



PHP

2. Hafta

PHP'de Kod Akışı

ve

Algoritmik Düşünebilmek

Bu derste görülecekler

- Algoritma nedir?
- Neden algoritmik düşünmek zorundayız?
- PHP'de temel yapılar
 - Sabitler, değişkenler
 - if ... elseif ... else yapısı
 - switch yapısı
 - while ve do while yapısı
 - for döngüsü
 - Operatörler
- PHP'de tip kontrolleri nasıl yapılmaktadır?

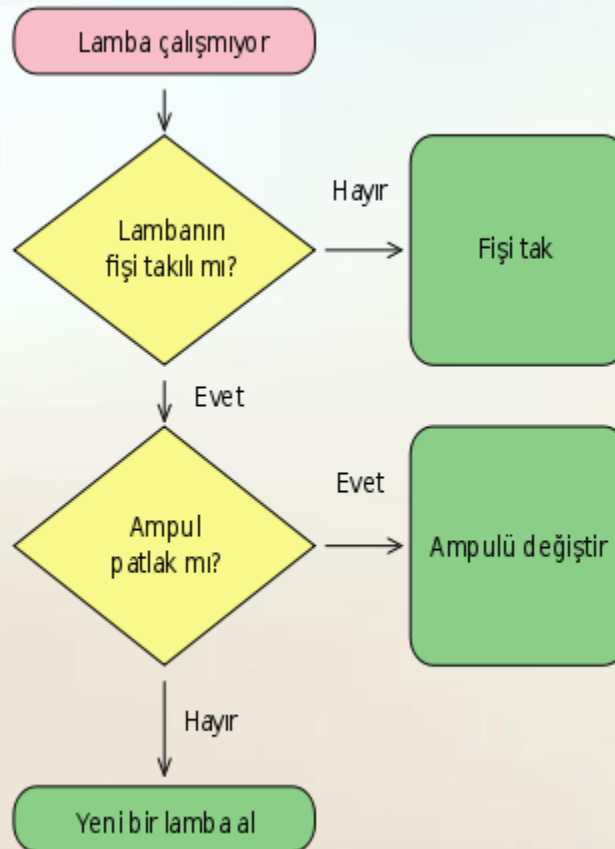
Algoritma nedir?

- Belli bir durumdan başlayarak **sonlu sayıda adımda** belli bir sonucu elde etmenin yöntemini tarif eden iyi tanımlanmış kurallar kümesine algoritma denir.

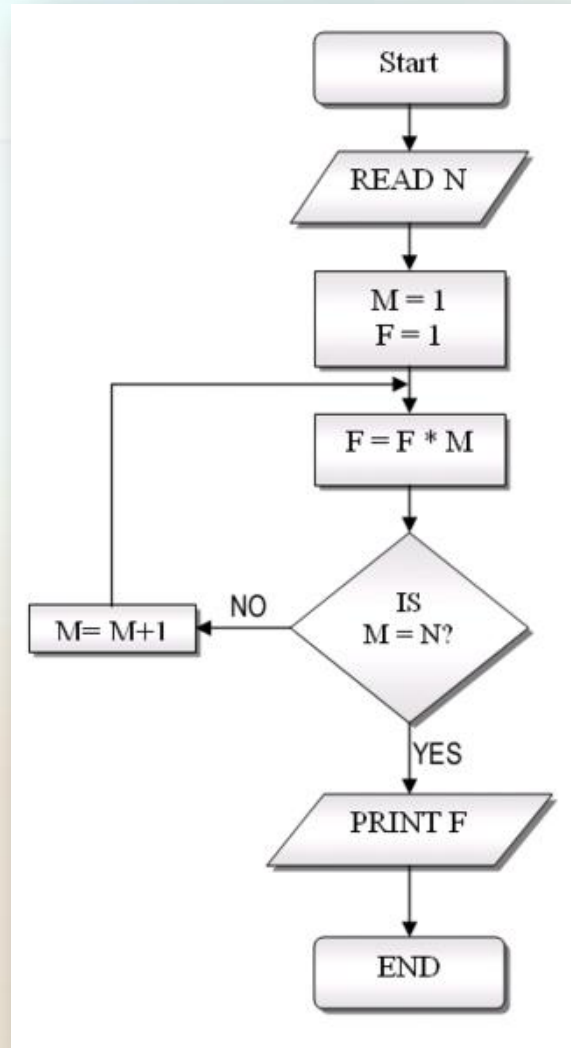
Yani?

- Problemimizi çözmek için adım adım uygulanması gereken işlemler dizisidir. Bu işlemler sonludur ve daha önceden belirlenmiştir.

Örnek: Lambayı yakma algoritması



Örnek: N! Hesaplama Algoritması



Neden algoritmik düşünmek zorundayız?

- Normal bir hayatta sabah uyanır okula gider dersi dinlersiniz. Bu kadar basittir.
- Bir bilgisayar ise böyle düşünemez. Ona her şeyi adım adım anlatmak zorundasınız.
- Bir bilgisayarın sabah uyanıp okula gitmesi için... →

Bilgisayarın okula gidiş hikayesi

- Saati kontrol et. Eğer saat 09.00 ise uyan. Değilse, bir dakika bu adımı tekrarla.
 - Yatakta doğrul. Sol tarafa bak. Eğer kalkacak yer varsa ayağa kalk, yoksa sağ tarafa bak, kalkacak yer varsa ayağa kalk. İkisi de yoksa hata ver.
 - Tuvalete gitmeye hazırlan. Bunun için yatakla tuvalet arasındaki en kısa yolu hesapla.
 - Hesapladığın yoldan tuvalete git. Tuvalet aydınlıksa içeri gir, değilse ışığı aç.
 - Önce soğuk suyu, saniyede yarım litre açacak şekilde aç, daha sonra toplam su sıcaklığı 22 derece olana kadar sıcak su vanasını aç.
 - Avucuna su doldur, yüzüne hızlıca götür ve yüzünü yıka.
- ... (bilgisayar okula gider) ...

Bilgisayarın okula gidiş hikayesi (Devam)

- Ders geçen hafta sıkıcı mıydı? Sıkıcıysa bu hafta da sıkıcı olabilir. İstatistikleri hesapla. Eğer sıkıcı geçeceği tahmin ediliyorsa, arkalarda bir yere otur. Değilse önlerde bir yere otur.
- Oturulacak bölgenin yanında samimi arkadaş var mı? Varsa yanına git. Yanında yer var mı? Yoksa başka samimi arkadaş aramaya başla. Varsa otur. Samimi arkadaş yoksa iki yanı boş bir koltuk seç. Otur.
- Hoca geldi mi kontrol et. Eğer gelmediyse geyik muhabbeti yap. Sonra bu adımı tekrar işle.
- Hoca geldiyse, kafayı sıraya koy ve uyumaya başla.

Düşündüklerimizi PHP’de ifade edebilmek de, en az algoritmik düşünebilmek kadar önemlidir.

Bunu yapabilmek için programlama dilinin bize hangi olanakları sunduğunu iyi bilmek gerekir.

Sabitler

- Sabitler: Deęeri sadece bir defa atanan ve atanan deęer bir daha deęiřtirilemeyen yapılardır.
- define (“SABITADI”, “DEGERI”); řeklinde tanımlanır.
- Sabit adları rakam ile bařlayamaz. Ayrıntılı isimlendirme kuralları ile ilgili kaynak “Okunması Gerekenler” kısmında bulunabilir.

Sabitler (devam)

- Örnek:
 - define (“PI”,3.14);
 - define (“ANTALYA”,07);
- defined (“SABİTADI”); ile sabitin daha önce tanımlanıp tanımlanmadığı kontrol edilebilir.

Değişkenler

- Program akışı boyunca içerisinde veri tutabileceğimiz, bu verileri okuyabileceğimiz ve değiştirebileceğimiz parçacıklardır.
- PHP’de değişkenlerin başında \$ vardır.
- Değişkenler harf ya da _ ile başlar, rakam ile devam edebilir. Daha ayrıntılı değişken isimlendirme kuralları için “Okunması Gerekenler” kısmındaki belgelere bakılabilir.

Değişkenler (devam)

- \$birsayi = 12;
- \$biryazi = “hörölöy”;
- \$baskabiryazi = “12”;

- Başka programlama dilleri bilenlerin fark ettiği üzere, PHP’de değişken türü tanımlamak söz konusu değildir.
- Değişkenin içine ne atarsanız, değişkenin tipi ve harcadığı bellek miktarı ona göre dinamik olarak değişecektir.
- Bu önemli bir kolaylık getirirse de, hata yapmayı ve olan hataları tespit etmeyi zorlaştıracığı da unutulmamalıdır.

Değişkenler (devam)

- \$birsayi = 12;
- \$biryazi = “hörölöy”;
- \$baskabiryazi = “12”;
- Başka programlama dilleri bilenlerin fark ettiği üzere, PHP’de değişken türü tanımlamak söz konusu değildir.
- Değişkenin içine ne atarsanız, değişkenin tipi ve harcadığı bellek miktarı ona göre dinamik olarak değişecektir.
- Bu önemli bir kolaylık getirirse de, hata yapmayı ve olan hataları tespit etmeyi zorlaştıracığı da unutulmamalıdır.
- Veri tiplerinin nasıl kontrol edildiğine sonraki slaytlarda değinilecektir.

if ... elseif ... else yapısı

if (ŞART) {

- Şartı sağlıyorsa yapılacaklar

} elseif (ŞART) {

- Başka bir şartı sağlıyorsa yapılacaklar

} else {

- Hiçbir şartı sağlamıyorsa yapılacaklar

}

şeklinde çalışan bir karar mekanizmasıdır.

if ... elseif ... else yapısı (devam)

Örnek

```
$sayi = 3;  
if ($sayi < 3) {  
    echo "sayi 3'ten kucuk";  
} elseif ($sayi > 3) {  
    echo "sayi 3'ten buyuk";  
} else {  
    echo "sayi 3";  
}
```

switch

```
switch ($gun) {  
  case "1":  
    echo "Pazartesi";  
    break;  
  case "2":  
    ....  
  case "7":  
    ....  
  default:  
    echo "Böyle bir gün yok.";  
}
```

while

```
while (ŞART) {
```

```
... şart sağlandığı sürece tekrarlanacak  
işlemler ...
```

```
}
```

do while

```
do {
```

```
    ... şart sağlandığı sürece tekrarlanacak  
    işlemler ...
```

```
} while (ŞART);
```

for

Bir olayın birden fazla tekrarlanacağı durumlarda kullanılır.

```
for ($degisken = İLK DEĞER; ŞART ; ARTIŞ  
MIKTARI) {  
... şart sağlandığı sürece yapılacak  
işlemler ...  
}
```

Sık Kullanılan Operatörler

Operatör	Anlamı
==	eşitse
!=	eşit değilse
===	değer ve tip olarak eşitse
>	büyükse
<	küçükse
<=	küçük eşitse
>=	büyük eşitse
!	değil (NOT)
.	metin birleştirme
+	toplama;
++	Bir arttırma
--	Bir azaltma
+=	Var olan değere toplamak
-=	Var olan değerden çıkarmak
.=	Var olan metne yenisini eklemek

PHP ve Veri Tipleri

- PHP'de veri tipleri dinamiktir. Siz değişkenin içine ne koyarsanız verinin tipi de o olur.
- Örneğin:
 - `$degisken = 1; /* integer */`
 - `$degisken = 1.0; /* float */`
 - `$degisken = (float)1 /* float */`
 - `$degisken = '1'; /* string */`

PHP ve Veri Tipleri (devam)

- Bir değişkenin hangi tipte olduğu `gettype($degisken)` komutu ile bulunabilir ancak bu kullanım önerilmez.
- Bir değişkenin hangi tipte olduğu (ve bulunup bulunmadığı) “`is`” fonksiyonları ile sorgulanabilir.

PHP ve Veri Tipleri (devam)

- Örnekler

```
$degisken1 = 1;
```

```
if (is_int($degisken1)) {
```

```
    echo "degisken1 integer";
```

```
}
```

isset

- Bir değişkenin olup olmadığını ve varsa içinde değer olup olmadığını denetleyen fonksiyondur.
- Örnek:

```
$degisken = NULL;  
if (isset($degisken)) { echo "degiskenin değeri var."; }
```

NULL da ne?

- NULL: Hiçbir şey. Boşluk. Değersiz. Değeri atanmamış. Kara delik.

Veri Tipi Dönüşümleri

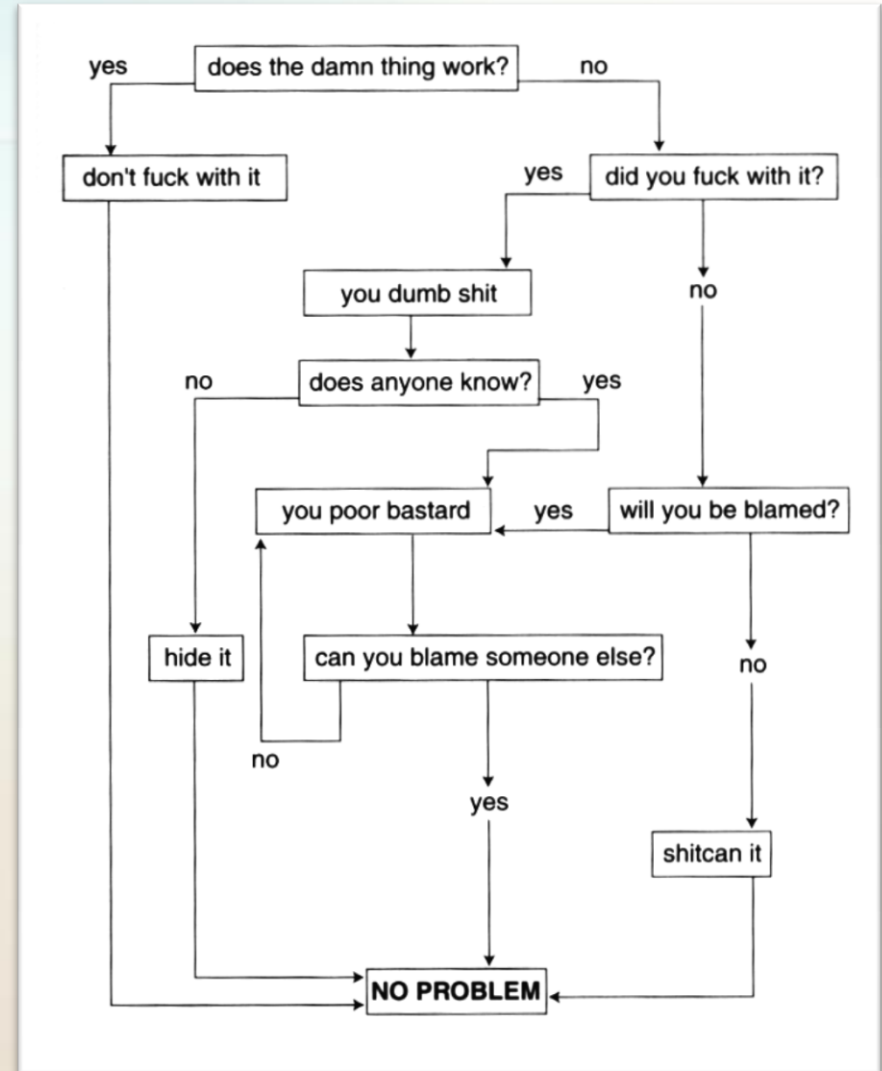
- PHP’de veri tipleri birbirine “casting” ile dönüştürülebilir.
- Örneğin iki sayısal değer arasında dönüşüm aşağıdaki gibi yapılabilir.
 - \$deger = 12.5;
 - \$yenideger = (int)\$deger;
- Yeni değer integer tipinde olacaktır. 12.5 tam sayıya dönüştürülürken bir veri kaybı meydana gelir.

Sormak istedikleriniz?

- Neden sabitlere ihtiyacımız var?
- Neden gettype önerilmez?
- Bir string'i integer'a cast edersek ne olur?

Ek 1: Akış Şemalarıyla Neler Yapılabilir?

- Sorundan kaçma algoritması



Okunması Gerekenler

- PHP'de sabitler:
 - <http://tr.php.net/manual/en/language.constants.php>
- PHP'de deęişkenler:
 - <http://tr.php.net/manual/en/language.variables.basics.php>
- PHP'de akış kontrolörleri:
 - <http://tr.php.net/manual/en/language.control-structures.php>
 - <http://tr.php.net/manual/en/control-structures.for.php>
 - <http://tr.php.net/manual/en/control-structures.if.php>
- PHP'deki operatörler:
 - <http://tr.php.net/manual/en/language.operators.php>
- PHP'de veri tipleri ve dönüşümler:
 - <http://www.php.net/manual/en/language.types.php>
 - <http://php.net/manual/en/language.types.type-juggling.php>