**CSS**

* **[Struktur və qaydalar](http://ilkaddimlar.com/CSS/166/css_1.html" \l "b1.2)**
* [**Daxili stil cədvəlləri**](http://ilkaddimlar.com/CSS/166/css_1.html#b1.3)
* [**Qlobal stil cədvəlləri**](http://ilkaddimlar.com/CSS/166/css_1.html#b1.4)
* [**Bağlanmış stil cədvəlləri**](http://ilkaddimlar.com/CSS/166/css_1.html#b1.5)

**Bir az CSS - dən  
CSS - nədir? Cascading Style Sheets (Kaskadlı Stillərin Cədvəli) - dildir, ixtiyari HTML - sənədin xarici görünüşünü özündə saxlayan xassələr toplusudur. Onun köməkliyi ilə dizayner stil və Web-səhifədəki hər bir element üzərində tam idarəyə malikdir, hansı ki sadə HTML sənədindəki teqlərin daxilində olan xassələrin istifadəsindən daha funksionaldır. Məsələn: Sizə yağlı, qırmızı və altı xətlənmiş mətn yaratmaq lazımdır.  
HTML NÜMUNƏ:  
<font color="red"><strong><u> Hansısa mətn </u></strong></font>  
  
Əgər bu stili bir neçə dəfə istifadə etmək lazımdırsa? Yaxşı əgər 5, bəlkə 10-20 dəfə? Bax burada bizə CSS kömək edəcək. Stil cədvəllərinin üç növü mövcuddur: Daxili stil cədvəlləri, Qlobal stil cədvəlləri və Bağlanmış stil cədvəlləri. Daxili stil cədvəlləri (Inline Style Sheets) xüsisi atributun köməkliyi ilə bir dəfəyə HTML teqlərinə yerləşirlər. Qloballar (Global Style Sheets) elementlərin stilini bütün sənəddə təyin edirlər. Bağlanmışlar isə (Linked Style Sheets) bir dəfəyə bir neçə sənəd üçün istifadə oluna bilərlər və xarici faylda saxlanılırlar. Bütün bunlar haqqında aşağıdakı bölmələrdə ətraflı yazılıb.**[**Yuxarı**](http://ilkaddimlar.com/CSS/166/Bir-az-CSS-den#bashliq)**Struktur və qaydalar  
Selectorlar (Selectors):   
  
Sintaksisi:  
selektor {xassə}  
  
HTML - in ixtiyari elementi - mümkün CSS - selektordur. Selectorun xassəsi təyin olunmuş elementin stilini təyin edir.  
  
NÜMUNƏ:  
  
H1 {color:red; size:20pt;}  
  
Sənəddəki bütün H1 elementləri qırmızı rəngli, 20 nöqtə ölçülü (pt, point) olacaqlar.  
  
Klasslı selektorlar (Class Selectors):  
  
Sintaksisi:  
selektor.klass {klass}  
  
CLASS - HTML elementinin klassını təyin edən atributdur. CSS - də eyni elementlər üçün müxtəlif xüsusi klasslar da yaratmaq olar.  
  
NÜMUNƏ:  
  
H1.blue {color:blue; size:20pt;}  
  
CLASS="blue" olan bütün H1 elementləri göy olacaqlar.  
  
Klasslar həmçinin təyin olunmuş elementlərə aydın bağlantı olmadan da yazıla bilərlər.  
  
Sintaksisi:  
.klass {xassə}  
  
NÜMUNƏ:  
.green {color:green;}  
  
Verilmiş halda CLASS="green" atributunun elementləri yaşıl olacaqlar.  
  
ID selektorlar (ID Selectors):  
  
Sintaksis:  
#id {xassə}  
  
ID - individual adlandırılmış stildir. Onun köməkliyi ilə müəyyən elementin klassına stilli əlavələr etmək olar. ındividuallar bir və ya bir neçə elementin klassına individual xassələr ötürmək üçün istifadə edilir. Deyək ki, siz blue - klassını göy və yanakı etmisiniz. Lakin sizə yağlı xətlənmiş mətn və həmçinin eyni cür xassəli göy yanakı mətn də lazım olacaq. Doğrudur, yeni klass yaratmaq olar, lakin nə üçün? Sadəsi ID yazın. Məsələn "boldunderline", və blue klassının bütün "boldunderline" ID qiymətli elementləri yağlı xətlənmiş göy yanakı olacaqlar. Elə bil ki blue klassı ilə "boldunderline" identifikatorunun xassələrinin sintezi baş verəcək.  
  
NÜMUNƏ:  
  
<html>  
<head>  
<title> Ilk addimlan - CSS numune </title>  
<head>  
<style>  
.blue {color:blue; font-style:italic}  
#boldunderline {text-decoration:underline; font-weight:bold}  
</style>  
<body>  
<p class="blue"> Salam, bu menim web-sehifemdir. </p>  
<p class="blue" id="boldunderline"> Helem ki duzelmek uzredir.... </p>  
<p id="boldunderline"> .....Lakin bu yaxinlarda açılacaq </p>  
</body>  
</html>  
  
Nümunədən göründüyü kimi ID atributu klass olmasa da göstərilə bilər (Nümunədəki paraqrafa bax). Belə halda paraqraf yalnız ID "boldunderline" xassələrinə malik olacaq (nümunədə - yağlı, xətlənmiş mətn).  
  
Kontekstual selektorlar (Contextual Selectors):  
  
Kontekstli selektorlar - bir neçə qəribə selektorun birləşməsidir. Stil yalnız qurulmuş kaskadlı ardıcıllıq ilə elementlərə verilir.  
  
NÜMUNƏ:  
  
P EM {color:silver;}  
  
Verilmiş nümunədə P elementinin daxilindəki bütün EM elementləri verilmiş stilə malik olacaqlar.  
  
Bir neçə elementə eyni xassənin verilməsi:  
  
Deyək ki sizin Web səhifədə bir-neçə elementə eyni xassəni vermək lazımdır. Bu halda selektorları təyin edərkən onlar xassədən qabaq bir-birləri ilə vergül ilə ayrılırlar.  
  
NÜMUNƏ:  
  
h1, h2, h3, p, strong {color: green; font-style:italic;}  
  
Bütün h1, h2, h3, p və strong elementəri yaşıl olacaqlar.  
  
Saxtaklass və saxtaelementlər:  
  
Sintaksisi:  
selektor:saxtaklass {xassə}  
selector.klass:saxtaklass {xassə}  
selektor:saxtaelement {xassə}  
selector.klass:saxtaelement {xassə}  
  
Saxtaklass və saxraelementlər - CSS-ə məxsus və brouzerləri ilə avtomatik təyin edilən xüsusi klass və elementlərdir. Saxtaklasslar bir elementin müxtəlif tiplərini ayırır və onları təyin edərkən hər biri üçün xüsusi stil yaradırlar. Saxtaelementlər digər elementlərin hissələrinə o elementin stilindən fərqli bir stil verərək onların bir hissəsi olurlar.  
  
Saxtaklass və saxtaelementlətin siyahısı:  
  
Anchor Pseudo Classes - Bunlar bağlantı bildirən <a href=" "> - elementinin saxtaklasslarıdır. Bu elementin saxtaklassları: link (bağlantı), active (aktiv bağlantı), visited (əvvəl baxılmış URL), hover (mausun kursorunu bağlantının üzərinə gətirəndə baş verən saxtaklass. Netscape-də işləmir).  
  
FirstLine Pseudo-element - first-line. Bu saxtaelement block-level elementləri ilə istifadə oluna bilər (p, h1 və s.). O bu elementlərin ilk sətrinin stilini dəyişir.  
  
FirstLetter Pseudo-element - first-letter. first-line oxşayır lakin sətirə yox ancaq ilk simvola təsir edir.  
  
NÜMUNƏ:  
  
a:link, a:visited {color:blue}  
a:active {color:red}  
a:hover {text-decoration:none}  
  
Verilmiş nümunədə Anchorun (bağlantılar) bütün elementləri göy olacaq. Sıxıldıqda (aktiv vəziyyətdə) rənglərini qırmızıya dəyişəcəklər və mausu üzərinə gətirəndə xətti çəkiləcək.  
  
Qeyd : Bir selektor üçün bir neçə xassə təyin edərkən, kontekstual selektor, klass, individual adlandırılmış stil və birləşmiş selektor qrupları bir-birindən nöqtə vergül ilə ";" ayrılırlar.**[**Yuxarı**](http://ilkaddimlar.com/CSS/166/Bir-az-CSS-den#bashliq)**Daxili Stil Cədvəlləri  
  
Deyildiyi kimi Daxili stillərin istifadəsi sadə HTML teqlərinin istifadəsindən heç də çox fərqlənmirlər. Onlar HTML - teqində STYLE atributunun köməkliyi ilə yalnız bir elementin stilini təyin edirlər.  
  
HTML NÜMUNƏ:  
  
< font color="blue" size="3" face="Arial"> Mətn </font>  
  
INLINE STYLE SHEET NÜMUNƏ:  
  
< font style="colo:blue; font-size:12pt; font-family:Arial"> Mətn </font>  
  
Göründüyü kimi Inline Style Sheet kodu HTML teqinin kodundan çox alındı. Ona görə də Inline Style Sheet-ləri yalnız HTML-də realizə olunmayan yalnız CSS klassifikasiyasında olan və hansısa teqə tətbiq etmək lazım olarkən istifadə etmək lazımdır. Və ya verilmiş elementin bütünlükdə vəziyyətini dəyişmək lazım olanda da istifadə etmək olar.**[**Yuxarı**](http://ilkaddimlar.com/CSS/166/Bir-az-CSS-den#bashliq)**Qlobal Stil Cədvəlləri  
Qlobal stillər bütün sənədlərin elementlərinin görünüşünü təyin edirlər. Bunun üçün <STYLE type="text/css"> teqindən istifadə edilir. O sənədin başlığında yerləşdirilir.  
NÜMUNƏ:  
<html>  
<head> <title> Ilk addimlar - CSS nümunə </title>  
</head>  
<STYLE type="text/css">  
h1{color:red; font-style:italic; font-size:32px}  
.blue{color:blue}  
#bold{font-weight:bold}  
</STYLE>  
<body>  
<h1> Bu başlıq iri qirmizi yanakı yazılıb </h1>  
Bax <font class="blue"> bu </font> Söz - goydur, <font id="bold">   
bu ise </font> - yağlıdır.  
</body>  
</html>  
Verilmiş nümunədə bütün H1 elementləri iri qırmızı və yanakı yazılacaqlar, bütün Blue klassı ilə yazılan elementlər göy yazılacaqlar, ID="Bold" identifikatoru ilə yazılmış elementlərin hamısı yağlı olacaqlar. Sadəlik üçün <Style type="text/css"> yerinə <Style> teqini də istifadə etmək olar, bu daha da savadlı olacaq.**[**Yuxarı**](http://ilkaddimlar.com/CSS/166/Bir-az-CSS-den#bashliq)**Bağlanmış Stil Cədvəlləri  
Bağlanmış stil cədvəlləri bir stili bir neçə sənədə ötürmək üçün istifadə olunurlar və onlar ayrıca faylda saxlanılırlar. Bu çox cəlbedicidir, belə ki əgər saytda bir stili təyin etmək üçün hər bir sənədə stilləri tətbiq etmək lazım gəlmir.  
  
NÜMUNƏ:  
  
styles.css faylı  
<STYLE type="text/css">  
body {background:black; font-size:9pt; color:red; font-family:Arial Black}  
.base{color:blue; font-style:italic}  
h1 {color:white}  
#bold {font-weight:bold}  
</STYLE>  
HTML faylının özündə isə <Link> teqinin köməkliyi ilə həmin fayla bağlantı qoyulur. Bu belə görünür: <LINK rel="STYLESHEET" TYPE="text/css" HREF="fayla qədər yol">  
  
NÜMUNƏ:  
  
index.html faylı  
  
<html>  
<head>  
<title> Ilk addimlar - CSS numune </title>  
</head>  
<LINK rel="stylesheet" type="text/css" href="styles.css">  
<body>  
Sənədin tutumu  
</body>  
</html>  
  
Bununla da I hissə sona çatır, II hissəyə keçək.**[**Yuxarı**](http://ilkaddimlar.com/CSS/166/Bir-az-CSS-den#bashliq)

**CSS - İN XASSƏLƏRİ**

**II Hissə: CSS - in xassələri**

* [**Font xassəsi**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#b2.1)
* [**Text xassəsi**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#b2.2)
* [**Color və Background xassələri**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#b2.3)
* [**Box xassəsi**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#b2.4)
* [**Təsnifat**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#b2.5)

**Font xassəsi**

* **font-family**

**Mümkün qiymətləri:**

**[1] istənilən font  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Bu xassə element tərəfindən istifadə olunan fontu təyin edir. Əgər URL təyin olunarsa onda həmin font avtomatik istifadəçinin kompüterə yüklənəcək.  
  
NÜMUNƏ:  
  
font-family: Arial Black URL ('arialblack.ttf')**

* **font-style**

**Mümkün qiymətləri:  
[1] normal - dəyişikliksiz  
[2] italic - yanakı  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Elementin stili. Yanakı və ya sadə.  
  
NÜMUNƏ:  
  
font-style: italic**

* **font-variant**

**Mümkün qiymətləri:  
[1] normal - dəyişikliksiz  
[2] small-caps - Bütün kiçik hərfləri böyüklərə dəyişir  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Fontun görünüş variantı. Bu xassəni Netscape dətəkləmir.  
  
NÜMUNƏ:  
  
font-variant: small-caps**

* **font-weight**

**Mümkün qiymətləri:  
[1] normal - dəyişikliksiz  
[2] bold - yağlı  
[3] bolder - çox yağlı (MSIE - də bolddan, Netscape -də isə normaldan fərqlənmir)  
[4] lighter - incə (normaldan fərqlənmir)  
[5] 100 dən 900-a dək ixtiyari qiymət  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Elementin seçilməsi (yağlılığı).  
  
NÜMUNƏ:  
  
font-weight: bold**

* **font-size**

**Mümkün qiymətləri:  
[1] ölçü(+)  
[2] xx-small, x-small, medium, large, x-large, xx-large - bu qiymətlərdən ixtiyarısı  
[3] smaller, larger - bu qiymətlərdən ixtiyarısı  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Fontun ölçüsü.  
  
NÜMUNƏ:  
  
font-size: 30pt**

* **font**

**Mümkün qiymətləri:  
[1] font-family  
[2] font-style  
[3] font-variant  
[4] font-weight  
[5] font-size  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Yuxarıda qeyd olunmuş xassələri istifadə edir..  
  
NÜMUNƏ:  
  
font: italic bolder Arial 12pt**[**Yuxarı**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#bashliq)**Text xassəsi**

* **word-spacing**

**Mümkün qiymətləri:**

**[1] uzunluq(+)  
[2] normal - dəyişikliksiz  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Sözlər arasındakı məsafə. Netscape və MSIE-də istifadə olunmur.  
  
NÜMUNƏ:  
  
word-spacing: 0.4em**

* **text-decoration**

**Mümkün qiymətləri:  
[1] none - yox  
[2] underline - altı xətli  
[3] overline - üzəri xətli (Netscape dəstəkləmir)  
[4] line-through - pozulmuş  
[5] blink - parıldayan (IE dəstəkləmir)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Mətnin bəzənməsi.  
  
NÜMUNƏ:  
  
text-decoration:line-through**

* **letter-spacing**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] normal - dəyişikliksiz  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Həriflər arasında məsafə. Netscape-də işləmir.  
  
NÜMUNƏ:  
  
letter-spacing: 100**

* **vertical-align**

**Mümkün qiymətləri:  
[1] baseline  
[2] sub  
[3] super  
[4] top-text  
[5] top  
[6] middle  
[7] bottom  
[8] bottom-text  
[9] faiz  
  
\*İstifadəsi: inline elementlər üçün  
  
Təsviri: Bir cərgədə dayanmış elementlərin digər elementlərlə münasibətdə mövqeyini təyin edilməsi. Netscape-də işləmir.  
  
NÜMUNƏ:  
  
vertical-align:top-text**

* **text-transform**

**Mümkün qiymətləri:  
[1] none - yox  
[2] Capitalize - Hər söz iri hərflə başlayır  
[3] UPPERCASE - mətnin bütün hərfləri iri olur  
[4] lowercase - mətnin bütün hərfləri kiçik olur  
  
\*İstifadəsi: inline elementlər üçün  
  
Təsviri: Mətnin dəyişməsi. Netscape-də işləmir.  
  
NÜMUNƏ:  
  
text-transform:capitalize**

* **text-align**

**Mümkün qiymətləri:  
[1] left - mətn solda  
[2] right - mətn sağda  
[3] center - mətn mərkəzdə  
[4] justify - dartılmış mətn  
  
\*İstifadəsi: block-level elementlər üçün  
  
Təsviri: Mətnin vəziyyəti.  
  
NÜMUNƏ:  
  
text-align:right**

* **text-indent**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[4] faiz(+)  
  
\*İstifadəsi: block-level elementlər üçün  
  
Təsviri: boşluq.  
  
NÜMUNƏ:  
  
text-indent:30 em**

* **line-height**

**Mümkün qiymətləri:  
[1] normal - dəyişikliksiz  
[2] uzunluq(+)  
[3] faiz  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Yuxarıdan boşluq.  
  
NÜMUNƏ:  
  
line-height:100%**[**Yuxarı**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#bashliq)**Color və Background xassələri**

* **color**

**Mümkün qiymətləri:  
[1] rəng(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: rəng.  
  
NÜMUNƏ:  
  
color:#f00000**

* **background-color**

**Mümkün qiymətləri:  
[1] rəng(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Elementin fonunun rəngi.  
  
NÜMUNƏ:  
  
background-color:#f00000**

* **background-image**

**Mümkün qiymətləri:  
[1] none - yox  
[2] URL(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Fon şəkilləri.  
  
NÜMUNƏ:  
  
background-image:URL(cool.gif)**

* **background-repeat**

**Mümkün qiymətləri:  
[1] repeat - fon şəkillərini bütün istiqamətlərdə çoxaldır  
[2] repeat-x - fon şəkillərini horizontal istiqamətlərdə çoxaldır  
[3] repeat-y - fon şəkillərini vertikal istiqamətlərdə çoxaldır  
[4] no-repeat - təkrar olunmayan şəkillər  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: fonun şəklinin təkrarlanması.  
  
NÜMUNƏ:  
  
background-repeat:no-repeat**

* **background-atachment**

**Mümkün qiymətləri:  
[1] scrool - fon şəkli sənə ilə birlikdə fırlanır  
[2] fixed - Fırlanmır. Bir yerdə dayanır. Netscape-də işləmir.  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri:fon şəklinin fırlanma imkanları .  
  
NÜMUNƏ:  
  
background-attachment:fixed**

* **background-position**

**Mümkün qiymətləri:  
[1] enindən faizlər+ hündürlükdən faizlər(+)  
[2] top, middle, bottor - qiymətlərindən biri  
[3] left, center, right - qiymətlərindən biri  
[4] sol tərəfdən məsafə + başdan məsafə  
  
\*İstifadəsi: block-level və replased elementlər üçün  
  
Təsviri: fon şəklinin mövqesi (background-repeat repeat-x, repeat-y və ya no-repeat - a bərabər olanda işləyir).  
  
NÜMUNƏ:  
  
background-position:50%0%**

* **background**

**Mümkün qiymətləri:  
[1] background-color  
[2] background-image  
[3] background-position  
[4] background-attachment  
[5] background-repeat  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələrin ümumiləşməsi.  
  
NÜMUNƏ:  
  
background:no-repeat black fixed 50%0%**[**Yuxarı**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#bashliq)**Box xassəsi**

* **margin-top**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
[3] auto - avtomatik  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıdan boşluq təyin edir.  
  
NÜMUNƏ:  
  
margin-top:100**

* **margin-right**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
[3] auto - avtomatik  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: sağdan boşluq təyin edir.  
  
NÜMUNƏ:  
  
margin-right:100%**

* **margin-bottom**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
[3] auto - avtomatik  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: aşağıdan boşluq təyin edir.  
  
NÜMUNƏ:  
  
margin-bottom:100em**

* **margin-left**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
[3] auto - avtomatik  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: soldan boşluq təyin edir.  
  
NÜMUNƏ:  
  
margin-left:100pt**

* **margin**

**Mümkün qiymətləri:  
[1] margin-top  
[2] margin-right  
[3] margin-left  
[4] margin-bottom  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələrin ümumiləşdirir.  
  
NÜMUNƏ:  
  
margin:100pt**

* **padding-top**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarı borderdən boşluq.  
  
NÜMUNƏ:  
  
padding-top:100pt**

* **padding-right**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: sağ borderdən boşluq.  
  
NÜMUNƏ:  
  
padding-right:100%**

* **padding-bottom**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: aşağı borderdən boşluq.  
  
NÜMUNƏ:  
  
padding-bottom:100em**

* **padding-left**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: sol borderdən boşluq.  
  
NÜMUNƏ:  
  
padding-left:100**

* **padding**

**Mümkün qiymətləri:  
[1] padding-top  
[2] padding-right  
[3] padding-left  
[4] padding-bottom  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: Yuxarıda qeyd olunmuş xassələri ümumiləşdirir. Eyni vaxtda müxtəlif istiqamətlər üçün bir neçə qiymət (dördə qədər) təyin etmək olar. Əgər bir qiymət təyin olunubsa - onda bütün istiqamətlər üçün eyni bir boşluq olur, əgər ikidirsə - onda bitişik olan tərəflər üçün müxtəlif boşluqlar olur, və əgər dörd qiymətdirsə - onda bütün tərəflər üçün ayrı-ayrı boşluqlar təyin olunur.  
  
NÜMUNƏ:  
  
padding:100px**

* **border-top-width**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] thin, medium və ya thick  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarı borderin qalınlığı.  
  
NÜMUNƏ:  
  
border-top-width:100pt**

* **border-right-width**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] thin, medium və ya thick  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: sağ borderin qalınlığı.  
  
NÜMUNƏ:  
  
border-right-width:thick**

* **border-bottom-width**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] thin, medium və ya thick  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: aşağı borderin qalınlığı.  
  
NÜMUNƏ:  
  
border-bottom-width:100em**

* **border-left-width**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] thin, medium və ya thick  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: sol borderin qalınlığı.  
  
NÜMUNƏ:  
  
border-left-width:medium**

* **border-width**

**Mümkün qiymətləri:  
[1] border-top-width  
[2] border-right-width  
[3] border-left-width  
[4] border-bottom-width  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: borderlərin qalınlıqları. Eyni vaxtda müxtəlif borderlər üçün bir neçə qiymət (dördə qədər) təyin etmək olar. Əgər bir qiymət təyin olunubsa - onda bütün istiqamətlər üçün bir qalınlıq olur, əgər ikidirsə - onda bitişik olan tərəflər üçün müxtəlif qalınlıqlar olur, və əgər dörd qiymətdirsə - onda bütün tərəflər üçün ayrı-ayrı qalınlıqlar təyin olunur.  
  
NÜMUNƏ:  
  
border-width:15pt**

* **border-color**

**Mümkün qiymətləri:  
[1] rəng(+)  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: borderin rəngi. Netscape-də işləmir.  
  
NÜMUNƏ:  
  
border-color:green**

* **border-style**

**Mümkün qiymətləri:  
[1] none  
[2] dotted, dashed, solid, double, groove, ridge, inset, outset  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: borderlərin stilləri. Eyni vaxtda müxtəlif borderlər üçün bir neçə qiymət (dördə qədər) təyin etmək olar. Əgər bir qiymət təyin olunubsa - onda bütün istiqamətlər üçün eyni bir stil verilir, əgər ikidirsə - onda bitişik olan tərəflər üçün müxtəlif stillər olur, və əgər dörd qiymətdirsə - onda bütün tərəflər üçün ayrı-ayrı stillər təyin olunur.  
  
NÜMUNƏ:  
  
border-style:dotted groove**

* **border-top**

**Mümkün qiymətləri:  
[1] border-top-width  
[2] border-style  
[3] border-color  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələri yuyxarı border üçün ümumiləşdirir.  
  
NÜMUNƏ:  
  
border-top: 100em red groove**

* **border-right**

**Mümkün qiymətləri:  
[1] border-right-width  
[2] border-style  
[3] border-color  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələri sağ border üçün ümumiləşdirir.  
  
NÜMUNƏ:  
  
border-right: 5pt magenta solid**

* **border-left**

**Mümkün qiymətləri:  
[1] border-left-width  
[2] border-style  
[3] border-color  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələri sol border üçün ümumiləşdirir.  
  
NÜMUNƏ:  
  
border-left: 15pc coral inset**

* **border-bottom**

**Mümkün qiymətləri:  
[1] border-bottom-width  
[2] border-style  
[3] border-color  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələri aşağı border üçün ümumiləşdirir.  
  
NÜMUNƏ:  
  
border-bottom: 30 orange outset**

* **border**

**Mümkün qiymətləri:  
[1] border-width  
[2] border-style  
[3] border-color  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: yuxarlda qeyd olunmuş xassələri ümumiləşdirir.  
  
NÜMUNƏ:  
  
border: thik black double**

* **width**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: block-level və replased elementlər üçün  
  
Təsviri: elementin eni.  
  
NÜMUNƏ:  
  
width: 10%**

* **height**

**Mümkün qiymətləri:  
[1] uzunluq(+)  
[2] faiz(+)  
  
\*İstifadəsi: block-level və replased elementlər üçün  
  
Təsviri: elementin hündürlüyü.  
  
NÜMUNƏ:  
  
height: 100pt**

* **float**

**Mümkün qiymətləri:  
[1] left - soldan  
[2] right - sağdan  
[3] none - susmaya görə  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: elementin yerləşməsi.  
  
NÜMUNƏ:  
  
float:right**

* **clear**

**Mümkün qiymətləri:  
[1] left - soldan  
[2] right - sağdan  
[3] both - iki tərəfdən  
[4] none - susmaya görə  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: elementlərin ətrafında yerləşməsi.  
  
NÜMUNƏ:  
  
clear:both**[**Yuxarı**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#bashliq)**Təsnifat**

* **display**

**Mümkün qiymətləri:  
[1] none - görsənmir  
[2] block - sətiri elementdən qabaq və sonra bölür (yəni, element eyni xətdə olan digər elementlərlə bir xətdə ola bilməz)  
[3] inline - sətiri bölmür  
[4] list-item - sətiri elementdən qabaq və sonra bölür + list-item elementlərdəki kimi maker əlavə edir  
  
\*İstifadəsi: Bütün elementlər üçün  
  
Təsviri: elementin necə görünəcəyini təyin edir.  
  
NÜMUNƏ:  
  
display:none**

* **while-space**

**Mümkün qiymətləri:  
[1] normal - yanakı gedən bir-neçə məsafəni sıxır  
[2] pre - yanakı gedən məsafələrin görsənməsinə icazə verir  
[3] nowrap - <BR> sətirlərindən keçidə icazə vermir  
  
\*İstifadəsi: block-level elementlər üçün  
  
Təsviri: elementlər arasındakı məsafələrin necə görünəcəyini təyin edir.  
  
NÜMUNƏ:  
  
while-space:nowrap**

* **list-style-type**

**Mümkün qiymətləri:  
[1] disc, circle, square, decimal, lower-alpha və ya upper-alpha  
[2] none - heç bir  
  
\*İstifadəsi: display-a bərabər olan list-item elementlər üçün  
  
Təsviri: list-item in makerinin görünüşünü təyin edir. Əgər list-style-image dəyişəni none-yə bərabərdirsə və ya təyin olunmubsa  
  
NÜMUNƏ:  
  
list-style-type:lower-alpha**

* **list-style-image**

**Mümkün qiymətləri:  
[1] none - yox  
[2] URL (+)  
  
\*İstifadəsi: display-a bərabər olan list-item elementlər üçün  
  
Təsviri: list-item makerinin görünüşünü şəkil kimi təyin edir.  
  
NÜMUNƏ:  
  
list-style-image:URL(cool.gif)**

* **list-style-position**

**Mümkün qiymətləri:  
[1] inside - növbəti sətirə keçəndə boşluqsuz görünəcək.  
[2] outside - susmaya görə  
  
\*İstifadəsi: display-a bərabər olan list-item elementlər üçün  
  
Təsviri: list-item elementdə aslı olan elementin makerinin duruşunu təyin edir.  
  
NÜMUNƏ:  
  
list-style-position:inside**

* **list-style**

**Mümkün qiymətləri:  
[1] list-style-type  
[2] list-style-position  
[3] list-style-image  
  
\*İstifadəsi: display-a bərabər olan list-item elementlər üçün  
  
Təsviri: yuxarıda qeyd olunmuş xassələri ümumiləşdirir  
  
NÜMUNƏ:  
  
list-style:inside**[**Yuxarı**](http://ilkaddimlar.com/CSS/171/CSS-in-xasseleri#bashliq)

**III Hissə: Əlavələr**

* [**Uzunluq ölçüləri**](http://ilkaddimlar.com/CSS/177/Elaveler#b3.1)
* [**Faiz ölçüləri**](http://ilkaddimlar.com/CSS/177/Elaveler#b3.2)
* [**Rənglər**](http://ilkaddimlar.com/CSS/177/Elaveler#b3.3)
* [**Bağlantılar**](http://ilkaddimlar.com/CSS/177/Elaveler#b3.4)

**Uzunluq ölçüləri  
  
Sintaksisi: "+" (yazmamaq da olar) və ya "-" sonra isə [rəqəm] və ona birləşik olan [uzunluq vahidi]  
  
NÜMUNƏ:  
  
-558pt  
  
Uzunluq vahidləri: ex - x-height, font ilə istifadə olunan elementin "x" eni  
px - pixels, piksellər   
in - inches, düymlər  
cm - centimeters, santimetrlər  
mm - millimetrs, millimetrlər  
pt - points, nöqtə (1pt = 1/72in)  
pc - picas (1 pc = 12pt)**[**Yuxarı**](http://ilkaddimlar.com/CSS/177/Elaveler#bashliq)**Faiz ölçüləri  
  
Sintaksisi: "+" va ya "-" sonra [rəqəm] əlavə "%" (ara məsafəsiz)  
  
NÜMUNƏ:  
  
-156%**[**Yuxarı**](http://ilkaddimlar.com/CSS/177/Elaveler#bashliq)**Rənglər  
  
Sintaksisi: [color]  
  
NÜMUNƏ:  
  
magenta  
  
Rəng dəyişəni onun adı (red, blue black və s.) və ya RGB dəyişəni ola bilər  
  
RGB (red green blue) rənglərinin təsviri növləri :**

* **#rrggbb (məsələn #00CC00)**
* **rgb(x,x,x) - harda lki "x" 0 ilə 255 arasında ədəddir (məsələn, rgb(0,204,0))**
* **#rgb (məsələn, #0C0)**
* **rgb (x%,x%,x%) - harda ki "x" 0.0 dan 100.0-a qədər qiymətlər alır (məsələn, 0%,80%,0%)**

**Bütün nümunələr eyni rəngi təsvir edirlər**[**Yuxarı**](http://ilkaddimlar.com/CSS/177/Elaveler#bashliq)**Bağlantılar  
  
Sintaksisi: "URL", sonra isə dırnaq içində bağlantı. Bağlantını həmçinin bir və ya ikili dırnaqların arasına salmaq olar.  
  
NÜMUNƏ:  
  
URL('cool.gif')**[**Yuxarı**](http://ilkaddimlar.com/CSS/177/Elaveler#bashliq)

**GİRİŞ. FREYMLƏR NƏ ÜÇÜN LAZIMDIR**

**Nəhayət, biz freymlərə (frames) gelib çatdıq. Onlar nedirlər və hansı üstünlükləri var? Freymlər bizə brauzerin pəncərəsində eyni vaxtda bir neçə sənəd açmağa imkan verir. (Məsələn, içində menyu olan "menu.html" sənədi, içində loqo olan "logo.html" sənədi, səhifənin yuxarısı və saytımızın məzmunu olan "content.html" sənədi).   
  
Freymləri müxtəlif cür tətbiq etmək olar. Bəzi web-ustalar freymlərlə öz ağılsız fikirlərini əks etdirir və istifadəçilərini məyus edirlər, bəziləri məcburiyyət qarşısında qalanda freymlərə müraciət edirlər. Bəziləri isə onlar üçün rahat olduğuna görə freym işlədirlər. Əsas odur ki, freym dəhşətli görünməsin və istifadəçi üçün rahat olsun.  
  
Siz müxtəlif forumlarda və ya başqa yerlərdə freymlərin pis olduğunu eşidə bilərsiniz. Təbii olaraq sual verəcəksiniz: "*Freymlər yaxşıdır, yoxsa pis?*"   
  
Baxır freymlərə necə yanaşılır. Məncə bu söz-söhbət heç vaxt bitməyəcək. Yeqin bu ondan irəli gəlir ki, freymlərin yaxşı cəhətlərindən başqa, həmçinin ciddi nöqsanları da var. Buna görə də onların tətbiqinin bütün tərəflərini öyrənmək məntiqə uyğun olardı.   
  
Amma mən freymlərlə bağlı bütün problemlərdən və onların həllindən uzun-uzadı danışmayacağam.  
  
Mənim şəxsi fikrim budur ki, freymlər yaxşıdır, hətta çox yaxşıdır, əgər onlar yerində və ağılla işləniblərsə, estetik və gözəl görünürlərsə. "Ağılla" o deməkdir ki, biz sənəddə freym işlədiriksə, HTML-in imkanları ilə kifayətlənmirik, biz həm də freymlərin düzgün işi üçün scriptlər də tətbiq edirik. Amma bizim işimiz freymləri standart üsullarla öyrənmək olacaq.  
  
Beləliklə, bizim sevimli brauzerimizin eyni vaxtda bir neçə sənədi göstərməsi üçün xüsusi freym-sənəd yaratmaq lazımdır. Bu sənəddə biz brauzer pəncərəsində neçə sənədin açılacağını, hər açılan sənədin nə qədər yer tutacağını, bir-birinə nisbətdə necə yerləşəcəyini təyin edirik. Amma hər şey danışdığımız kimi elə də sadə və asan deyil:).   
  
Saytda istifadəçiyə ilk göstərilən sənəd adətən index.html (ya da nəsə başqa sənəd) olur, ona görə də biz ona "əziyyət" verəcəyik, qoy freym-sənəd o olsun. Beləliklə, index.html sənədi yaradaq:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> </html>** |

**İlk baxışdan standart başlanğıcdır, amma... "body" teqi yoxdur?! Bu səhv deyil, freym-sənəddə "body" teqi olmur.   
  
"body" teqinin yoxluğu sizi narahat etməsin, onun layiqli əvəzini taparıq:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset></frameset> </html>** |

**Salam "frameset", necə də gözəl addır. Deyirlər siz qərara almısınız ki, bizim saytda nəyisə deyişəsiniz və freymlkər yaradasınız :), çox yaxşı, biz həmişə yenilikləri xoşlayırıq.  
  
İndi "frameset" üçün layiqli köməkçılər tapaq və ona zəruri atributlar verək. Amma növbəti addımda.**

**FREYM YARATMAĞI ÖYRƏNİRİK**

**İşə başlamazdan əvvəl, müəyyənləşdirmək lazımdır ki, biz sənədimizi hansı prinsiplə yerləşdirəcəyik. Yəni biz hansı sənədləri eyni vaxtda ziyarətçilərə göstərəcəyik. Klassik variantı təklif edirəm - logo.html, menu.html, content.html - bunlar haqqında mən sizə əvvəlki dərsdə danışmışdım.   
  
Bunları biz müxtəlif üsulla yerləşdirə bilərik. Budur, nümunə üçün 4 varianta baxın:**

****

**Freymin köməyi ilə biz brauzeri verilmiş ölçülərlə bir neçə pəncərəyə bölüb, onlarda bir-birindən asılı olmayan sənədlər yerləşdiririk. Bu bizə imkan verir ki, bəzi informasiya saytda həmişə görünsün, eyni vaxtda da başqa bir informasiya müxtəlif şəkildə dəyişilsin. (Məsələn, biz menyunu və loqonu həmişə görəcəyik, amma 3-cü pəncərə, ondakı bağlantılara sıxdıqda uyğun olaraq müxtəlif mətn, şəkil və s. olan sənədləri yükləyəcək.  
  
Yeni ki, bizim səhifədə müstəqil freym kimi yerləşdirdiyimiz hər bir sənəd özündə bir çox məlumat saxlaya bilər (mətn, şəkil, cədvəl və s.). Niyə mən bütün bunları vurğulayıram, çünki bəzi insanlar nəyə görəsə elə bilirlər ki, freymlər cədvəlin alternatividir. Bu düz deyil!  
  
Beləliklə, əvvəlcə bu variantı yaradaq (qabaqcadan özündə loqo, menyu və əsas məzmunu saxlayan adi \*.html sənədlər yaratmağı unutmayın):**

****

|  |
| --- |
| **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset rows="100,\*,150"> <frame src="logo.html"> <frame src="content.html"> <frame src="menu.html"> </frameset> </html>** |

[**Nümunə**](http://ilkaddimlar.com/samples/12019/105.html)

**Gəlin ayırd edək ki, nə nə üçündür. <frameset> teqinə verdiyimiz "rows" atributu ilə göstəririk ki, bzim freymlər üfiqi olaraq yerləşəcəklər. "rows" atributunun qiymətlərində biz hər freymin hündürlüyünü yazmışıq (rows="100,\*,150"). Birinci freymin hündürlüyü: 100 piksel, üçüncününkü: 150 pikseldir, ikinci isə digər iki freymin arasında qalan məsafədə yerləşir. Bunu biz \* işarəsi ilə göstərmişik - güman edirəm ki, burda hər şey aydındır.  
  
"frame" teqi brouzerə bildirir ki, freymlərdə hansı sənədlər yüklənəcək. Bizim nümunədə birinci freymdə "logo.html" sənədi (içində loqo olan sənəd) yerləşib, ikinci freym səhifənin məzmununu təşkil edir (content.html), üçüncüsü isə menyudur. Əgər istəyirsinizsə menyu 2-ci sırada olsun (ortada), onda onun yerini "content.html" ilə dəyişin.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset rows="100,\*,150"> <frame src="logo.html"> <frame src="menu.html"> <frame src="content.html"> </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/106.html) |

**Görün bizdə nə alındı. Yerlərini dəyişməyinə dəyişdik, amma gəlin indi "rows" atributuna yeni qiymət verək ki, bizim menyu yenidən 150 piksel hündürlükdə, məzmun isə yerdə qalan hündürlükdə olsun:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset rows="100,150,\*"> <frame src="logo.html"> <frame src="menu.html"> <frame src="content.html"> </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/107.html) |

**Bax bu başqa məsələ:) Doğrudan da toplananların yerini dəyişdikdə nə isə dəyişir...**

**FREYM YARATMAĞI ÖYRƏNİRİK (ARDI)**

**Bu dərsi biz kiçik bir fokusla başlayacayıq. "rows" atributunu "cols" ilə əvəz edin:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,150,\*"> <frame src="logo.html"> <frame src="menu.html"> <frame src="content.html"> </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/108.html) |

**Hə, nümunəyə baxın. Bir balaca hiyləgərlik, başqa heç nə :)   
  
Beləliklə, "cols" atributu bizim brauzerə freymlərin şaquli (sütunlarla) yerləşəcəyini bildirir. Deməli "cols" atributunun qiymətləri (cols="100,150,\*") artıq freymin hündürlüyünü yox, enini təyin edir. Birinci freymin eni 100 piksel, ikincininki 150 pikseldir, üçüncü isə yerdə qalan məsafədə yerləşir.  
  
Yeri gəlmişkən, biz göstərdiyimiz 4 variantdan birini də aldıq:**

****

**Gəlin yekuna yaxınlaşaq. Biz freymləri bir-birinə nisbətən üfiqi və şaquli vəziyyətdə yerləşdirə bilirik. Bunları biz yalnız <frameset> teqi, "cols" və "rows" atributlarının köməyi ilə edirik. Bunun başqa üsulu yoxdur. Bu atributları eyni vaxtda tətbiq etmək olmaz.  
  
"rows" atributnun köməyi ilə biz freymləri üfiqi bölüb yerləşdiririk və hər bir freymə hündürlük ölçüsü veririk, "cols" atributnun köməyi ilə ise biz freymləri şaquli bölüb yerləşdiririk və hər bir freymə en olçüsü veririk.  
  
Lap unutmuşdum, en və hündürlük təkcə piksellərlə verilmir, onlar faizlərlə də verilə bilər.   
  
<frameset cols="10%,15%,75%">   
  
Yadda saxlayın ki, cəm 100%-ə bərabər olmalıdır.**

**FREYM YARATMAĞI ÖYRƏNİRİK (ARDI)**

**Bu dərsdə biz sənədləri növbəti üsullarla yerləşdirməyi öyrənəcəyik.**

**         və ya   **

**Əvvəlki dərslərdə mən demişdim ki, biz <frameset> teqində "rows" və "cols" atributlarını eyni vaxtda işlədə bilmərik. Bəs onda sənədləri şəkildə göstərildiyi kimi necə yerləşdirmək olar? Çox sadəcə, freym sənədimizə daha bir <frameset> teqi daxil etməklə, amma gəlin hər şeyi səliqə ilə edək.   
  
Birinci şəkildən başlayaq:**

****

**Pəncərəni hansı yolla böləcəyik? Üfiqi şəkildə. Yuxarıda bizim "logo.html" sənədimiz yerləşəcək, 2-ci freymi isə biz içində "menu.html" və "content.html" sənədləri yerləşən 2 sütuna böləcəyik.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset rows="100,\*"> <frame src="logo.html"> <???> </frameset> </html>** |

**Quruluş aydındır, bəs 2 sütuna bölünəcək freymi necə işarə edək? Dediyim kimi burda bizə cənab "frameset" kömək edəcək.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset rows="100,\*">  <frame src="logo.html"> <frameset cols="150,\*"> <frame src="menu.html"> <frame src="content.html"> </frameset>  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/109.html) |

**Kim mənə yaxşı diqqət yetirmədi, onlar üçün izah edirəm. Birinci üfiqi freymi biz lazım olduğu kimi - "frame" teqinin köməyi ilə düzəltdik. İkinci freymdə isə yenidən <frameset></frameset> teqi köməyimizə gəlir.   
  
<frameset></frameset> teqinin "cols" atributu ilə biz ikinci freymi 2 sütuna bölürük. (Birinci 150 piksel eninde, ikinci isə yerdə qalan məsafəni tutur.) <frameset></frameset> teqinin içindəki <frame> teqi isə ikinci üfiqi freymə hansı sənədlərin şaquli xətt üzrə yerləşəcəyini müəyyən edir (menu.html və content.html).   
  
İndi 2-ci şəkildəki variantı araşdıraq:**

****

**Burda biz pəncərəni sütunlara böləcəyik. İkinci sütun özündə "content.html" sənədini (məzmun) saxlayır, birinci sütunu isə biz 2 cərgəyə bölürük və onlara "logo.html" və "menu.html" sənədlərini yerləşdiririk.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*">  <frameset rows="100,\*"> <frame src="logo.html"> <frame src="menu.html"> </frameset> <frame src="content.html">  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/110.html) |

**Bax belə. Çətin heç nə yoxdur:)**

**"FRAME" TEQİNİN XASSƏLƏRİ. SCROLLİNG-İ YIĞIRIQ, ÇƏRÇİVƏLƏRİ SİLİRİK**

**Deyək ki, biz sənədimiz üçün**[**sonuncu variantı**](http://ilkaddimlar.com/derslikler/html_tutor/sample/primtoframez6.html)**seçmişik, onda gəlin onu daha normal şəklə gətirək. Əvvəlcə freymdəki "logo.html" sənədinin scrolling-ini silək.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*">  <frameset rows="100,\*"> <frame src="logo.html" scrolling="no"> <frame src="menu.html"> </frameset> <frame src="content.html">  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/111.html)**)** |

**"scrolling" - <frame> teqinin atributudur. O bir neçə qiymət ala bilər: "no" - bu o deməkdir ki, ümumiyyətlə scrolling olmayacaq, "yes" - bu o deməkdir ki, scrolling həmişə görünəcək, "auto" - scrolling ancaq lazım olduqda meydana çıxacaq.  
  
Əslində, scrolling="auto" yazılmaya da bilər, çünki əgər "scrolling" atributu verilməyibsə, onda scrolling lazım gəldikdə görünəcək, lazım olmazsa görünməyəcək.  
  
Siz əgər siçanı hər hansı freymin sərhədinə yaxınlaşdırsanız, görəcəksiniz ki, ona toxunarkən freymin ölçülərini dəyişmək olur. Bəz bunu istifadəçilərinizə necə qadağan edə bilərsiniz? Əlbəttə ki, <frame> teqinə "noresize" atributu təyin etməklə:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*">  <frameset rows="100,\*"> <frame src="logo.html" scrolling="no" noresize> <frame src="menu.html" noresize> </frameset> <frame src="content.html" noresize>  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/111.html)**)** |

**İndi isə gəlin freymlərin arasındakı çərçivələrdən xilas olaq. Bunun üçün biz artıq çoxdan tanıdığımız "border" atributunu tətbiq edirik, beləliklə border="0".**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*" border="0">  <frameset rows="100,\*"> <frame src="logo.html" scrolling="no"> <frame src="menu.html"> </frameset> <frame src="content.html">  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/111.html)**)** |

**Bax belə artıq yaxşıdır :). Amma yenə də istəyirsən ki, nəyi isə qaydaya salasan. Yuxarıdakı şəkli küncə sıxmaq lazımdır:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*" border="0">  <frameset rows="100,\*"> <frame src="logo.html" scrolling="no" marginwidth="0" marginheight="0"> <frame src="menu.html"> </frameset> <frame src="content.html">  </frameset> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/111.html)**)** |

**"marginwidth" və "marginheight" atributları sizə artıq tanışdırlar, onlar freymin sahəsinin enini təyin edirlər. Bizim sənəddə biz bu atributlara sıfır qiyməti verməklə yuxarıdakı şəklin (logo.html) sahələrini yığdıq.  
  
Dayan, dayan. Siz deyəcəksiniz ki, bəs biz bu atributlar haqqında "standart olmayan atributlar" kimi danışmışdıq? Əlbəttə, amma onlar yalnız <body> teqi ilə işlənərkən "standart olmayan" kimi sayıla bilərlər, <frame> teqində onları çəkinmədən işlədə bilərsiniz.  
  
Bax belə:) Qalanı da o biri dərslərdə.**

**BAĞLANTIYA SIXARKƏN SƏNƏD DİGƏR FREYMDƏ TAM PƏNCƏRƏDƏ AÇILIR**

**Bu dərsdə biz freymləri öyrənməyə təzə başlayanlarda qarşıya çıxan problemlərdən danışacayıq.  
  
Gəlin bizim**[**Nümunə**](http://ilkaddimlar.com/samples/12019/112.html)**diqqət yetirək. Ordakı hər hansı bağlantıya basın. Bağlantının açdığı sənəd menyu olan sənədin yerləşdiyi freymin içində açılacaq. Amma bizə lazımdır ki, o, içində məzmun olan freymdə açılsın, menyu isə toxunulmamış qalsın. Bunu necə edək?  
  
Əvvəlcə <frame> teqinin yeni atributu olan "name" (yeri gəlmişkən, mənə elə gəlir ki, biz "name" ilə artıq tanışıq:) ilə tanış olun. "name" atributunu işə salırıq:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Freymlərlə səyahət</title> </head> <frameset cols="100,\*" border="0">  <frameset rows="100,\*"> <frame src="logo.html" scrolling="no" marginwidth="0" marginheight="0"> <frame src="menu.html"> </frameset> <frame src="content.html" name="window-1">  </frameset> </html>** |

**"name" atributu freymə unikal ad verir (bizim nümunədə biz bu atributu özündə məzmunu saxlayan "content.html" sənədinə tətbiq edirik).  
  
Freymin bu adı başqa sənədlərdən (freymlərdən) də ona bağlanmaq üçün istifadə oluna bilər. <a> teqinin "target" (target="freymin\_adı") atributu ilə biz artıq tanışıq.  
  
"target" atributunu bağlantılar olan sənədimizə (menu.html) daxil edək.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Menyu sənədi</title> </head> <body background="cherti3.gif" text="#ffffff" link="#ffffff" alink="#ffffff" vlink="#ffffff"> <center> <a href="content.html">Əsas</a> <a href="tumki.html">Tumki</a> <a href="bumki.html">Bumki</a> <a href="tururumki.html">Tururumki</a> <a href="tra-la-la.html">Tralyalya</a>  </center> </body> </html>** |

**Bizim nümunədə o sənəd (menu.html) belədir. İndi isə hər bağlantı üçün target="window-1" əlavə edək, "window-1" - bu içərisində məzmun sənədi (content.html) yerləşən freymin adıdır.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **<html> <head> <title>Menyu sənədi</title> </head> <body background="cherti3.gif" text="#ffffff" link="#ffffff" alink="#ffffff" vlink="#ffffff"> <center> <a href="content.html" target="window-1">Əsas</a> <a href="tumki.html" target="window-1">Tumki</a> <a href="bumki.html" target="window-1">Bumki</a> <a href="tururumki.html" target="window-1">Tururumki</a> <a href="tra-la-la.html" target="window-1">Tralyalya</a>  </center> </body> </html>**[**Nümunə**](http://ilkaddimlar.com/samples/12019/113.html) |

**İndi artıq bütün bağlantılar bizə lazım olan freymdə açılır, menyu isə heç yerə itmir. Bütün bunları isə biz cəmi iki sənəddə xırda dəyişikliklər etməklə aldıq.  
  
Bəzən elə olur ki, biz istəyirik açılacaq sənəd tam pəncərədə açılsın, bunun üçün "target" atributuna "\_top" qiymətini vermək lazımdır:  
  
<a href="project.html" target="\_top">Mənim balıqıar haqqında layihəm</a>  
  
"Mənim balıqıar haqqında layihəm" səhifəsi bütün qalan kadrları (freymləri) örtərək tam pəncərədə açılacaq.**

### IFRAME NƏDİR

**Budur, artıq biz freymlər haqqında sonuncu dərsə gəlib çatdıq. Əslində biz freymlər haqqında bundan əvvəlkı dərsdə danışıb qurtardıq, bu dərsdə isə biz "IFrame" ("üzən" freym) haqqında danışacağıq.   
  
Beləliklə, "Iframe" nədir və o, adi freymdən nə ilə fərqlənir. IFrame - "üzən" və ya daxili freymdir. Siz yəqin ki, onunla müxtəlif saytlarda rastlaşmısınız. "Iframe" üçün ayrıca freym-sənəd yaratmaq və ona bizim səhifənin freym strukturunu yazmaq lazım deyil. "Iframe" bizə bir HTML sənədi başqa bir HTML sənədin içinə qoymağa imkan verir.  
  
  
   
  
  
Gördüyünüz kimi, verilmiş ölçüdə, içində bizə lazım olan sənəd olan forma yaranır. Bu çox rahatdır, siz yəqin ki, hansısa saytda qeydiyyatdan keçəndə, qaydalarla tanış olarkən "üzən" freymlərlə rastlaşmısınız. Bu onun üçündür ki, siz başqa səhifəyə keçmədən qaydalarla rahat tanış olasınız.  
  
"Üzən" freym bizim HTML sənədimizə <iframe></iframe> teqi vasitəsilə daxil olunur:  
  
<IFRAME src="ancorpri.html" width="300" height="250" scrolling="auto" frameborder="1"></IFRAME>  
  
"src" atributunda biz freymdə göstəriləcək sənədə olan yolu göstəririk.  
  
"height" və "width" atributları daxili freymin enini və hündürlüyünü teyin edir.  
  
"scrolling" atributu və onun qiyməti ilə biz artıq köhnə dərslərdən tanışıq. O bir neçə qiymət ala bilər: "no" - bu o deməkdir ki, ümumiyyətlə scrolling olmayacaq, "yes" - bu o deməkdir ki, scrolling həmişə görünəcək, "auto" - scrolling ancaq lazım olduqda meydana çıxacaq.  
  
"frameborder" atributu daxili freymin sərhədlərinin görünüb-görünməməsini göstərir. Mümkün qiymətləri: 1 (görünsün) və 0 (görünməsin).  
  
İstifadəçi daxili freymin sərhədlərini siçanın köməyi ilə uzada bilməz, ona görə də daxili freymdə "noresize" atributu yoxdur.  
  
"Iframe" səhifənin içindəkilərə (mətnə, şəklə və s.) nəzərən düzlənə bilər.  
  
"Iframe" tətbiq edərkən hansı problemlərlə üzləşə bilərik: O bir çox brauzerlər tərəfindən dəstəklənmir, əsasən köhnə brauzerlər onu dəstəkləyirlər. Ona görə də bu elementi saytınızın səhifəsində işlədərkən əvvəlcə düşünün, çox vacib deyilsə, işlətməyin.  
  
Bununla da freymlər haqqında söhbətimizi bitiririk, amma sizə bunlarla kifayətlənməyib, HTML-in daha geniş imkanları ilə maraqlanmağı, onları öyrənməyi, HTML haqqında biliyinizi artırmağı tövsiyə edirəm.**

**OBYEKT ANLAYIŞI**

Obyekt, mürəkkəb verilən tipidir. Obyektin tərkibinə dəyişən çoxluğu - xassə və bu dəyişənlər üzərində manipulyasiya etmək üçün funksiya çoxluğu - metod daxildir. Xassə verilən saxlayır, metod isə onları emal edir. Bu şəkildə obyektə proqramın digər fraqmentlərindən asılı olmayaraq, ayrıca baxmaq olar.  
  
Obyektə nümunə kimi, HTML sənədə baxaq. O, xassələr çoxluğuna malikdir: internet-ünvan, ölçü, simvol kodlaşdırılması və s. Bəzi xassələrini dəyişə bilərsiniz, bəzilərini isə yox. O, həmçinin metodlar çoxluğuna da malikdir: Web browser pəncərəsində göstərmək, disk üzərində saxlamaq, çap etmək və s. Belə bir analogiya vermək də olar: xassə - obyektin atributu, metod - idarəetmə vasitəsi.  
  
Obyekt digər verilənlərin tiplərindən əlavə olaraq bir də onunla fərqlənir ki, onun istifadəsi üçün uyğun obyekt sinfinin nüsxəsini yaratmaq lazımdır. Bu zaman sinif dəyişən tiplərinə oxşar olaraq obyekt tipinin nümunəsidir, nüsxə isə onunla işləyə biləcəyiniz konkret obyektdir. Nüsxə obyekt "new" operatoru vasitəsilə yaradılır. Bu zaman obyektə istinad dəyişənə mənimsədilir. İstinad (və ya göstərici) yaddaşda bir yerə nişanlanır, hansı ki, yaradılan nüsxə obyekt saxlanılır. Dəyişəndə saxlanılan bu göstərici ilə obyektə müraciət edə bilərik.

var obj;

obj = new SomeObject();

Burada obj dəyişəndir, hansı ki, yaradılmış obyektə istinad edən göstərici mənimsədilib, SomeObject isə, obyekti yaradan sinifdir.  
Obyekti isifadə edib qurtardıqdan sonra, onu silmək olar. Bunun üçün "delete" operatorundan istifadə edilir.

delete obj;

**Qeyd:** "delete" operatoru yalnız Internet Explorer 6.0 və Navigator 4.0 versiyalarından başlayaraq dəstəklənir. Internet Explorer 5 və Navigator 3.0 versiyalarında dəyişənə null qiymətini mənimsədərək, obyekti silə bilərsiniz. Navigatorun çox köhnə versiyaları obyekti silməyi ümumiyyətlə dəstəkləmir.  
  
Yaxşı, biz obyekt yaratdıq və onun xassə və metodlarına müraciət etmək istəyirik. Bunu necə etməliyik? Çox sadə. Aşağıdakı çağırma formatından istifadə edərək, xassələrə müraciət edə bilərsiniz:  
**{Obyektə istinad edən dəyişənin adı}.{Xassənin adı}**  
  
Bu şəkildə xassəyə müraciət etmək dəyişənə müraciət etmək kimidir, fərq yalnız ondan ibarətdir ki, burada dəyişənin - xassənin əvvəlinə obyekti göstərən dəyişənin adını və nöqtə işarəsini qoymaq lazımdır.

obj.SomeProperty = 9;

encoding = HTMLDocument.CodePage;

Metod da oxşar şəkildə çağırılır:  
**{Obyektə istinad edən dəyişənin adı}.{Metodun adı}([{Parametrlər}])**  
  
Yəni demək olar ki, adi funksiya kimi çağırılır, istisna olaraq, obyektin adını (obyektin adı - obyektə istinad edən dəyişənin adı mənasındadır) yazmaq lazımdır.

void HTMLDocument.SaveToDisk('somefile.htm');

Obyekt massivin elementinə mənimsədilə və funksiyaya göndərilə bilər. Bu zaman funksiyanın gövdəsində onun xassələrini dəyişməyə və metodunu çağırmağa icazə verilir.  
Obyekt həmçinin obyekt tipində xassəyə malik ola bilər, yəni xassə daxili obyekt adlandırılan obyektə göstərici saxlaya bilər. Analoji qaydada daxili obyektin xassə və metodlarına müraciətə icazə verilir:

encoding = someObject.internalObject1.property1;

v = someObject.internalObject1.method1;

Koddan göründüyü kimi, əvvəlcə obyektin adı, sonra daxili obyektin adı, daha sonra isə, daxili obyektin xassəsinin və ya metodunun adı göstərilib.  
Yeri gəlmişkən, obyektin özü bu şəkildə yaradılır:

obj = new SomeObject();

O, funksiya çağırılmışına çox oxşayır, elə deyilmi? Bu da öz növbəsində funksiyadır, konstruktor adlanan obyektin xüsusi metodudur. Bu metod digərlərindən onunla fərqlənir ki, bu metod obyekt yaradılarkən çağırılır və onun xassələrinin ilkin qiymətlərini verir. Konstruktorun adı həmişə sinifin (class) adı ilə eyni olur.  
Konstruktor parametr ala bilər.

obj2 = new SomeObject2(param1, param2, x + y, 3);

Konstruktorsuz yaradılan obyektlər də mövcuddur. Onlar inisiallaşdırıcı vasitəsilə yaradılır:

obj3 = new SomeObject3(prop1: param1, prop2: param2, prop3: x + y, prop4: 3);

Yəni bu cür obyekt yaradarkən, biz onun xassələrini qiymətlərlə doldururuq.  
  
**Qeyd:** İnisiallaşdırıcı Internet Explorer və Navigator browserlərində 3.0 versiyalarından başlayaraq dəstəklənir. Navigatorun (JavaScript-i Internet Explorer 3.0 versiyasından başlayaraq dəstəkləyir) daha əvvəlki versiyalarında, siz obyekti yalnız konstruktor vasitəsilə yarada bilərsiniz.  
  
JavaScript-də olan bütün mümkün obyektlər daxili və istifadəçi yaradan olmaqla iki qrupa bölünür. Daxili siniflərə misal olaraq Array, Boolean, Date, Function, Math və s. misal göstərmək olar. Daxili siniflər burada araşdırılmayacaq.  
typeof operatoru obyekt dəyişəni üçün "object" sətirini qaytarır.

### BİR NEÇƏ YENİ OPERATOR

Obyektlərlə işləyərkən tətbiq olunan bir neçə yeni operatora baxmaq zamanı gəldi.  
**in** operatoru verilmiş obyektdə verilmiş xassənin olub-olmadığını yoxlayır və true və ya false qiymətini qaytarır.  
**{xassənin adı} in {obyektin adı}**  
  
Burada xassənin adı sətir şəklində verilir, obyektin adı isə, obyekt tipli dəyişən kimi verilir.  
Bu operatoru məsələn, aşağıdakı kimi istifadə etmək olar:

if ("SomeProperty" in obj) ...

**Qeyd:** in operatorunu yalnız Internet Explorer dəstəkləyir.  
  
**instanceof** operatoru verilmiş obyektin verilmiş sinifin nüsxəsi olub-olmadığını yoxlayır, true və ya false qiymətini qaytarır.  
**{obyekt} instanceof {sinif}**  
  
Burada obyekt obyekt tipli dəyişən şəklində verilir, sinif isə sinifin adı şəklində verilir.

if (obj instanceof SomeObject) ...

**Qeyd:** **instanceof** operatoru Internet Explorer-in 5.5-ci versiyasından başlayaraq dəstəklənir.  
  
**for-in** operatoru obyektin bütün xassələrinə və ya massivin bütün elementlərinə baxmağa və hər biri üçün əməliyyat aparmağa imkan verir. Bu operator xüsusiləşdirilmiş dövr operatorudur.

for({sayğac} in {obyekt və ya massiv})

... dövrün gövdəsi

sayğacı obyektin xassəsinə və ya massivin elementinə müraciət etmək üçün istifadə etmək olar.

for(k in obj)

{

s += k + ‘ ’;

st += obj[k] + ‘ ’;

}

Bu proqram fraqmenti obj obyektinin bütün xassələrinə baxır və xassələrin adlarını boşluq işarəsi ilə ayıraraq, s dəyişəninə yazır, st dəyişəninə isə, xassələrin qiymətlərinin yenə də boşluq işarəsi ayıraraq saxlayır.  
  
**Qeyd:** **for-in** operatoru Internet Explorer-in 5.5-ci versiyasından başlayaraq dəstəklənir.  
  
**with** operatoru, əgər proqramda hər hansı obyektin (qeyd: bir obyektin) xassə və metodları istifadə olunursa, JavaScript ifadələrinin uzunluğunu xeyli qısaltmağa imkan verir. Misala baxaq.

someobject.property1 = 1;

someobject.property2 = 2;

someobject.property3 = 3;

someobject.method1;

Bu, bizim **with** operatoru istifadə olunmayan proqram fraqmentimizdir. Baxın ifadələr necə uzun sətirdir. İndi isə, **with** operatorunu istifadə edək.

with(someobject)

{

property1;

property2;

property3;

method1;

}

Bu proqram fraqmenti isə, olduqca qısadır.

### YARADILMA

Bəzən JavaScript-in tərkibində olan Class-ların (siniflərin) imkanları məsələlərin həlli üçün kifayət etmir. Bu zaman istifadəçi öz class-larını yaradır.  
İstifadəçi class-ını yaratmaq üçün nə etmək lazımdır? O qədər çox nə isə tələb olunmur. Əgər class konstruktora malik deyilsə, onda onun nüsxəsini sadəcə inisiallaşdırıcı istifadə etməklə **new** operatoru ilə yaratmaq lazımdır.

var somePoint;

somePoint = new SomePoint(x: 100, y: 50, color: "black");

Hamısı bu qədər. Bu şəkildə biz bir qrafiki nöqtəni göstərən, koordinat və rəng xassələri olan obyekt yaratdıq. Artıq indi biz onu istifadə edə bilərik.

somePoint.y = somePoint.x / 2 + 20;

Hətta yaradılmış bu obyektə xassə əlavə etmək mümkündür.

somePoint.radius = 0.5;

Lakin biz öz sinifimizi yaratmadıq. Öz sinifimizi həqiqətən yaratmaq üçün konstruktor təyin etmək lazımdır.  
Əvvəlcədən bildiyimiz kimi konstruktor sinif üçün xassə təyin edir və onlara susmaya görə (default) qiymət verir. (Susmaya görə qiymət proqram kodunda sərt olaraq təyin oluna bilər və ya konstruktora parametr kimi verilə bilər.) Həmçinin konstruktor sinfə metod təyin edir (əlavə edir).  
  
Gəlin, qrafiki nöqtə sinfini yaradaq. Bunun üçün aşağıdakı konstruktoru yazaq.

function Point(x, y, color)

{

this.x = x;

this.y = y;

this.color = color;

}

Biz cari obyektə istinad etmək (müraciət etmək) üçün yeni **this** operatorundan istifadə etdik. Bu operatoru yaddan çıxarmayın, o bizə çox lazım olacaq.  
  
İndi biz Point sinfindən obyekt yarada bilərik.

var somePoint;

somePoint = new Point(100, 50, "black");

Gəlin, yenicə yaradılmış Point sinfinə metod əlavə edək. Yadda saxlayın ki, dəyişən funksiyaya göstərici saxlaya bilər. Bu dəyişən sinfin xassəsi də ola bilər.  
Əvvəlcə, funksiya təyin edək. Bu metod koordinatların müəyyən aralıqda olub-olmadığını yoxlayır və buna uyğun olaraq, true və ya false cavabını qaytarır.

function isPointInBounds()

{

if(this.x>0 && this.x<400)

{

return (this.y>0 && this.y<200);

}

else

{

return false;

}

}

Yuxarıda elan etdiyimiz Point sinfini yenidən təyin edək.

function Point(x, y, color)

{

this.x = x;

this.y = y;

this.color = color;

this.isPointInBounds = isPointInBounds;

}

Sonuncu sətirdə biz isPointInBounds xassəsinə isPointInBounds funksiyasına olan göstərici mənimsədirik. İndi biz bu xassəni isPointInBounds funksiyasını çağırmaq üçün istifadə edə bilərik. Faktiki olaraq, bu funksiya obyektin bir hissəsi olur, yəni metodu.

if(somePoint.isPointInBounds())

{

...

}

Əgər istifadəçi xassələrə mənimsətmək üçün lazım olan parametrləri konstruktora verməyibsə, bu zaman, xassə üçün susmaya görə olan qiymətdən istifadə etmək olar. Bunu etmək üçün || (məntiqi "və ya") operatorunun xüsusi formasından istifadə olunur.

function Point(x, y, color)

{

this.x = x || 100;

this.y = y || 50;

this.color = color || “black”;

this.isPointInBounds = isPointInBounds;

}

Əgər || operatorundan sol tərəfdə olan arqument təyin olunubsa, o qaytarılır. Əks halda, || operatorundan sağ tərəfdə olan arqument qaytarılır, bizim halda susmaya görə qiymət. Buradan belə çıxır ki, biz obyekti aşağıdakı şəkildə yarada bilərik:

somePoint = new Point();

İndi bir az əvvələ qayıdaq. İnisiallaşdırıcı ilə yaradılmış obyektə yeni xassə əlavə etdiyimizi yada salın. Biz eyni hərəkəti konstruktor ilə yaradılmış somePoint obyekti üçün də edə bilərik.

somePoint.radius = 0.5;

Amma burada qeyd etmək lazımdır ki, əlavə etdiyimiz xassə yalnız somePoint obyektinə məxsus olur, digər Point sinfindən olan obyektlərdə bu xassə olmur. Bəs bu xassənin Point sinfində olan bütün obyektlərdə olmasını necə təmin etmək olar?

### PROTOTİP İSTİFADƏ ETMƏK

Çıxış var - obyektin prototipini istifadə edin və xassə əlavə edin.  
Prototip - bu bir növ obyektin sinfinə istinaddır. JavaScript-də olan bütün siniflərin (hətta JavaScript-in daxili siniflərinin) prototype adlı xassəsi var və onun prototipini göstərir. Prototipi sinfi genişləndirmək üçün istifadə edirik.  
  
Obyektə xassə əlavə etmək üçün aşağıdakı kodun əvəzinə

somePoint.radius = 0.5;

yeni xassəni onun prototipinə daxil edək (istənilən sinfə məxsus olan prototype xassəsini istifadə edərək):

Point.prototype.radius = 0.5;

İndi isə, Point sinfindən olan bütün obyektlərin radius xassəsi var. Lakin ancaq somePoint obyektinin radius xassəsi 0.5-ə mənimsədilib, digər obyektlərdə isə, bu qiymət təyin olunmayıb.  
Biz həmçinin daxili sinifləri də dəyişə bilərik, xassə və metod əlavə edə bilərik. Məsələn, Math sinfinə lazımi riyazi funksiyalar və konstantlar əlavə edə bilərik.  
  
Lakin prototipi digər məqsədlər üçün də istifadə etmək olar. Məsələn, biz sinfi irsən ala bilərik. Misal üzərində izah edək.  
Tutaq ki, biz Point sinfinə bir neçə yeni xassə əlavə etmək istəyirik. Bunu hansı şəkildə edək? Əlavə olunacaq bütün yeni xassələri həmin sinifdən olan hər hansı obyektə növbə ilə əlavə etmək olar, lakin bu, çox da yaxşı deyil. Bunu başqa cür edək. Yeni SuperPoint nöqtə sinfi yazaq və bu sinifdə köhnə Point sinfinin bütün xassə və metodlarını irsən alaq. SuperPoint sinfi Point sinfinin varisi olacaq və valideyninin (və ya əcdadının) bütün xassə və metodlarını istifadə edə biləcək. Bu şəkildə biz iki nöqtə sinfinə malik olacağıq: sadə və təkmilləşdirilmiş.  
  
Əvvəlcə SuperPoint sinfini və onun bütün yeni xassələrini təyin edək. Və bizim koda sətir-sətir baxaq.

function SuperPoint(x, y, color, radius, edgeColor)

{

Bura qədər hər şey aydındır. Yeni sinif üçün konstruktor funksiyası təyin etdik. Qeyd edək ki, konstruktorun parametr siyahısında həm yeni xassələri elan etdik, həm də irsən valideyn sinifdən alınan xassələri elan etdik.

this.base = Point;

this.base(x, y, color);

Biz SuperPoint sinfinin yeni base xassəsini təyin etdik və ona Point sinfinin konstruktoruna göstərici mənimsətdik. Bundan sonra, valideyn sinfin konstruktorunu tələb olunan parametrləri verərək çağırdıq. Bu çağırmanın nəticəsində valideyn sinifdən irsən alınan bütün xassələr qiymət aldı. Base xassəsinin adı burada bir rol oynamır, yalnız rahatlıq üçün belə seçilib.

this.radius = radius || 0.5;

this.edgeColor = edgeColor || "black";

Burada isə, yeni xassələrə ilkin qiymət veririk. Bu xassələr üçün susmaya görə qiymət təyin olunduğuna diqqət yetirin. Bu susmaya görə təyin olunan qiymətlər, əgər bizim bu sinfi istifadə edən proqramçı təyin olunan qiymətlərdən birini (və ya ümumiyyətlə heç birini) konstruktora verməzsə, istifadə olunur. Susmaya görə qiymət yaxşı stil proqramlaşdırmadır.

}

SuperPoint.prototype = new Point;

Qeyd edək ki, bu sətir konstruktorun gövdəsindən kənarda yerləşir.  
Artıq indi yeni sinifdən obyekt yarada və onun yeni xassələri kimi, irsən alınan xassələrini də istifadə edə bilərik.

var obj;

obj = new SuperPoint(100, 100, “green”, 0.6);

Burada biz konstruktora bütün parametrləri vermədik, lakin bu qəribə deyil, axı biz xassə üçün susmaya görə qiymətə baxmışıq.

obj.radius = Math.SQRT(5);

İndi biz SuperPoint sinfinin yeni xassəsinə müraciət etdik və ona 5-in kvadratı qiymətini mənimsətdik.

a = obj.x + 20;

İndi isə, Point sinfindən irsən alınan xassəyə müraciət etdik.  
Biz valideyn sinfindən irsən alınan istənilən xassə və ya metodu yenidən təyin (override) edə bilərik. Daha bir sadə misala baxaq.

function SuperPuperPoint(x, y, color, radius, edgeColor)

{

this.base = SuperPoint;

this.base(x, y, color, radius, edgeColor);

this.isPointInBounds = isPointInBounds2;

Burada biz isPointInBounds xassəsinə digər isPointInBounds2 funksiyasına göstərici mənimsədərək, isPointInBounds metodunu yenidən təyin etdik.

}

SuperPuperPoint.prototype = new SuperPoint;

Və bizim yeni sinfin valideynini göstərməyi unutmayın.  
Point, SuperPoint və SuperPuperPoint siniflərinin tam kodları aşağıda verilmişdir.

<html>

<head>

<title>

Class Test

</title>

<script language="JavaScript">

// Begin Declaration Point class

function Point(x, y, color)

{

this.x = x || 100;

this.y = y || 50;

this.color = color || “black”;

this.isPointInBounds = isPointInBounds;

}

function isPointInBounds()

{

if(this.x>0 && this.x<400)

{

return (this.y>0 && this.y<200);

}

else

{

return false;

}

}

// End Declaration Point class

// Begin Declaration SuperPoint class

function SuperPoint(x, y, color, radius, edgeColor)

{

this.base = Point;

this.base(x, y, color);

this.radius = radius || 0.5;

this.edgeColor = edgeColor || “black”;

}

SuperPoint.prototype = new Point;

// End Declaration SuperPoint class

// Begin Declaration SuperPuperPoint class

function SuperPuperPoint(x, y, color, radius, edgeColor)

{

this.base = SuperPoint;

this.base(x, y, color, radius, edgeColor);

this.isPointInBounds = isPointInBounds2;

}

SuperPuperPoint.prototype = new SuperPoint;

function isPointInBounds2()

{

if(this.x>0 && this.x<300)

{

return (this.y>0 && this.y<300);

}

else

{

return false;

}

}

// End Declaration SuperPuperPoint class

</script>

</head>

<body>

</body>

</html>

**PHP-NİN YARANMA TARİXİ VƏ İMKANLARI**

PHP dili 1994-cü ildə yaradılıb. Onun yaradıcısı Rasmus Lerdorf, ilk olaraq, bu dilin vasitəsilə sayta daxil olan ziyarətçilərin sayını müəyyən etməyi bacarıb.  
Yaradıcının sözünə görə, dil demək olar ki, çox qısa zamanda yazılıb. Əvvəlcə proqram Perl dilində yazılaraq, sayt əlavəsinin interfeysindən ibarət olub.   
Proqram saytda daxil olan informasiyanın dinamik dəyişməsi funksiyasını yerinə yetirirdi. Ancaq yaradılmış proqram həddindən artıq yavaş işlədiyi üçün Rasmus Lerdorf proqramı yenidən işləmiş və əlavə olaraq dilə C/C++, Fortran, Perl, Java, Visual Basic imkanlarını daxil etmişdir.   
Bundan sonra, serverdən istifadə edənlər bu yeni dil ilə maraqlanmışdır. Beləliklə, PHP dilinin birinci versiyası yayılmağa başlamışdır. Artıq 1995-ci ildə bu dil rəsmi olaraq yayılmağa başlamış və PHP (Personal Home Page) adlandırılmışdır.   
Bununla belə, o vaxt bu dilin imkanları həddindən artıq aşağı olmuşdur. 1995–ci ilin ortalarında PHP-nin, inkşaf etdirilərək, ikinci versiyası yaradıldı. Bu versiyaya əlavə olaraq verilənlər bazası ilə işləmək imkanları daxil edildi. Beləliklə, PHP daim inkşaf etdirilərək 3-cü, 4-cü və hal-hazırda 5-ci versiyası yaradıldı.   
Bu gün PHP dili güclü imkanlara malik HTML səhifəsində xüsusi kodları emal edən bir proqramlaşdırma dilidir. Məhz bu dilin köməyi ilə saytları dinamik etmək olur. Bu dilin köməyi ilə yaradılmış fayllar serverdə yerləşir və orada emal olunur. İstifadəçi PHP kodlu səhifələrə müraciət edərsə, informasiya emal olunaraq istifadəçinin brauzerində görünür.   
Yuxarıda qeyd etdiyimiz kimi, PHP dilinin böyük imkanları ilə yanaşı, bir çox çatışmayan cəhətləri də mövcuddur. Ən böyük çatışmayan cəhət yaradılmış proqramın yavaş işləməsidir. C dili ilə müqayisədə bu dilin surəti dəfələrlə aşağıdır. Ancaq HTML səhifəsində bu dildən script-lər şəklində istifadə olduqda bu problem hiss olunmur.   
Bundan başqa, artıq PHP–nin 4.1 versiyasında yaradılmış kompilyator surətin kifayət qədər artırılmasına imkan verir. Ona görə bu versiyadan sonra PHP dili kompilyasiyası olan dillər qrupuna daxil edildi. Bununla yanaşı, dildə qismən interpretasiya etmək imkanı da saxlanılıb.

### PHP-NİN İMKANLARI

- Verilənlər bazası ilə işləyən Web səhifələrin yaradılması üçün PHP ən əlverişli proqramlaşdırma dilidir. PHP Oracle, Adabas D, Sybase, FilePro, MySQL, Informix, Solid, dBase, ODBC və s.  
- Proqramlara girişi təşkil etmək üçün xüsusi funksiyanın yaradılması və login/parolun qoyulması imkanları;  
- PHP nəinki HTML-in kodunu, mətnini, eyni zamanda, müxtəlif formatlı (GIF, JPEG və ya PNG) şəkilləri emal edə bilir;  
- PHP serverdən faylların daşınmasını təşkil edir. Bundan başqa, PHP yüklənən fayllara tam nəzarət edir və onun ölçülərinə məhdudiyyət qoya bilir, eyni zamanda, həmin faylları tam idarə edir;  
- İstifadəçinin kompüterində brauzer vasitəsilə baxılmış fayllar haqqında (4kb) informasiya saxlayır. Belə informasiya "Cookie" adlanır. Cookie HTTP-nin başlığıdır. İstənilən göndərilən Cookie avtomatik olaraq PHP dəyişəninə çevrilir;  
- İstənilən mətn sətirləri üzərində hər cür manipulyasiya aparmağa imkan verir.  
- Lokal və ya Qlobal səviyyədə səhvləri müəyyən edə bilir və bu səhvlərin istifadəçiyə göstərə və ya göstərməyə bilər. Eyni zamanda, bu səhvləri elektron poçt vasitəsilə istənilən ünvana göndərə bilər.  
- Elektron ünvanları idarə edir və istənilən məktubu istənilən ünvanlara göndərmək imkanı vardır.  
- Sonda onu qeyd etmək olar ki, tam PHP vasitəsilə yaradılmış proqramlar yükləmə zamanı heç bir gecikmələrə yol vermir və normal fəaliyyət göstərir.

### PHP PROQRAMLAŞDIRMA

PHP dilində proqramlaşdırma 2 üsul ilə yerinə yetirilə bilər: Web server əlavəsi kimi və ya Konsol proqramı kimi.  
Biz əsasən birinci üsul üzrə proqramlara baxacağıq. Yəni brauzerin serverə müraciət edərək, yerinə yetirilən proqramın icrasına baxaq.  
Beləliklə, brauzer .php tipli faylı serverdən yükləyir və nəticəni HTML kodu şəklində verir.  
PHP kodunu HTML səhifəsinə daxil etmək üçün teqlərdən istifadə olunur və 4 formada, aşağıdakı şəkildə yazıla bilər:  
  
**1. XML stili:**

<?php

...

?>

**2. SGML stili:**

<?

...

?>

**3. HTML stili:**

<script language="php">

...

</script>

**4. ASP stili:**

<%

...

%>

PHP dilində kodları HTML səhifəsində yazmaq üçün kodları aşağıdakı şəkildə daxil etmək lazımdır:

<?php

.....

?>

Qeyd etmək lazımdır ki, bu yazılışda "php" sözünü yazmamaq da olar.  
Mətnləri və ya dəyişənlərin qiymətini ekrana vermək üçün "echo" əmrindən istifadə olunur. Burada əmrin böyük və ya kiçik hərflərlə yazılmasının heç bir fərqi yoxdur.  
Nümunə:

<?php

echo "Salam, xoş gəlmisiniz.";

?>

PHP dilində dəyişənləri istənilən simvolla və ya simvol birləşmələri ilə işarə etmək olar. Ancaq birinci simvol rəqəm və ya altdan xətt işarəsi ola bilməz.   
Bu zaman dəyişən qarşısında $ işarəsi qoymaq lazımdır. Nəzərə almaq lazımdır ki, C dilində olduğu kimi, dəyişənin böyük və ya kiçik hərflərlə işarə edilməsində fərq vardır.

<?php

$a=5;

$b=6;

$y=$a+$b;

echo $y;

?>

### ŞƏRHİN VERİLMƏSİ

PHP dilində şərhlərin verilməsinin bir çox üsulu vardır. Ən sadə üsul ikili (//) sləşdir.  
Bu işarədən sonra PHP kompilyatoru bütün sözləri əmr kimi qəbul etmir və yerinə yetirmir.  
Bununla yanaşı çoxsətirli şərhdən də (/\*...\*/) istifadə etmək olar. Bir sətirlik şərh vermək üçün (#) işarəsi tətbiq olunur.

<?php

echo("<p>Hello</p>"); // Şərh

echo("<p>Hello</p>"); # Şərh

/\*

Şərh

\*/

?>

### SABİTLƏR

PHP dilində sabit ədədlər define() funksiyası vasitəsilə elan edilir.

<?php

define(CONSTANT, value)

?>

Birinci sabitin adı, ikinci isə onun qiymətidir. Sabitlərdən istifadə etdikdə onların adına əsalanılır:

<?php

define(CONSTANT1,15);

define(CONSTANT2,"\x20");

define(CONSTANT3,"Hello");

echo(CONSTANT1);

echo(CONSTANT2);

echo(CONSTANT3);

?>

### PHP DİLİNDƏ VERİLƏNLƏRİN TİPİ. TİPLƏRİN DƏYİŞDİRİLMƏSİ

Yuxarıda qeyd edildiyi kimi, PHP dilində tipləri elan etmək mümkündür. Bu tipləri proqram boyu sətirlər və ya rəqəmlər üçün istifadə etmək olar. Bir çox tip toplusu mövcuddur:  
integer; string; boolean; double; array; object;  
  
**gettype()** funksiyası dəyişilmiş tipi əvvəlki vəziyyətə qaytarır.

<?php

$var = "5";

$var1 = "5";

echo(gettype($var));

echo "<br />";

echo(gettype($var1));

?>

Birinci halda PHP string tipini, iknici halda integer tipini qaytarır.  
  
**settype()** funksiyası tipi müəyyən edir:

<?php

$var = "5";

echo(gettype($var));

settype($var,integer);

echo "<br />";

echo(gettype($var));

?>

### OPERATORLAR

**Riyazi operatorlar**

|  |  |
| --- | --- |
| + | Toplama |
| - | Çıxma |
| \* | Vurma |
| / | Bölmə |
| % | Qalığın modul qiyməti (nümunə: 5 % 2=1) |

**Məntiq operatorları**

|  |  |
| --- | --- |
| & | (and) |
| | | (or) |
| ^ | və ya (xor) |
| ~ | (not) |
| << | Qüvvətə yüksəltmə |
| >> | Qüvvətə azaltma |

<?php

echo(4<<2); // bərabər 16

echo"<br />";

echo(5>>1); // bərabər 2

echo"<br />";

echo(6&5); // bərabər 4

echo"<br />";

echo(6|5); // bərabər 7

echo"<br />";

echo(6^5); // bərabər 3

?>

**print()** (çapa vermək) funksiya sayılmır (dilin konstruksiyasıdır) və ona görə də, yanında mötərizələr qeyd olunur. Bu komandadan sonra müxtəlif formalara əmrlər vermək olar.  
Ümumi qeyd: Bütün funksiyaların sətri bitdikdən sonra ; (nöqtəli vergül) qoyulmalıdır. Digər halda ondan sonra gələn yeni funksiyanız işləməyəcək və nəticədə error (səhv) verəcək.  
Burda diqqətli olun. Bu funksiya echo ilə oxşar olsa da, onunla eyni formada yazılmır. Ümumilikdə aşağıda göstərdiklərimi siz də yoxlayın.

<?php

print("Hello World"); // Hello World

$z="Nazim";

print $z; // Nazim

print "\" Rüstəmov Nazim\".";

// eynilə echo-dakı kimi, dırnaqlar slash işarəsi ilə görünür " Rüstəmov Nazim".

$i="Rüstəmov";

print "Nazim $i"; // Nazim Rüstəmov

?>

### SADƏ HESAB ƏMƏLİYYATLARI

Digər proqramlaşdırma dillərində olduğu kimi, burada da hesab əməlləri sadə məntiqlə yerinə yetirilir.  
PHP bütün riyazi funksiyaları, mötərizələri, məntiqi funksiyaları, bir vahid artmağa, azalmağa və başqaları.  
  
Məsələn:  
**$x++;** - **x** dəyişəninin bir vahid artması;  
**$x=$y=4;** - **x** və **y** dəyişənlərinin hər ikisi 4-ə bərabərdir;  
**$b=$a=5;**  
**$c=$a++;**  
  
Burada **c** dəyişəni **a** dəyişəninə bərabər olur və bir vahid artır.  
$x=2\*($a++);  
  
İndi ardıcıl 5 ədədin ekrana yazılması proqramına baxaq:

<?php

$i = 1;

echo $i;

$i++;

echo "".$i;

$i++;

echo "".$i;

$i++;

echo "".$i;

$i++;

echo "".$i;

?>

### GET VƏ POST METODLARI

Bu metod vasitəsilə dəyişənlərin qiymətlərini daxil etmək olar. Bu zaman lokal kompüterdən serverə müraciət olunur və dəyişənlərin qiymətini serverə çağırır. Bu metodun yazılış qaydası aşağıdakı kimidir:

<form action="hesabla2.php" method="get">

Daxil et x: (bura istənilən sözü yazmaq olar) <input type="text" name="x" /><br />

Daxil et y: <input type="text" name="y" /><br />

<input type="submit" value="Hesabla" />

</form>

Burada **x** və **y** dəyişənlərinin qiymətini daxil edib "Hesabla" düyməsini basan kimi "hesabla2.php" faylı yüklənir.

$x=$\_GET['x'];

$y=$\_GET['y'];

Bu zaman GET metodu vasitəsilə **x** və **y** qiymətləri oxunaraq serverin yaddaşına yüklənir.  
  
Misal:  
**Birinci fayl:**

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">

<title>Hesabla1</title>

</head>

<body>

<form action="hesabla2.php" method="get">

Daxil et x: (bura istənilən sözü yazmaq olar) <input type="text" name="x" /><br />

Daxil et y: <input type="text" name="y" /><br />

<input type="submit" value="Hesabla" />

</form>

</body>

</html>

**İkinci fayl:**

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">

<title>Hesabla2</title>

</head>

<body>

<?php

$x=$\_GET['x'];

$y=$\_GET['y'];

$result=$x\*$y;

echo "y=$result<br>";

$result=sqrt($x);

echo $result

?>

</body>

</html>

### PRİNTF() И SPRİNTF() FUNKSİYALARI

Qeyd etmək lazımdır ki, hər iki funksiya nəticələrin verilməsi formatını müəyyən etmək məqsədi üçündür. **printf()** funksiyası nəticə selinin brauzerdə verilməsi, **sprintf()** funksiyası isə sətirin veriməsi üçün uyğun formatın seçilməsi üçündür. Yazılış qaydası belədir:

int printf(string format [, mixed args])

string sprintf(string format [, mixed args])

Ümumiyyətlə, bu funksiyaların yazılışı belə olur:

<?php

printf("Hello!"); // "Hello!" yazılır.

sprintf("Hello!"); // Heç nə verilmir.

$str = sprintf("Hello!"); // Həmin sətri dəyişənə mənimsədir.

printf($str); // Həmin söz ekrana verilir.

?>

**Request metodu**  
Bu metod da GET və ya POST metodu kimi dəyişənləri serverə göndərmək üçündür.  
  
**Şərti keçid operatoru (if)**  
Bir çox hallarda qoyulmuş şərtdən asılı olaraq, müxtəlif kodları yerinə yetirmək tələb olunur. Bu halda **if** əmrindən istifadə edilir. Yazılış qaydası aşağıdakı kimidir:

if (şərt)

ifadə\_1;

else

ifadə\_2;

<?php

$i = 10;

$j = 5\*2;

if ($i == $j)

echo '$i və $j bərabərdir';

else

echo '$i və $j bərabər deyil';

?>

Göründüyü kimi, **$i** və **$j** arasındakı şərt == (iki bərabərlik) işarəsi vasitəsilə yerinə yetirilir. Beləliklə, şərtlərin yoxlanması aşağıdakı işarələr vasitəsilə göstərilir:  
Burada **if/else** operatoru şərti operator kimi istifadə olunur. Ümumilikdə bu bele görünür:  
!= - bərabər deyil;  
< - kiçik;  
> - böyük;  
<= - kiçik bərabər;  
>= - böyük bərabər.

<?php

$i = 10;

$j = 5\*2;

if ($i == $j)

echo '$i və $j dəyişənlərinin eyni qiyməti var';

?>

<?php

$i = 10;

$j = 11;

if ($i > $j)

$diff = $j - $i;

echo '$j $i -dən böyükdür; $j və $i arasındakı fərq: '. $diff; // Səhvdir!

?>

Bir çox hallarda qoyulmuş şərtə əsasən, bir neçə əmrlər ardıcıllığı yerinə yetirmək tələb olunur. Bu zaman həmin əmrlər fiqurlu mötərizə daxilində yazılır:

<?php

$i = 10;

$j = 11;

if ($i > $j) {

$diff = $j - $i;

echo '$j $i -dən böyükdür; $j və $i arasındakı fərq: '. $diff;

}

?>

Qeyd etmək lazımdır ki, hətta təkcə bir əmri yerinə yetirmək lazımdırsa, onda da bir sətri fiqurlu mötərizə daxilində yazmaq məsləhətdir. Bu zaman təsadüfi səhvlərə yol vermək şansı azalır.  
Bundan başqa, şərt daxilində yeni şərt vermək olar və bu zaman yazılış qaydası bu cür olur: if... else if... else.

<?php

$i = 10;

$j = 11;

if ($i > $j) {

echo '$i $j -dən böyükdür';

} else if ($i < $j) {

echo '$i $j -dən kiçikdir';

} else { // Bu halda bərabərlikdən başqa bir şey qalmır :)

echo '$i və $j bərabərdirlər';

}

?>

Dəyişənlərin qiymətini serverə göndərərkən ilkin olaraq, həmin dəyişənin elan olunmamasına görə ekrana bu haqda sorğu çıxacaq. Belə vəziyyəti aradan götürmək üçün şərti keçid əmrindən istifadə olunur.

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">

<title>Untitled Document</title>

</head>

<body>

<form action="vurmaq.php" method="post">

<input type="text" name="a" size="20" maxlength="8" /><br />

<input type="text" name="b" size="20" maxlength="8" /><br />

<input type="submit" value="Vurmaq"></form>

<?php

if (isset($\_POST["a"])) {$a=$\_POST["a"]; }

else {$a=0;};

if (isset($\_POST["b"])) {$b=$\_POST["b"];

} else {$b=0;};

echo "<br> hasil=".$a\*$b;

?>

</body>

</html>

Bu yazılış qaydası hər üç - GET, POST və REQUEST metodları üçün doğrudur.

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">

<title>Mühazirə 3</title>

</head>

<body>

<form method="post" action="muhaz3.php">

<input name="i" type="text" size="30" />

<input name="j" type="text" size="20" />

<input type="image" src="../photos/search.jpg" border="0" />

</form>

<?php

if (isset($\_POST["i"])) {$i=$\_POST["i"];}

else {$i=0;};

if (isset($\_POST["j"])) {$j=$\_POST["j"];}

else {$j=0;};

$i=$\_POST["i"];

$j=$\_POST["j"];

if ($i > $j)

{

$diff=$j+$i;

echo "Cəm".$diff;

}

else {

$diff=$j-$i;

echo "Fərq".$diff;

}

?>

</body>

</html>

### SWİTCH

Bir çox hallarda eyni dəyişənin qiymətindən asılı olaraq, proqramın icrasını şaxələndirmək lazım gəlir. Bu zaman **switch** əmrindən istifadə olunur.  
Məsələn, aşağıdakı əmrlər toplusunun əvəzinə

if ($i==1) {

ifadə 1

} else if ($i==2) {

ifadə 2

} else if ($i==3) {

ifadə 3

}

Daha əhəmiyyətli yazılış qaydasından istifadə etmək olar:

<form action="switch.php" method="post">

<input type="text" name="i" />

<input type ="submit" value="Ok" />

</form>

<?php

if (isset($\_POST["i"])) {$i=$\_POST["i"];}

else {$i=1;};

switch ($i) {

case 1:

echo 'Bir';

break;

case 2:

echo 'İki';

break;

case 3:

echo 'Üç';

break;

default:

echo '$i > 3';

};

?>

**switch** əmrindən sonra yazılmış dəyişənin qiymətindən asılı olaraq, ekranda **case** əmrindən sonra göstərilmiş qiymətə uyğun sətir yerinə yetirilir. Burada yazılan ifadəni fiqurlu mötərizənin içərisinə almaq lazım deyil. Burada proqramın icrası **break** əmri vasitəsilə yekunlaşır.  
**default** əmri qalan bütün qiymətlərdə doğruluğunu göstərir. Bu əmr sonda olduğu üçün **break** əmrinə ehtiyac yoxdur. Bu əmrə ehtiyac yoxdursa, onda yazmamaq olar.

<?php

$i = 1;

switch ($i) {

case 0: // "break" qəsdən yazılmayıb!

case 1:

echo 'Sıfır və ya bir';

break;

case 2:

echo 'İki';

break;

case 3:

echo 'Üç';

break;

}

?>

#### **Dövrlər**

İstənilən proqramlaşdırma dilində bu və digər şəkildə dövrlərdən istifadə olunur. Bu baxımdan PHP dilində 3 cür dövr vardır:  
  
**while**  
Bu əmrin yazılış qaydası aşağıdakı kimidir:

<?php

$i = 1;

while($i < 10) {

echo $i . "<br>\n";

$i++;

}

?>

Dövr **while** əmrindən sonra gələn şərti yoxlayır və ondan asılı olaraq, növbəti ifadə və ya fiqurlu mötərizənin içərisindəki ifadələr yerinə yetirilir.  
  
**do..while**

<?php

$i = 1;

do {

echo $i . "<br>\n";

} while ($i++ < 10);

?>

### POST METODU

Bu dərsimizdə sizinlə birlikdə PHP-də **post** funksiyasını öyrənəcəyik. PHP-də **post** funksiyası müəyyən məlumatın daşınması üçün istifadə olunur. Bunun üçün HTML kod ilə qutucuq yaradırıq:

<?php

echo "<input name=\"php\">";

?>

İndi isə, **post** metodunu daha dəqiq öyrənək. Kodlara baxaq:

<?php

// **post** əmri gəldikdə görsənməsi üçün

if($\_POST) {

echo "$\_POST['php']";

}

elseif"<form action=\"hansı səhifəyə yönlənməsi üçün biz bu səhifədə iş gördüyümüz üçün buranı boş saxlayın\" method=\"POST\">"; {

echo "<input name=\"php\">"; // Qutucuq ("php" sözünü istədiyiniz sözlə əvəz edə bilərsiniz.)

echo "<input type=\"submit\" value=\"Göndər\">";

}

?>

Gördüyünuz kimi, əsas tərəfdə kodları yazdıq, ikinci tərəfdə isə, echo $\_POST['postun adı']; ilə ekrana çıxardıq.

### WHİLE FUNKSİYASI

**while** funksiyası funksiyanın təkrarlanması üçün istifadə olunur. Məsələn, bir dəyişənin hər təkrarda artması funksiyasını yazaq.  
Kodlar:

<?php

$a=0;

while($a<100) {

$a++; // Dəyərin artması üçün

echo "$a<br/>";

?>

Bu funksiya əsasən, məntiqi proqramların yazılmasında istifadə olunur. Məsələn, sıfır və yüz arasında beşə bölünən ədədləri ekrana çıxaraq:

<?php

$a=0;

while($a<100)

$a=$a+5;

echo "$a<br/>";

?>

### FOREACH

**foreach** funksiyası PHP-də gələn məlumatları seçdirmək üçün istifadə olunur. Məsələn, bir array var və içində 100-dən artıq məlumat var. Əlbəttə, onu tək-tək seçib yazmaq çətin olur, bu vaxt **foreach** işə yarayır.  
Koda baxaq:

<?php

$ar=array("facebook.com","ilkaddimlar.com","mobiz.az","google.az");

echo "Dəyərin içindəki məlumatlar:<br/>";

$a=0;

foreach($ar as $murik=>$site) {

$a++;

echo "$a)$site<br/>";

?>

### PHP-Nİ YÜKLƏMƏ YOLU

PHP dinamik dil sayılır, yəni belə proqramlar məlumatların sorğu şəklində müəyyən bir serverə göndərilib və oradan sorğuya uyğun cavabların alınması sistemi ilə işləyir. Bu sistemin işləməsi üçün bizə baza məlumatlarına girə bilən serverlə bağlı proqramlar lazımdır. Apache, MYSQL kimi. Mən Apache-ni uyğun bilirəm (daha ətraflı izah olunacaq).  
  
Bizə lazım olacaq ləvazimatlar (proqramlar):   
**PHP 5.1.2** [www.php.net/downloads.php](http://www.php.net/downloads.php)  
**Apache 1.3.34** [www.php.net/downloads.php](http://www.php.net/downloads.php)  
  
PHP 5.1.2 arxivini (\*.rar arxivi) C:\php direktoriyasına boşaltdıqdan sonra C:\php\php5ts.dll faylını C:\WINDOWS\system32 direktoriyasına köçürürük (copy). C:\php qovluğunda php.ini-dist faylını tapacaqsınız. Bu faylı C:\WINDOWS direktoriyasına köçürür və adını dəyişdiririk (php.ini kimi). Php.ini faylını notepad vasitəsi ilə açırsınız və Ctrl+F düymələrini birlikdə basaraq, axtarışa "globals", "magic\_quotes" (ayrı-ayrılıqda) yazın və aşağıdakı formada olmasından əmin olun (əks halda, böyük ehtimalla işləməyəcək). Bunun səbəbini irəlidə izah edəcəyəm.   
registry\_globals= on  
magic\_quotes\_gpc= off  
  
Sonra, 2-ci əsas məsələ, Apache 1.3.34 yükləməyə çalışaq.  
İki dəfə üzərinə basın və "Next" düyməsi ilə irəliləyin, sonra Apache License qəbul edərək (I accept) işarə edin, "Next" basıb irəliləyin və belə bir ekran çıxanda şəkildəki kimi doldurun. (localhost, öz e-poçtunu yaz )



Yüklədikdən sonra C:\Program Files\Apache Group\Apache\conf\httpd daxil olub, orada Ctrl+F vasitəsi ilə "LoadModule" yazıb axtarışa verin və aşağıda yazdığım formada yazı tapsanız, deməli, doğru yoldasınız. Mənim qırmızı ilə işarələdiyim yazını (LoadModule php5\_module C:/php/php5apache.dll) eyni ilə LoadModule-ların sonunda yaz. Səhvə yol verməyin və aşağıdakı ilə eyni olmasını yoxla.  
  
#LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so  
#LoadModule mime\_magic\_module modules/mod\_mime\_magic.so  
#LoadModule status\_module modules/mod\_status.so  
#LoadModule info\_module modules/mod\_info.so  
#LoadModule speling\_module modules/mod\_speling.so  
#LoadModule rewrite\_module modules/mod\_rewrite.so  
#LoadModule anon\_auth\_module modules/mod\_auth\_anon.so  
#LoadModule dbm\_auth\_module modules/mod\_auth\_dbm.so  
#LoadModule digest\_auth\_module modules/mod\_auth\_digest.so  
#LoadModule digest\_module modules/mod\_digest.so  
#LoadModule proxy\_module modules/mod\_proxy.so  
#LoadModule cern\_meta\_module modules/mod\_cern\_meta.so  
#LoadModule expires\_module modules/mod\_expires.so  
#LoadModule headers\_module modules/mod\_headers.so  
#LoadModule usertrack\_module modules/mod\_usertrack.so  
#LoadModule unique\_id\_module modules/mod\_unique\_id.so  
  
LoadModule php5\_module C:/php/php5apache.dll  
  
#  
# Reconstruction of the complete module list from all available modules  
  
2-ci addım. Yenə axtarış verin və bu dəfə "AddModule" axtarın. Eynilə yuxarıda izah elədiyim kimi, AddModule-ların axırında AddModule mod\_php5.c yazın. Yaddaşa verin, faylı bağlayıb çıxın.  
  
#AddModule mod\_vhost\_alias.c  
AddModule mod\_env.c  
AddModule mod\_log\_config.c  
#AddModule mod\_mime\_magic.c  
AddModule mod\_mime.c  
AddModule mod\_negotiation.c  
#AddModule mod\_status.c  
#AddModule mod\_info.c  
AddModule mod\_include.c  
AddModule mod\_autoindex.c  
#AddModule mod\_unique\_id.c  
AddModule mod\_so.c  
AddModule mod\_setenvif.c  
AddModule mod\_php5.c  
  
#  
# ExtendedStatus controls whether Apache will generate "full" status  
  
Sonra Start-dan "Run" düyməsinə basın və şəkildəki kimi yazıb "OK" düyməsinə basın. (net stop apache)



Sonra, bir neçə saniyədən sonra aşağıdakı kimi yazıb OK düyməsinə basın.(net start apache)

