek için
<a href=”http://www.adu.edu.tr”>bunu tıkla</a>
</html>
```

Adnan Menderes Üniversitesinin WEB sayfasına gitmek için bunu tıkla

Bilgisayar Programlama

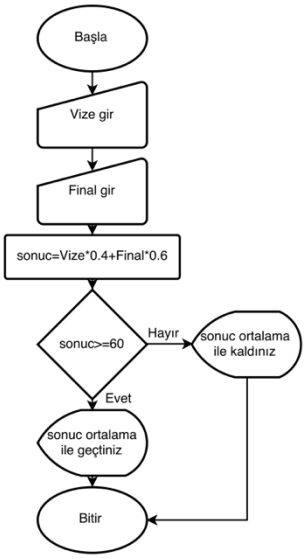
Programlar programcılar tarafından *programlama dilleri* kullanılarak oluşturulurlar. Yani kullandığımız programlar da başka bir program tarafından meydana getirilirler. Programcılar kullandıkları programlama diline özgü (ki çok sayıda programlama dili vardı) bir takım komutları alt alta yazarak programı oluştururlar. Bilgisayar programcıların program yazabilmeleri için bu yazdıkları programın hangi sorunu çözeceğini ve neye yarayacağını iyi irdelemesi ve anlaması gerekmektedir. Yani yazılan bilgisayar programının çözeceği sorun tam olarak ortaya konulmalıdır. Sorunu anlayan programcı programı yazmadan önce sorunun nasıl çözüleceğine dair fikir jimnastiği yapar, araştırmalarda bulunur. Sorunun eksiksiz çözülmesinin yanında hızlı bir şekilde çözülmesi de gerekmektedir. Bu konuları göz önünde bulunduran programcı *Algoritma* adı altında neyi, ne zaman ve nasıl yapacağını iyice gözden geçirerek bir *akış şeması* oluşturur. Bu akış şeması üzerinde alıştırmalar yaparak sorunun çözülüp çözülmediğini kontrol eder. Daha sonra seçtiği ya da hakim olduğu bir programlama dili ile oluşturduğu bu algoritmayı programlama diline özgü komutları yazarak oluşturur. Çalışıp çalışmadığını farklı koşullarda veriler girer doğru hesaplayıp hesaplamadığını kontrol eder. Buna ek olarak yazdığı programa uygun olmayan veriler girerek bu uygunsuz koşullarda programın nasıl davranacağını hata verip vermeyeceğini kontrol eder ve programı son haline getirir. Bu süreci bir örnekle açıklayalım.

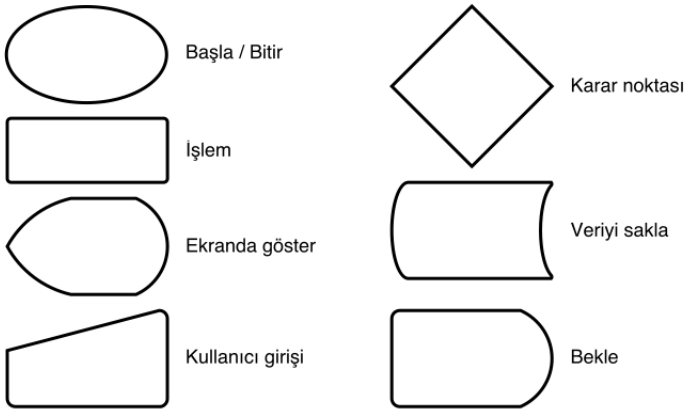
İlk önce bilgisayarla çözülmesi gereken bir sorunun/problemin ortaya konması gerekmektedir. Bizden öğrencilerin aldığı vize ve final notundan ortalamalarının ne olacağı, geçip geçemeyeceklerine dair bir bilgisayar programı yazmamız istensin. Bu soruyu çözebilmemiz için bazı temel bilgilere ihtiyacımız var. Örneğin vizenin % kaçı ve finalın % kaçı ortalamayı etkilemektedir. Ortalama kaç ve üzeri olduğunda öğrenci geçmektedir. Bu bilgiler dikkat edilecek olursa sabittir. Programcı hemen bunu sorgular ve bu sabit değerleri ister. Programcı adım adım neler yapacağını düşünmeye başlar. İlk önce vize ve final notlarının girilmesi istenmelidir. Daha sonra öğrendiğimiz sabit sayılar kullanılarak bu girilen notlar ile ortalama hesaplanmalıdır. En sonunda elde edilen bu ortalama notun öğrencinin geçip geçemeyeceği konusunda karar verilmelidir. Tüm bu işlemler bittikten sonra sonucun ekrana yansıtılması gerekmektedir. Bu aşamaya kadar dikkat edilecek olursa daha program yazmaya başlanmamıştır. Sorunun tanımı ve nasıl yapılacağı konusunda fikirler tartışılmaktadır. Bu aşamaya algoritma denilmektedir. şekiller çizerek sembolize etmeye de akış şeması denilmektedir. Örnek basit gibi görünse

de değişik senaryoların da düşünülmesi gerekmektedir. Bu hatalı bir sonucun verilmemesi ve programın hata verip kendini kapatmaması için önemlidir. Örneğin yukarıda verilen örnekte vize notu istendiğinde kullanıcı 120 yazarsa ne olur? Programcı algoritmayı oluştururken bu gibi durumları düşünmek zorundadır. Algoritmaya yani sıra ile yapılması gerekenler şemasına vize ve veya final notu girildiğinde notun 0 ile 100 arasında olup olmadığı kontrol edilmesi varsa bir yanlışlık kullanıcının değeri tekrar girmesi istenmelidir. Bu ve bunun gibi tüm olasılıklar düşünülmek zorundadır. Bilgisayar göz açıp kapayıncaya kadar bu bahsedilen problemi tek kişi için çözebilir. Peki bu şekilde notu hesaplanması gereken milyonlarca kişi varsa? Bu noktada ürettiğiniz çözümün kısa ve hızlı olması gerekmektedir. Örneğin karmaşık matematiksel bir işlem yapmanız istendi ve işlem 1 saniyenin yarısı bir sürede hesaplanıp bitti. Gayet kısa bir süre, 0.5 saniyede anca göz açılıp kapanır, sonuç iyi mi peki? Aynı işlemi verilen 1 milyon rakam ile yapmanız istendi, her işlem 0.5 saniye sürüyordu x 1.000.000 = 500.000 saniye bu da yaklaşık 6 gün demektir. Tasarladığınız çözümde yapacağınız bazı değişiklikler ile bu çözümü kısaltmanız gerekmektedir. Tek bir işlemde anlaşılamayan hız sorunu çok işlemde büyük bekleme sürelerine denk gelebilmektedir. Belki de bu tüm işlemleri 1 dakikanın altında yapmanın başka bir yolu vardır? Belki de! Önünüzde 10 tane farklı sayı olduğunu hayal edin ve bunları sıralamanız gerektiğini düşünün, beyniniz hemen kendine göre bir çözüm bulur ve bunları sıralarsınız. 100 sayı 1.000 sayı ya da 10.000 olduğunu düşünün! Artık beyninizin bunları hemen sıralayamadığını göreceksiniz ve bunları sıralamanın bir yolunu bulmaya çalışacaktır. Ve farklı yöntemlerin olduğunu göreceksiniz, kiminin hızlı, kiminin çok emek istediğini fark edeceksiniz. İşte algoritma bir bilgisayar programının sorunu nasıl çözeceğine dair hangi sıralı adımları işleyeceğini tasarlamaktır. Bu tasarının şekiller ile gösterilmesi ise akış şemasıdır. Programlama bu akış şeması bittikten sonra yazılmaya başlanır.

Yukarıda da anlatıldığı üzere algoritma bir şeyin nasıl yapılacağı konusunda önceden düşünmek ve bu düşünceyi şematize etmektir. Günlük yaşantımızdaki her şeyin algoritması çıkarılabilir. Örneğin bir çamaşır makinesinin çamaşırı nasıl yıkadığını düşünebilirsiniz. Belli bir sıra içerisinde devam eden olaylar kolaylıkla şematize edilebilir. Örneğin su al, deterjan al, tamburu 50 kez sağa sola çevir, suyu at, tekrar su vb. Ancak daha detaylı düşünüldüğünde bu adımların çok basit kaldığı aslında çok daha detaylı bilgilerin gerektiği ortaya çıkmaktadır. Örneğin kapısı açık bir çamaşır makinesi çalışmaz, demek ki algoritmasının içerisinde kapı açık mı > evet > o zaman çalışma gibi bir adım

içermektedir. Bu ve bunun gibi gözden kaçırabileceğimiz çok sayıda adım yer alabilir. Bu adımlar uzun çalışmalar sonucunda mühendislerin yaptığı denemeler ve hesaplamalar ile kesinleştirilmiştir. Mesela tamburu kaç tur attırmak çamaşırı iyi yıkar, ne kadar su çamaşırın iyi yıkanması için yeterli gelmektedir, iyi durulanması için ne kadar suya ihtiyaç vardır? Tüm bunlar araştırılıp, derlenip, planlanıp tek bir detaylı algoritma çıkarılması gerekmektedir. Yapılan algoritmanın iyi bir şekilde algılanabilmesi için şekilsel sembolleştirilmesi çok yararlı olacaktır. Nitekim programcılar belli standart şekilleri algoritmalarında kullanarak şematize ederek akış şemalarını oluştururlar. Böylece yapılan kurgu çok rahat bir şekilde takip edilebilmektedir.





Algoritma evrenseldir, programı hangi bilgisayar programında yapılacak olursa olsun tek tiptir. Bu günlük yaşantımızdaki yapacağımız herhangi bir iş planı gibidir. Programlama ise bu algoritmanın verilen akış sırası ile komut yazmasıdır ve her bilgisayar programı için farklılıklar gösterir.

Algoritma ve akış şemasında kullanılan başlıca şekiller ve anlamları aşağıda verilmiştir. Buna göre yukarıdaki problemin akış şeması çıkarılacak olursa resimdeki gibi bir akış çıkmaktadır. Bu örnek kullanıcı girişi, işlem, karar verme ve ekrana gösterme adımları içeren basit bir örnektir. Dikkat edilecek olursa vize çarpanı 0.4, final çarpanı 0.6 ve geçme sınırı 60 sabit olarak verilmiştir. Bu tür değeri değişmeyen ve program içinde sıklıkla kullanılan değerlere *sabitler* adı verilmektedir.

Programlama

Neyin nasıl yapılacağı akış şeması ve algoritma oluşturulduktan sonra sıra bunun bilgisayar programının yazılmasına gelmektedir. Bilgisayar programları programlama dilleri kullanılarak yazılan komut dizilerinden ibarettir. Yazılan program programlama dili içerisinde çalıştırılarak kullanılabileceği gibi, kendi başına çalışır ve başka bilgisayarlara gönderilerek başka kullanıcıların da çalıştırmasına olanak sağlayan kendi çalışır bir forma (derleme) da sokulabilmektedir. Yazdığınız program çeşitli komutlardan oluşmaktadır. Program yazma işi bittiğinde bu komutlar derlenir ve *EXE* uzantılı bir dosya oluşturulur. Bu dosya istenilen bir bilgisayara kopyalanarak çalıştırılabilir yapıdadır ve sizin komutlarını yazdığınız programı çalıştırır. Yazdığınız kodların yalın halde başka bir bilgisayara kopyalanıp çalıştırılması mümkün değildir, muhakkak derlenip EXE uzantılı dosyanın oluşturulması gerekmektedir. Bu çalışan EXE uzantılı dosya sizin istediklerinizi tek başına yapmasına karşın bu dosyanızın içine girilip yazdığınız komutlar görülememektedir. Bu dosyaların içinin okunup yazdığınız kodların görülmesi imkansızdır. Programda yaptığınız bir hatayı fark ederseniz yazdığınız kodlar üzerinde gerekli değişiklikleri yapıp tekrar derleyerek EXE dosyanın oluşturulması gerekmektedir. EXE dosyası üzerinde değişiklik yapılması imkansızdır. EXE dosyası üzerinde yapılacak herhangi bir değişiklik bilgisayarın kilitlenmesine dahi neden olabilecek sorunlar çıkarabilir.

Peki bilgisayar programı nasıl yapılmaktadır? Bilgisayar programı kullanılan dile özgü belli komutlar içermektedir. Klavyeden giriş yapmanın komutu farklıdır, ekrana bir şey yazdırmanın farklıdır, karar vermenin farklıdır. Bu komutlara ileride bahsedilecektir. Program aynı bir mektup gibi yukarıdan aşağıya doğru yazılır ve bilgisayar bu programı aynı sizin yazdığınız satır sırasına göre bunu çalıştırır. Yani yapmak istediklerinizi mantık ya da oluşturduğunuz algoritma sırasına göre komut satırlarını yazmanız gerekmektedir. Yukarıda bahsedilen örnekte ekrandan ilk önce vize notu istenmeli, daha sonra final notu istenmeli, işlem yapılmalı, karar verilmeli ve sonuç ekrana yazılmalıdır. Bu sıranın değişmesi kurgu hatasıdır ve istenmeyen bir durumdur, programın amacına ulaşması imkansızdır. Program yukarıdan aşağıya doğru komutları çalıştıra çalıştıra inmektedir. Peki en başa ya da istediğimiz bir noktaya geri dönmek mümkün olur mu? Evet bazı durumlarda bu akışın içinde bilgisayarı istediğimiz noktaya yönlendirmek mümkündür, bilgisayar bu noktaya gider ve buradan komut akışını aşağıya doğru işletmeye devam

eder. Artık tüm satırlar biter çalıştırılacak komut kalmadığında ise program sonlanır ve otomatikman programdan çıkılır.

Programlama mantığının anlatıldığı bu notta BASIC programlama dili esas alınmıştır. Bu programlama dili eski olmasına karşın, anlaşılması basit ve günümüz modern programlama dillerinin sahip olduğu hazır fonksiyonlar içermemektedir. Programlama mantığı tüm programlama dilleri için yaklaşık olarak aynıdır. Bir programlama dilinin öğrenilmesi diğer bir programlama dilinin öğrenilmesini ciddi ölçüde kolaylaştırmaktadır.

Tek komutluk ilk programımız şöyle olsun;

```
PRINT "Merhaba Dünya"
```

```
Merhaba Dünya
```

Yukarıdaki örnekte komut PRINT "Merhaba Dünya" dır sonucu ise kutu içine alınmış olan ve bilgisayarın ekranına yansıyacak olan **Merhaba Dünya** dır. Verilen örnekler bu şekilde aktarılacaktır. BASIC programlama dilinde ekrana herhangi bir şey yazdırılmak istendiğinde PRINT komutu kullanılmaktadır. Her PRINT komutu ile yapılan yazdırılma işleminden sonra imleç aşağıya düşer. Eğer yeni bir satırdan ziyade yan yana yazılması isteniyorsa ilk yazdırılma işleminden sonra ; konulmalıdır.

```
PRINT "Merhaba"
PRINT "Dünya"
PRINT "Algoritmalar ve Bilgisayar ";
PRINT "Programlama"
PRINT "Ders Notları"
```

```
Merhaba
Dünya
Algoritma ve Bilgisayar Programlama
Ders Notları
```

PRINT <değer> ekrana istenilen bir bilgiyi yazdırmak için kullanılır

Nümerik, Alfanümerik Değişkenler ve Sabitler

Programlamada işlem ve hesap yapmak için sayılar ve harfler kullanılır. Sayılara *nümerik* değişkenler denilirken isim, adres gibi matematiksel işlem yapılamayacak değişkenlere ise *alfanümerik* değerler adı verilmektedir. Alfanümerik değerler sayı içeriyor olsalar bile bunlar ile matematiksel işlemler yapılamamaktadır. Alfanümerik değerler "" içinde bilgisayara verilirler. Bu tırnak işareti içindeki bilginin harflerden sayılardan oluşan ve matematiksel işlem yapılamayacak bir değer olduğunu göstermektedir. *Değişken* ise nümerik ya da alfanümerik değerlerin atandığı kelimelerdir ve her programlama dilinde kullanılmaktadır. Örneğin vize=67 demek vize değişkenine 67 nümerik değerini atamakta ve vize değişkeninin değeri bundan sonra değiştirilinceye kadar 67 olmaktadır. Başka bir örnek verecek olursak soyadi$="yılmaz", bu örnekte soyadi değişkeninin değeri yılmaz olarak atanmıştır, bilgisayara ne zaman soyadi$ nin değeri sorulsa size yılmaz değerini verecektir. Dikkat edecek olursanız alfanümerik değişkenler sonuna $ işareti almaktadır ve "" içinde ifade edilirler. sonuc$="37" örneğinde sizce 37 nin atandığı sonuc$ değişkeni nümerik midir, alfanümerik midir. Cevap alfanümeriktir, her ne kadar sayı gibi görülse de $ ve "" den dolayı bilgisayar bunu sayı olarak değil bir isim gibi algılar ve matematiksel işlem yapamaz. Yukarıdaki örneklerde vize, soyadi ve sonuc değişkenlerinde dikkat edildiği üzere Türkçe karakterler kullanılmamıştır. Değişken isimleri özgürce seçilebilir. Ancak değişken isimleri verilirken bazı kurallar vardır. Değişken isimleri içinde kesinlikle Türkçe karakter ve noktalama işareti kullanılmamalıdır. Boşluk karakteri içermemelidir. "-" yerine "_" kullanılmalıdır. Sayı ile başlamamalıdır (örnek 2nciadi kullanılamaz) ancak içinde sayı kullanılabilir (örnek adi2nci kullanılabilir). Değişkenlerin ne olduğu konusunda açıklık kazandırmak açısından bazı örnekler aşağıdadır. Her örneği anlamaya çalışınız.

```
PRINT "Deneme"
```

```
Deneme
```

```
PRINT 67
```

```
67
```

```
PRINT 2 + 4
```

```
6
```

```
vize = 40
PRINT vize
```

```
40
```

```
vize = 34
PRINT vize + 2
```

```
36
```

```
sayi1 = 3
sayi2 = 5
sayi3 = 7
PRINT sayi1 + sayi2 + sayi3
```

```
15
```

```
sayi1$ = "24"
sayi2$ = "85"
PRINT sayi1$ + sayi2$
```

```
2485
```

sonuca dikkat ediniz, değişkenler “” dolayı alfanümerik. Matematiksel bir toplama işleminden ziyade iki değişken birbirine uç uca eklendi.

```
vize = 50
final = 65
PRINT vize * 0.4 + final * 0.6
```

```
59
```

Kullanıcı Girişi

Yukarıdaki örnekler incelendiğinde tüm kullanılan değerlerin her bir örnek için sabit olduğu görülmektedir. Program içinde kullanıcının giriş yapılması istenmemekte her seferinde program içine sabit yazılmış olan sayılar için işlem yapılmaktadır. Bu örnekler derlenip çalışır program haline getirildiklerinde kullanıcıdan hiç bir şey istenmeden hepsi aynı sonuçları verecektir. Kullanıcıların program içinde sayıları ve değerleri değiştirmesi gibi bir şey söz konusu değildir. Yukarıda yazılan örnekler program satırlarıdır, kullanıcı girişi yapılması istenmemiştir.

Program çalışırken kullanıcının giriş yapabilmesi için ne yapmak gerekmektedir?

INPUT "ne istendiğine dair açıklayıcı yazı"; <değişken> programın bu noktasında bilgisayar durur, ekrana komutun yanında "" içinde verilen yazıyı yazar ve kullanıcıdan bir değer girmesini ister. Giriş işlemi bittiğinde kullanıcı ENTER tuşuna basar ve girilen değer <değişken> ne atanır.

```
INPUT "vize notunuzu giriniz: ";vize
PRINT vize
```

```
vize notunuzu giriniz: 65 <65 kullanıcı tarafından girilmiştir
65
```

Matematiksel işlemlerde kullanılan karakterler ve fonksiyonlar;
3*2 (çarpma), 3/2 (bölme), 3+2 (toplama), 3-2 (çıkarma), 3^2 (üslü işem) ve 3 mod 2 (bölmede kalan)

Matematik hesaplamalardaki işlem önceliği aynı şekilde programlama için de geçerlidir.
Formül içinde ilk önce çarpma ve bölme işlemleri yapılır daha sonra ise toplama ve çıkartma işlemleri. Çarpma ve bölme aynı önceliğe sahiptir, aynı şekilde toplama ve çıkartma da aynı önceliğe sahiptir. Eğer formül içinde önceliğe sahip olmadığı halde belli bir işleme öncelik tanınmak isteniyorsa ( ) içine alınmalıdır. ( ) içindeki işlemlerin önceliği vardır. Ondalıklı sayılarda ayraç olarak "." kullanılır.

```
INPUT "vize notunuzu giriniz: ";vize
INPUT "final notunuzu giriniz: ";final
PRINT "notunuz"
PRINT vize * 0.4 + vize * 0.6
```

```
vize notunuzu giriniz: 65 <65 kullanıcı tarafından girilmiştir
final notunuzu giriniz: 55 <55 kullanıcı tarafından girilmiştir
notunuz
59
```

```
INPUT "Adınız nedir: ";ad$
PRINT "Merhaba ";
PRINT ad$;
PRINT " nasılsın"
```

```
Adınızı nedir: mehmet <mehmet kullanıcı tarafından girilmiştir
Merhaba mehmet nasılsın
```

Koşullu işlemler

Programın belli bir noktada karar vermesi ve buna göre işlem yapması isteniyorsa kullanılan bir fonksiyondur. Normal koşullar altında programın yukarıdan aşağıya doğru komutları çalıştırdığını öğrenmiştik. Ancak belli bir koşul sağlandığı zaman programın bu yönü değiştirilebilir. Bir örnekle açıklamak gerekirse; öğrenci ortalama notunun 60 ve yukarısı olması durumunda öğrenci dersten geçmekte, aksi halde kalmaktadır. Dikkat edilecek olursa burada bir koşul vardır. Öğrenciye geçtiği mi yoksa kaldığı mı bildirilmesi gerekmektedir? Koşul şudur; notu 60'a eşit veya büyük mü? Bunun tek bir cevabı vardır. Evet veya hayır. Evet ise öğrenciye geçtiği, hayır ise kaldığı bildirilecektir. Komut şu şekildedir.

```
IF ortalamanot >= 60 THEN PRINT "Geçti" ELSE PRINT "Kaldı"
```

Görüldüğü üzere komut 3 kısımdan oluşmaktadır. İlk kısım ***IF*** den sonra yazılan koşul kısmıdır. Bu kısımda belirtilen koşulun cevabı evet ise ***THEN*** kısmı, hayır ise ***ELSE*** kısmı yapılmaktadır. Örnekte ortalamanot değişkeninin değerinin 72 olduğunu

düşünürsek bu koşulun cevabı evet olacaktır ve ***THEN*** kısmı gerçekleştirilerek ekrana "Geçti" ibaresi yazılacaktır. Bu durumda ***ELSE*** kısmı atlanarak işlem görmez. ortalamanot değerinin 40 olduğunu düşündüğümüzde koşul hayır cevabını verecektir ve hayır cevabı verildiğinde işlem yapılan ***ELSE*** kısmı devreye girecektir, ekrana "Kaldı" ibaresi yazılacaktır. Bu durumda da ***THEN*** kısmı işlem görmeden atlanacaktır.

IF <koşul> **THEN** komut **ELSE** komut

koşul gerçekleşirse THEN den sonraki kısımda yer alan, koşul gerçekleşmezse ELSE den sonraki kısımda yer alan komut işlem görmektedir. Aşağıda koşullar verilmiştir;

x>y	x, y den büyükse
x<y	x, y den küçükse
x=y	x, y ye eşitse
x>=	x, y ye eşit veya büyükse
x<=	x, y ye eşit veya küçükse
x<>y	x, y den farklıysa

Bu tek satırlık bir komuttur, eğer ***THEN*** veya ***ELSE*** kısmında yazılacak komut sayısı fazla ise komut aşağıdaki gibi de kullanılabilir.

```
IF <koşul> THEN
    komut
    komut
    komut
    komut
ELSE
    komut
    komut
    komut
END IF
```

Görüldüğü üzere ***THEN*** veya ***ELSE*** kısmına çok sayıda komut bu şekilde eklenebilmektedir. Ancak ***END IF*** ile komutun sonlandığını bildirmek gerekmektedir. Bu durumda END IF konulması unutulmamalıdır. Tek satırlık ***IF*** satırında ***END IF*** kullanımı yoktur.

ELSE kısmının kullanımı opsiyoneldir. Yani kullanılması zorunlu değildir. Bu durumda eğer koşul sağlanıyorsa ***THEN*** kısmı yapılır, koşul sağlanmıyorsa hiç bir şey yapılmaz.

IF <koşul> **THEN** komut

Bu komutun kullanılması ile örnekler verelim

```
vize = 50
final = 70
sonuc = vize * 0.4 + final * 0.6
IF sonuc >= 60 THEN PRINT "Geçtiniz" ELSE PRINT "Kaldınız"
```

Yukarıdaki örnek ekrana “Geçtiniz” yazacaktır. Örneği biraz daha karıştıralım.

```
INPUT "Vize notunuzu giriniz(girmediyseniz boşgeçiniz)";vize
INPUT "Final notunuzu giriniz(girmediyseniz boşgeçiniz)";final
IF vize = 0 THEN PRINT "Vizeye girilmemiş"
IF final = 0 THEN PRINT "Finale girilmemiş"
sonuc = vize * 0.4 + final * 0.6
IF sonuc >= 60 THEN
        PRINT "Geçtiniz"
                ELSE
        PRINT “Kaldınız..!”
        gerekennot = (60 - (vize * 0.4)) / 0.6
        PRINT "Finalden "; gerekennot;" alsaydın geçerdin.”
        PRINT "Hoca sana "; gerekennot - final;
        PRINT "puan daha verseydi geçecektin.”
        PRINT "Bütünlemeden"; gerekennot;" ve üzeri al.”
END IF
```

Yukarıdaki örnek zor bir örnektir. Bilgisayara komutları yazarak denemeniz anlamanıza yardımcı olacaktır.

Döngüsel işlemler

Döngüsel işlemler programcılıkta çok kullanılan fonksiyonlardandır. Aynı işlemin birden fazla kez yapılması için kullanılır. Örneğin adınızı aşağıya doğru 10 kere yazmak isterseniz normalde yazmanız gereken aşağıdaki gibi bir komut dizisidir.

```
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
PRINT "Hüseyin"
```

Ancak bunun yerine PRINT "Hüseyin" komutunun 10 kere tekrarlanması bu garip durumu ortadan kaldıracaktır. Bu amaç için kullanılan komut ***FOR..NEXT*** döngüsüdür.

FOR sayaç= başlangıçdeğeri **TO** bitişdeğeri **STEP** adımdeğeri **.. NEXT**

FOR ve NEXT satırları arası sayaç değeri bitişdeğerine ulaşıncaya kadar tekrar edilir. Bu tekrar sırasında sayaç değeri adımdeğeri kadar arttırılır.

Yukarıdaki örneği bu komut ile yazacak olursak

```
FOR a = 1 TO 10
PRINT "Hüseyin"
NEXT
```

Bu komut dizilimi bir yukarıdaki komut dizilimi ile aynı çıktıyı verecektir. Çalışma sistemi ise şu şekildedir. Bilgisayar ***FOR..NEXT*** arasında döngüye girer ve ***FOR..NEXT*** arasındaki komut ya da komutları uygular. örnekte ***FOR***'un yanındaki "a" değişkendir ve değeri örnekte 1 dir, bu "a" değişkeninin başlangıç değerini vermektedir, ***TO*** dan sonradaki sayıya kadar (örnekte 10) "a" değeri 1 arttırılır, bu değere ulaşıldığında artık ***FOR..NEXT*** arası yapılmaz ve program ***NEXT*** ten sonraki satıra gider ve normal akışına devam eder. Başka bir örnek vererek durumu netleştirelim .

```
PRINT "Sayım başlıyor"
FOR sayac = 3 TO 7
PRINT sayac
NEXT
PRINT "Sayım bitti"
```

```
Sayım başlıyor
3
4
5
6
7
Sayım bitti
```

Örnek incelendiğinde ***FOR..NEXT*** döngüsünde sayaç değişkeninin değeri 3 ten 7 ye 1'ere basamak adımlarla arttırılmış ve döngü içinde sayaç değişkeninin değeri yazdırılmıştır. Normalde artış değeri 1 olarak bilgisayar tarafından sabit olarak alınır. Eğer artış değeri 1 den farklı ise bu ***FOR*** satırının sonuna ***STEP*** yazılarak verilmelidir. Aşağıdaki örneği inceleyiniz.

4 ten başlayarak 2şer 2şer 140'a kadar saymak istiyoruz. Yazacağımız döngü şu şekilde olacaktır.

```
FOR say = 2 TO 140 STEP 2
PRINT say
NEXT
```

Geriye doğru bir sayım söz konusu ise ***STEP*** kısmına - değer verilir. Normalde artış değeri 1 ise ***STEP*** kısmının yazılmasına gerek yoktur, ancak azalma olarak verilecek ise ve azalma değeri 1 olsa bile ***STEP*** kısmına -1 yazılmalıdır. Azalma değeri 3 olmasını istiyorsak ***STEP*** kısmına -3 yazılmalıdır. Aşağıdaki örnekte 10 dan 0 a 2şer 2şer düşülecektir.

```
FOR say = 10 TO 0 STEP -2
PRINT say
NEXT
```

Daha karmaşık bir işlemle ***FOR..NEXT*** döngüsüne örnek verelim. Örneğin sınıf mevcudu belli olmayan bir sınıfta boy ortalamasını almak isteyelim. Fakat kaç kişinin boyunun girilmesi gerektiği sabit değildir. Yani program bize kaç kişinin boyunun hesaplanması gerektiğini sorması gerekmektedir. İlk önce kodu yazalım, daha sonra açıklayalım

```
10 INPUT "Sınıf mevcudu nedir";mevcut
20 toplamboy = 0
30 FOR say = 1 TO mevcut
40 PRINT say;
50 INPUT ". kişinin boyunu giriniz ";boy
60 toplamboy = toplamboy + boy
70 NEXT
80 PRINT "Sınıfın boy ortalaması ";
90 PRINT toplamboy / mevcut
```

Çok karışık bir örnek gibi gözükse de satır satır incelemeye başlayalım. İlk önce satırların başında 10nar 10nar artan satır numaralarından bahsedelim. Bu daha önce görmediğimiz bir şey idi. Bunlara satır numaraları denilmektedir ve kullanılması zorunlu değildir. Komutların yerlerini belirlemek için kullanılmaktadır. Genelde 10nar 10nar arttırılır. Bu eğer araya bir satır eklenecek ise kolaylık sağlaması açısındandır. 10 ile 20 satır arasında isterseniz 15. satırı ekleyebilirsiniz ya da 17. satırı ekleyebilirsiniz. Satır numaralarını isterseniz 1'er 1'er artacak şekilde isterseniz 1000ner 1000ner artacak şekilde verebilirsiniz. Bu tamamıyla programcının tercihine bırakılmıştır. İsterseniz satır numarası yazmak zorunda da değilsiniz. Ancak satır numaralarının olması ileride de anlatılacağı üzere size bazı kolaylıklar sağlayacaktır. Yukarıdaki örnek programın açıklanmasında satır numalaları baz alınacaktır. Örneğimize geri dönecek olursak. 10 nolu satırda sınıfın mevcudu istenmekte ve girilen değer mevcut değişkenine

atanmaktadır. Dikkat ederseniz ***FOR..NEXT*** mevcut kadar dönecek (30. satır) ve bizden mevcut kadar sayısı girişi isteyecektir. Bu kod dizisi içinde alışık olmadığımız bir yapı daha göze çarpmaktadır. O da 60. satırdadır. Dikkat edecek olursanız toplamboy değeri 20. satırda 0 değerini alıyor. Ancak ***FOR..NEXT*** döngüsü içinde bu değerin üzerine toplana toplana geliyor. 60. satırın anlamı şudur toplamboy değişkeninin yeni değeri şu andaki değerine boy değişkenin eklenmiş halidir. Döngü her döndüğünde toplamboy değerine boy değeri eklenecek ve en sonda elde edilen toplamboy değeri girilen tüm boyların toplamı olacaktır. Bilindiği üzere ortalama boytoplamının mevcuda bölünmesi ile bulunmaktadır. 90. satırda toplamboy/mevcut ortalamayı hesaplamakta ve yazmaktadır. Yukarıdaki örnekle ister 10 kişinin ister 10000 kişinin boy ortalaması alınabilmektedir. Görüldüğü üzere döngünün ne kadar döneceği mevcut ile bizden girilmesi istenmektedir.

Alfanümerik fonksiyonlar

Alfanümerik değişkenler için bazı fonksiyonlar bulunmaktadır. Bu fonksiyonlar sayesinde alfanümerik değişkenler içinden bazı karakterleri alıp çıkarmak oldukça kolaydır. Örneğin

```
ad$ = "Mehmet"
a$ = LEFT$(ad$,2)
b$ = RIGHT$(ad$,3)
c$ = MID$ (ad$,2,3)
d = LEN(ad$)
PRINT ad$
PRINT a$
PRINT b$
PRINT c$
PRINT d
```

Mehmet
Me
met
ehm
6

Aşağıda alfanümerik fonksiyonların açıklaması verilmiştir.

LEN(*n$*) Alfasayısal dizinin uzunluğunu verir, LEN(“Mehmet”)=6
LEFT$(*n$,x*) Alfasayısal dizinin solundan *x* kadarını verir, LEFT$(“ali”,2)=“al”
RIGHT$(*n$,x*) Alfasayısal dizinin sağından *x* kadarını verir, RIGHT$(“ali”,2)=“li”
MID$(*n$,x,y*) Alfasayısal dizinin *x* inci sırasından sonra *y* kadar karakteri verir, MID$(“merhaba”,3,4)=“rhab”

Ek1

Color parametresi
color parametresi için ön tanımlı renkler. Örnek: color=”Blue”
AliceBlue, AntiqueWhite, Aqua, Aquamarine, Azure, Beige, Bisque, Black, BlanchedAlmond, Blue, BlueViolet, Brown, BurlyWood, CadetBlue, Chartreuse, Chocolate, Coral, CornflowerBlue, Cornsilk, Crimson, Cyan, DarkBlue, DarkCyan, DarkGoldenRod, DarkGray, DarkGreen, DarkKhaki, DarkMagenta, DarkOliveGreen, Darkorange, DarkOrchid, DarkRed, DarkSalmon, DarkSeaGreen, DarkSlateBlue, DarkSlateGray, DarkTurquoise, DarkViolet, DeepPink, DeepSkyBlue, DimGray, DimGrey, DodgerBlue, FireBrick, FloralWhite, ForestGreen, Fuchsia, Gainsboro, GhostWhite, Gold, GoldenRod, Gray, Green, GreenYellow, HoneyDew, HotPink, IndianRed , Indigo , Ivory, Khaki, Lavender, LavenderBlush, LawnGreen, LemonChiffon, LightBlue, LightCoral, LightCyan, LightGoldenRodYellow, LightGray, LightGreen, LightPink, LightSalmon, LightSeaGreen, LightSkyBlue, LightSlateGray, LightSteelBlue, LightYellow, Lime, LimeGreen, Linen, Magenta, Maroon, MediumAquaMarine, MediumBlue, MediumOrchid, MediumPurple, MediumSeaGreen, MediumSlateBlue, MediumSpringGreen, MediumTurquoise, MediumVioletRed, MidnightBlue, MintCream, MistyRose, Moccasin, NavajoWhite, Navy, OldLace, Olive, OliveDrab, Orange, OrangeRed, Orchid, PaleGoldenRod, PaleGreen, PaleTurquoise, PaleVioletRed, PapayaWhip, PeachPuff, Peru, Pink, Plum, PowderBlue, Purple, Red, RosyBrown, RoyalBlue, SaddleBrown, Salmon, SandyBrown, SeaGreen, SeaShell, Sienna, Silver, SkyBlue, SlateBlue, SlateGray, Snow, SpringGreen, SteelBlue, Tan, Teal, Thistle, Tomato, Turquoise, Violet, Wheat, White, WhiteSmoke, Yellow, YellowGreen

KAYNAKLAR

http://www.ulakbim.gov.tr/dokumanlar/kurslar/html/index.uhtml

http://www.htmldersleri.org/index.php

http://www.w3schools.com/html/default.asp

http://www.alternetwebdesign.com/htmltutorial/index.htm