

Author: L.Erdenebayar

Хакер.MN

PHP & HTML анхлан
суралцагчдад зориулав.

Primary Edition

E-mail:
Sosa247@yahoo.com



100100100101101111011101100

Оршлын оронд

Энэх?? мэдээллийг анхнаас нь олж суралцахад г?н х?ндтэй туслалцаа ??лсэн, итгэлийн хайрыг ?гс?н аав, ээж, 2 ах, д?? м?н Ерт?нийн эзэнд г?нээ талархаж байгаагаа энэ ялдамд дурдмаар байна. ? д?р б?рийн минь муу б?хний эсрэг ерт?нийн эзний амь х?н т?р?лхт?нд зориулагдсан гэдгийг та бид сайн мэдэх билээ. Удахг?й би болон миний хэдэн анд найз нарын х?д?лм?р шингэсэн ном гарах гэж байна. Гагцх?? олны х?ч оломг?й далай гэдэгчлэн шинэ д?р т?рх??р энэ зууныг элээх санаатай байна. Алдарт GENERAL MOTORS компаны захирал хэлэхдээ “Бид 40 насанд ажил х?д?лм?рийг жигдр??лж болно. Учир нь 40 насанд амьдрал эхэлдэг юм” гэж номдоо бичсэн бий. Ухаан бодлоос холдож салчихаарг?й л юм бол алив з?йл ч бидэнд тээг т?в?гтэй санагдахг?й байсан бизээ. Хэрэвзээ эл ерт?нийн ужиг учирт толгой балбан зовох, шаналах, бодлогоширохг?й л байвал мунхаг х?ний н?дийг нээж ?г?хг?й гэж ?? Ямартаа ч миний эл санаа бодол нь жинхэнэ биш, худал хуурмаг, залхаалт цээрл??лэлт, ш?дний ?вчин гэж та бодож байж магадг?й. Гагц эл ?зэл санаа минь цорын ганц, энэ ерт?нийн тоосыг х?д?лг?ж чадахг?й юм бол... (?ргэлжлэл бий) Та санал сэтгэгдлээ <http://www.xakep.mn> –д бичиж ?лдээгээрэй. Юу вэ гэвэл Миний сая ??лсэн з?йл “ЭССЭ”.

Ламжавын Эрдэнэбаяр
Sosa247@yahoo.com

Вэб гэж юу вэ? HTML үндэс

Вэб гэж юу вэ?

Вэб бол Интернетийн сүлжээнд холбогдсон компьютер дээр байрлах нэгэн т?рлийн файл, файлууд юм. Товчхон б?г??д тодорхой хариулт. Интернет гэдэг нь олон компьютерийн хоорондын сйлжээ гэж ойлгож болох юм. Эдгээр нь хоорондоо ихэвчлэн HTTP гэсэн протокол ашиглан ашиглан хоорондоо холбогдож мэдээллээ солилцдог. Мэдээлэл гэдэг маань бидний мэдэх Вэб хуудас юм. Вэб хуудасыг байрлуулах компютерийг Вэб сервер гэх ба харин тэр мэдээллийг уншиж х?рвүүлэн харуулж байгаа компьютерийг Client гэдэг.

Client үүд нь вэб харуулагч, х?рвүүлэгчүүдийг ашиглан вэб хуудсыг харуулдаг. Х?рвүүлэгчийн жишээ гэвэл бидний сайн мэдэх Internet Explorer, Netscape Navigator, Opera ... гэх мэт маш олон байдаг. Түгээмэл хэрэглэгддэг нь эхний хоёр юм.

Вэб хуудас болгон ??рийн хаяг буюу Domain нэртэй байна. эдгээр хаягыг ашиглан хүмүүс хэрэгцээт хаягрууаа хандан мэдээлэл авах боломжтой байдаг. Вэбийн хаяг нь цорын ганц байдаг б?г??д ижил

хаягтай байх боломжгүй байдаг.
ж.н : <http://www.sr.mn>
харин дэд домайн нэртэй байж болдог.
ж.н : <http://theone.sr.mn> г.м

HTML кодчлол гэж юу вэ?

Вэб харуулагч, уншигч програмууд нь тогтсон зааварын дагуу мэдээллийг уншиж харуулдаг. Тэрхүү зааврыг HTML гэгдэх кодчлолоор бичдэг. нэг үгээр хэлбэл Вэб хуудас маань ??р?? кодчлол юм. Кодоор ямар үйлдэл хийх, юуг яаж харуулах вэ гэх мэт зааврыг вэб харуулагчид ?гд?г. Вэб харуулагч нь тэрхүү зааврын дагуу вэбийг харуулдаг. Тэрхүү зааврыг World wide Web Consortium буюу дэлхий нийтийн сүлжээний холбоо W3C гэдэг байгууллагаас тогтоон гаргадаг байна.

HTML гэдэг нь Hyper Text Markup Language гэсэн үгийн товлол юм. HTML кодчлолыг энгийн Text Editor текст боловсруулагч програмуудыг ашиглан бичиж болох б?г??д одоо вэб байгуулах ?рг?н боломж бүхий програмууд ихээр гарч ирж байгаа болхоор вэб бүтээхэд маш хялбар болгож ?гч байна. Уг програмууд байгаагүй вед з?вх?н HTML хэлийг судлаж мэдсэн хүмүүс л вэб програмчлаж байсан бол, одоо байдал эрс дээшилж энгийн компьютерийн хэрэглээний програмууд дээр багахан мэдлэгтэй хүмүүс вэб байгуулах ?рг?н боломжтой болсон. Энэ програмуудын гол т?л??л?гч бол Macromedia Dreamweaver юм. Одоо энэ программын сүүлийн хувилбар нь Macromedia Dreamweaver 2004 гэж гарсан. Энэхүү программыг та ??рийн сайтаас нь <http://www.macromedia.com> гэсэн хаягаар орж инэ т?лб?ргийгээр татан авч ашиглах боломжтой.

HTML кодчлолоор үүссэн файлууд нь .html .htm гэсэн ?рг?тг?лтэйг??р хадгалагддаг. HTML нь <> гэсэн хашилтан дотор командуудыг агуулан бичигддэг. ихэвчлэн энэ хосоороо бичигдэнэ. Эхлэл т?гсг?л гэдгийг зааж ?гд?г.

ж.нь

 text baina

энэ юу гэсэн үг вэ гэхээр энэ хооронд бичигдсэн текст Bold буюу ?рг?н тэкс гэсэн параграфаар хэвжүүлэгдэнэ.

HTML кодчлол <html> ээр эхлэн </html> ээр т?гс?н?.

```
<html>
....
...
...
</html>
```

HTML кодчлолоор яаж вэб бүтээх тухай дараа дараачийн хичээлээр тодорхой үзэх болно.

HTML дээр зураг оруулах, Солигддог зураг хийх

Мэдээж зураг оролцоогүй бүтсэн вэб гэж ховор байхаа. HTML кодчлолд зураг байна гэдгийг тагаар заана. Энэ таг дангаараа буюу тегсгөх таггүй бичигддэг.

Жишээ нь:

```
<html>
<body> Энэ бол миний зураг : <br>

</body>
</html>
```

Мөн зургийнхаа урт ергөн, хүрээ, зэрэгцүүлэлт зэргийг зааж өгч болно. Жишээ харья :

```
<html>
<body>
<br> <!--Huree zaaj bna -->
<br> <!--Urt,
ondoriig zaaj bna -->s
<br> <!--ene n zurgандаа
tailbar hij bna Hereh tanii zurag yamar negen shaltgaanaar haragdahgui
yed ene tailbar haragdana. -->
<br> <!--zurgийн зуун, баруун талд
yamar zai avahiig zaaj bna -->
<br> <!--zurgийн deed,dood талд
yamar zai avahiig zaaj bna -->
</body>
</html>
```


Зургаа холбоос болгохдоо өмнөх хичээл дээр үзсэн <a тагыг
ашиглана.

Солигддог зураг гэж юу вэ?

Та өөр вэбүүдийг үзэж байхдаа ажигласан бол хулганы заагч очих
вед өөр өнгөтэй ч юмуу болж өөрчлөгддөг. Энэ нь солигддог зургыг
ид шид юм. Солигддог зургыг ихэвчлэн Вэбний меню хийвэл илүү
их тохиромжтой байдаг. Үүнийг ганц HTML ашиглан хийх
боломжгүй харин JavaScript ашиглан хвссэн үр дүнгээ авч чадна.
JavaScript ийн талаархи хичээлийг та манай хичээлийн бүлгүүдээс
олж үзэх боломжтой. Доор жишээ болгон нэгэн хэсэг кодыг тавилаа.

```
<html>
```

```
<head>
```

```
<title>Rollover image test</title>
```

```
<script language="JavaScript" type="text/JavaScript">
```

```
function MM_swapImgRestore() { //v3.0
```

```
var i,x,a=document.MM_sr;
```

```
for(i=0;a&&i<a.length&&(x=a)&&x.oSrc;i++) x.src=x.oSrc ;
```

```
}
```

```
function MM_preloadImages() { //v3.0
```

```
var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
```

```
var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0;
```

```
i<a.length; i++)
```

```
if (a.indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a;}}
```

```
}
```

```
function MM_findObj(n, d) { //v4.01
```

```
var p,i,x; if(!d) d=document;
```

```
if((p=n.indexOf("?"))>0&&parent.frames.length) {
```

```
d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
```

```
if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++)
```

```
x=d.forms[n];
```

```
for(i=0;!x&&i<d.layers.length;i++)
```

```
x=MM_findObj(n,d.layers.document);
```

```
if(!x && d.getElementById) x=d.getElementById(n); return x;
}
```

```
function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new
Array; for(i=0;i<(a.length-2);i+=3)
if ((x=MM_findObj(a))!=null){document.MM_sr[j++]=x; if(!x.oSrc)
x.oSrc=x.src; x.src=a[i+2];}
}
</script>
</head>
<body>
<a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Image1','zurag1.gif',0)"></a>
</body>
</html>
```

Энд Zurag1.gif бол энгийн байх үеийн төлөвийн зураг, Zurag2.gif бол хулганы заагч очих үед солигдох зураг юм.

HTML дээр хvснэгт vvcгэн хэрэглэх

Вэб хуудсанд ихэвчлэн хvснэгтийг ашиглан мэдээлэл, зургыг з?в байрлуулан хэрэглэдэг. Хамгийн гол нь хаана хэдэн, ямар хэмжээтэй хvснэгт хэрэглэвэл илvv их vр дvнд хvрч чадах вэ гэдгийг л з?в тооцоолж чадвал а хvссэн хуудсаа бvтээж чадна.

Хvснэгт байна. Гэдгийг тодорхойлдог таг бол `<table>` юм. Энэ бол з?вх?н хvснэгт эхэлж, дуусч байна. Гэдгийг заана. Харин м?р эхэлж , дуусч байгааг `<tr>` тагаар, харийн багана буюу нvд эхэлж дуусч байгааг заахдаа `<td>` тагыг бичдэг.

Жишээ болгож нэг 2 м?р ,3 баганатай хvснэгт vvcгэе.

```
<table>
<tr>
<td>1 м?рийн эхний нvд </td>
<td>1 м?рийн хоёрдохь нvд </td>
<td>1 м?рийн гуравдахь нvд </td>
</tr>
<tr>
<td>2 м?рийн эхний нvд </td>
<td>2 м?рийн хоёрдохь нvд</td>
<td>2 м?рийн гуравдахь нvд</td>
</tr>
</table>
```

Дахин танд сануулхад ?мн?х хичээлvvд дээр хэлж байсан нэг таг нь н?г??д?? vргэлж багтсан байх ёстой гэсэн дvрмийг мартагчvй байхыг з?вл?е. Дээрхи жишээн дээр харагдаж тодорхой харагдаж байгаа байх эхлээд м?р эхэлж буйг заагаад дараа нь хэдэн нvдтэйг нь заагаад м?р?? хаана, дахиад м?р оруулах бол м?р?? м?н адилаар хийнэ. Хvснэгт бичих ер?нхий зарчим, бvтэц нь иймэрхvv байна. Мэдээж м?рийн нэг нvдийг н?г?? м?рийн нvдтэй нэгтгэж, нэг нvдийг н?г?? нvдтэй нь нэгтгэн харуулж болно. Жишээ нь тvрvvний жишээн дээрээ эхний м?рийн 1,2 дэхь нvднvдийг нэгтгэе гэвэл :

```
<table>
<tr>
<td colspan="2">Энэ бол нэгтгэгдсэн нvд</td>
<td>1 м?рийн гуравдахь нvд </td>
</tr>
<tr>
<td>2 м?рийн эхний нvд </td>
```

```

<td>2 м?рийн хоёрдохь нүд</td>
<td>2 м?рийн гуравдахь нүд</td>
</tr>
</table>

```

Харин 1 м?рийн эхний нүдийг доод мрийн нүдтэй нэгтгэн харуулнагэвэл :

```

<table>
<tr>
<td rowspan="2">Энэ бол хоёр м?рт хоёуланд нь харагдана. </td>
<td>1 м?рийн хоёрдохь нүд </td>
<td>1 м?рийн гуравдахь нүд </td>
</tr>
<tr>
<td>2 м?рийн хоёрдохь нүд</td>
<td>2 м?рийн гуравдахь нүд</td>
</tr>
</table>

```

Дээрхи нэгтгэх хэсгийг та анхааран сайн ойлгочихвол хүснэгт байгуулахад ямарч асуудал үүсэхгүй. Хүснэгтээ ямар хүрээтэй, нүд хоорондох зай зэргийг мэдээж тодорхойлж болно. Хүрээний хэмжээг зааж г?г?гвй вед автоматаар 1 гэсэн хэмжээтэй байдаг. Хүрээ заах :

```
<table border="0">
```

Хүснэгийн нүдний шугамаас доторхи мэдээлэл хүртэлх зайг заах :

```
<table cellpadding="5">
```

Энэ вед хүрээнээс 5 рх ийн зайтайгаар таны мэдээлэл байрлана. Харин Хүрээний шугамын буюу нэг нүднээс н?г?? нүд эхлэх хүртэл зайг заахдаа :

```
<table cellspacing="4">
```

Хүснэгтийнхээ фон ?нгийг, хүрээний ?нгийг, фон зургийг м?н адил зааж ?гч болно. Фон ?нг? ?г?х :

```
<table bgcolor=" #CCCCCC">
```

Фон зураг ?г?х :


```
<table background="images/table_bg.gif">
```

Хүрээний ?нг? заах :

```
<table border="2" bordercolor="#996600">
```

Олон дэд кодчлолыг зэрэг хэрэглэж нь мэдээж боломжтой. Заавал хвснэгтдээ биш з?вх?н нэг нүдэндээ, м?р?нд?? бас саяны шинж чанаруудыг зааж ?гч хэрэглэж болно.

Линк буюу холбоос хийх

Вэбийн гол ажиллах зарчим бол линк юм. HTML кодчлолд холбоос хийх вед <a> тагыг ашиглана. Бичигдэх хэлбэр нь :

```
<a href="http://www.webmn.net">go to webmn.net</a>
```

дээрхи жишээн дээр got to webmn.net гэсэн вгэн дээр дарах вед вэбийн URL руу холбосон байна. Href="" гэсэн хэсэгт оорийн холбох гэж буй хуудсыг хаягыг бичнэ.

```
<a href="index.html">Index huudas</a>
```

```
<a href="contact.html">Contact me</a>
```

```
<a href="../index.html">Index huudas</a>
```

```
<a href="../page/index.html">Index huudas</a>
```

энд сүүлийн хоёр холбоосийг анхаарч харах хэрэгтэй. Учир нь эхний жишээ тухайн фолдерт байгаа хуудасруу холбож байхад сүүлийнх нь тухайн фолдерийн агуулсан фолдерт байгаа хуудасруу, тэр фолдерт байгаа фолдер доторх хуудасруу холбож байна. Ж.нь :

```
http://www.jishee.com/jishee/index.html >>
```

```
http://www.jishee.com/index.html
```

```
http://www.test.mn/ner/index.html >>
```

```
http://www.test.mn/page/index.html гэсэн вг юм.
```

Зарим вэбийг вэжэ байхад линк дарах вед шинэ цонх нээгдэж хуудсаа харуулдагаар хийсэн байдаг. Энийг хийхийн тулд <a таг дундаа Target шинж чанарыг ашиглана.

```
<a href="new_page.htm" target="_blank">New window</a>
```

дээрхи жишээн дээр target="_blank" гэж шинэ цонхонд нээгдэхийн зааж байна.

Зангуу хийх , Зангуу гэж юу вэ?

Ихэнх вэб хэрэглэгчид тухайн вэбийн вээхдээ Scrollbar –ыг гвйлгэн, ходолгож мэдээллээ вээх сонирхолгвй байдагыг судалгаагаар тогтоосон байдаг. Та магадгвй нэг хуудсанд асар урт хэмжээтэй мэдээлэл, текст байрлуулах болсон вед энд тэнд нь дээшээ буцах, тэр хэсэгт очих гэх мэт хуудас доторх холбоосыг хийж ?г?х хэрэгтэй. Ингэснээр тухайн хэрэглэгч таны хуудсанаас хэрэгтэй мэдээллээ түргэн хугацаанд олж авч чадна. Мон хвссэн хэсэгтээ хврч чадна. Та лав бусад сайтуудыг вээж байхдаа ажигласан байх энд дэрж дээшээ очно уу, дээшээ, top гэх мэт холбоосууд байдаг. Эдгээрийг зангуу гэж нэрлэнэ. Зангуу хийх :

```
<body>
<a name="deed">Top</a>
<b>Ene bol huudasnii ehlel</b>
<a href="#22">22 –r hesegt ochih</a>
<br><br><br><br> <br><br><br><br><br> <br><br><br><br><br>
<br><br><br><br><br><br>
<a name="22"><b>Ene bol 22-r heseg</b></a>
<a href="#deed">Deeshee ochih</a>
</body>
```

Дээрхий кодыг та туршаад вээрэй. Их хэмжээтэй мэдээлэлтэй вэб хуудсанд зангууны хэрэг их гардаг.Мон оор хуудаснаас ногоо хуудасныхаа тухайн хэсэгрвв холбож болно. Энийг сайтын мэдээ бичх зэрэгт ихэвчлэн ашигладаг.Жишээ нь омнох кодоо та medee.html гэж хадгалсан гэж бодъё. Тэгвэл index.html хуудсандаа доорхи кодыг бичиж вэнэ вв?

```
<p align=center>Medee</p>
Aguulga<br>
1-r buleg<br>
...
<a href="medee.html#22">22-r buleg</a><br>
```

Ажиллуулж вэхэд вр дvн нь харагдаж байгаа байх. Зангуу болон холбоосыг ингэж ввсгэн хэрэглэдэг.

HTML жагсаалт үүсгэх

HTML кодчлолд эрэмблэгдээгүй жагсаалтыг кодчлол тодорхойлно. Харин эрэмблэгдсэн жагсаалтыг таг тодорхойлно. Жагсаалтын элемент бүр тагт байрлана.

Энгийн жишээ үзье :

Монголын томоохон сайтууд

```
<ul>
<li>Asuult.net</li>
<li>Mongolduu.com</li>
<li>Mongolmedia.com</li>
<li>Mongoliatoday.mn</li>
<li>Newspapers.mn</li>
</ul>
```

харин эрэмблэгдсэн буюу дугаарлагдсан жагсаалт нь :

Сайн б?хчүүд

```
<ol>
<li>Бат-эрдэнэ аврага</li>
<li>? с?хбаяр</li>
<li>Сүхбат</li>
</ol>
```

гэсэн байдлаар жагсаалтаа үүсгэдэг. Эрэмблэгдсэн жагсаалтын тэмдэглэгээний т?рлийг ??рчл?хд?? type ыг бичиж ?гн?. Ж.нь:

Дугуйгаар

```
<ul type="disc">
<li>Монгол</li>
<li>Орос</li>
<li>АНУ</li>
</ul>
```

Хоосон дугуйгаар

```
<ul type="circle">
<li>Монгол</li>
<li>Орос</li>
<li>АНУ</li>
</ul>
```

Д?рв?лжин дүрсээр

```
<ul type="square">
<li>Монгол</li>
```

```
<li>Орос</li>
<li>АНУ</li>
</ul>
```

Ўвний ижилээр эрэмблэгдсэн жагсаалтын тэмдэглэгээг ??рчл?ж болно.

```
<ol> <!-- тоогоор дугаарлана. -->
<ol type="A"> <!-- том үсгээр дугаарлана. -->
<ol type="a"> <!-- жижиг дугаарлана. -->
<ol type="I"> <!-- Том ром тоогоор дугаарлана. -->
<ol type="i"> <!-- Жижиг ром тоогоор дугаарлана. -->
```

М?н та жагсаалт дотор давхар жагсаалтыг үүсгэн ашиглаж болно.

Үсгийн хэлбэр, хэмжээ, ?нгийг тодорхойлох

HTML кодчлолд текстийн ?нг? хэлбэрийг тодорхойлохдоо тагыг ашиглана. кодчлолоор ?нг?, хэмжээ, хэлбэрийг зэргэ зааж ?гч болдог. Face- Verdana, Arial, Tahoma .. гэх мэт фонтуудын нэрийг бичиж ?гн?. Нэг биш олон т?рлийн шрифт ашиглах бол таслалаар тусгаарлан бичиж ?г?х шаардлагатай. Хэрэв хэрэглэгчид таны бичсэн шрифт байхгвй бол таслалаас хойшхи шрифт хэвжүүлэгддэг. Ж.нь : та Arial mon, Impact гэсэн хоёр шрифтыг бичсэн байхад ??р хэрэглэгчид эхний Arial mon гэсэн шрифт байхгвй вад дараагийн Impact гэсэн шрифт хэвжүүлнэ. Тэгэхээр та шрифт сонгож авахдаа аль болох хмүүст түгээмэл байдаг шрифтыг сонговол тохиромжтой. Түгээмэл байдаг шрифт : Arial, Tahoma, Verdana

Бичихдээ

```
ene n yamarch font hevjigui
<font face="Verdana"> ene textiin fot n Verdana yum</font>
```

гэсэн байдлаар бичигдэнэ. Ўвний адилаар үсгийн ?нг? хэмжээг таг дотор бичдэг. ? нгийг т?л??ний нэрээр ?гч болно / red , black, blue, orange / . Эсвэл 16 тын кодоор ?нг?ний кодыг бичнэ. 16 тын код гэдэг нь 0-9 A-F үсэг хүртэл 10 цифр 6 үсгээр илэрхийлдэг тоолол юм. Жишээ нь улаан ?нг? гэхэд : #FF0000 байна.

Энийг талбарлавал эхний FF хэсэг нь улаан ?нгийн оролцох хэмжээг заана. Голын хоёр нь ногоон ?нгийн оролцох хэмжээ. Сүүлийнх нь х?х ?нг?ний холилдох хэмжээг тус тус заадаг. Хар ?нг? : #000000, Цэнхэр : #0000FF

Бичихдээ :

Та ногоон онгийн бичгийг харж байна
мон адил ногоон онго

Ўсгийн хэмжээг заахдаа SIZE гэж бичнэ. Size нь 1-7 хвртэл хэмжээг авна. 1-жижиг, 7- том болгоно. Бичих нь :

та том хөмгөж харж байна

Эдгээр дэд тагуудаа бүгдийг нь зэргэ хэрэглэж болно. Ж.нь :

engiin text

 ene bol avarga том
ulaan ongotoi impact fonttoi text

гэх мэтээр нэг таг дотор дэд тагуудыг бичин хэвжүүлэлт хиж болдог.

Текст хэвжүүлэх тагууд

Текст хэвжүүлэх тагууд

Тэксгийг тод, налуу, доогуур зураастайгаар хэвжүүлэх гэвэл доорхи тагуудыг хэрэглэнэ.

 тод үсгээр бичих

ene bol buduun

<i> налуу үсгээр бичих

<i>ene naluu</i>

<u> доогуураа зураастай текст

duugaараа zuragdana

Мэдээж HTML тагуудыг давхар,цуг хэрэглэж болно. Харин анхаарах зүйл бол нэг таг нь н?г??гийн дотор бүрэн багтсан байх ёстой. Ж.нь:

<u>ene buruu aldaatai</u>

Энэ бол буруу , нэг нь н?г??д?? багтсан байх хэрэгтэй. З?в бичлэг нь :

```
<b><u>Narin одоо зов байна</u></b>
```

гэх мэтээр бичнэ.

HTML кодчлолд м?р шилжихийг харуулах таг нь
 юм. Энэ нь дангаараа бичигддэг . ж.нь :

```
ene enhii mor<br>ene bol hoyordoh mor
```

гэж бичнэ. Догол м?р байна гэдгийг тодорхойлохдоо <p></p> таг бичнэ.

```
ene neg heseg text<p>ene bol dogol mor</p>ene neg heseg text
```

Гарчигийг тодорхойлохдоо <h> таг хэрэглэнэ. <h> нь 1-6 хвртэл утга авна.

```
<h1>Ene garchigiin hemjee 1</h1>
```

```
<h2>Ene garchigiin hemjee 2</h2>
```

.....

```
<h6>Ene garchigiin hemjee 6</h6>
```

м?н элдэв кодыг бичихдээ <code> тагыг хэрэглэнэ.

```
<code>
```

```
function name {  
if ($a=11) echo "$a";  
}
```

```
</code>
```

гэнэ. Текстийн хоорондуур тусгаарлах зураас оруулахыг хэсвэл <hr> тагыг хэрэглэж болно.

```
ene minii textuud<hr>ene bas oor negen textuud
```

бас <small><big> гэсэн тагуудыг байна. Эдгээр нь жижг тэкс болон том хэмжээтэй текст бичихэд хэрэглэгдэнэ.

```
heviin text<big>ene bol mash tom text байна</big>
```

```
<small>ene bol ih jijighen text байна</small>
```

HTML бичих дүрэм

HTML кодчлол нь вэб хуудасыг хэрхэн харуулахыг заасан байдаг гэж ярьж байсан.

Ерөнхий бүтэц нь :

```
<html>
<head>
толгойн хэсэг
</head>
<body>
их биеийн хэсэг
</body>
</html>
```

HTML хуудсыг үндсэн хоёр хэсэгт хувааж болох юм. Толгойн хэсэг ба их биеийн хэсэг гэж. Толгойн нэсэг нь :

```
<head>...</head>
```

гэсэн таган дотор бичигдэнэ. Толгойн хэсэгт вэб хуудсын гарчиг, хайх түлхүүр үг, тайлбар, UNICODE , хэлбэр ашиглах бол хаяг гэх мэт зүйлсийг бичдэг. <head> хэсэгт бичигдсэн зүйлс вэб хуудсан дээр харагдаггүй. Их биеийн хэсэг нь :

```
<body>
..
</body>
```

гэсэн таган дотор бичигдэнэ. Энэ хэсэгт бичигдсэн зүйлс нь вэб хуудсыг бүрдүүлж байгаа гэж ойлгож болно. Энэ хэсэгт бүх тагууд кодчлолуудыг хэрэглэн бичиж вэб хуудсаа үүсгэнэ. HTML тагууд <ба> хашилтан дотор бичигдэх ба ихэвчлэн хосоороо бичигддэг.

Жишээ нь : <u>text</u>, <p>parag</p>

гэж байна.

<head> хэсэгт бичигдэх кодчлолууд :

```
...
<head>
<title>Ene bol garchig</title>
</head>
..
```

<title> кодчлол нь хуудасны гарчигыг заана. ???р хэлбэл Title bar дээр гарах тэмдэгтүүд юм.

```
...
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
..
```

энэ бол вэб хуудсыг харуулах Unicode ийг заадаг. Монгол Крилл үсэгийн Unicode нь Windows-1251 юм. Та Unicode оо ???р бичээд крилл үсгэр бичсэн бол крилл үсгүүдийг таниж харуулж чадахгвй. Тэгэхээр крилл үсгээр бичсэн тэкт агуулсан бол Unicode оо тохируулах хэрэгтэй.

```
...
<head>
<meta name="keywords" content="free tutorials, php codes">
</head>
..
```

энэ бол таны вэб хуудсыг хайх түлхүүр үгс гэж ойлгож болно. Хайх үгсийг content="" дотор бичнэ.

```
...
<head>
<meta name="description" content="ene bol tailbar8 hailtand oldoh"></head>
..
```


ЭНЭ БОЛ ВЭБ ХУУДСЫН ТАЙЛБАР, ХАЙЛТАНД ОЛДОХОД ХАРУУЛАХ ТАЙЛБАР
ЮМ.

```
...  
<head>  
<script language="JavaScript"> code code code </script>  
..
```

Энд JavaScript кодчлол бичиж ашиглаж болно.

HTML Б?лэг - нэгд?гээр хичээл

Notepad нээгээд дараах зааврын дагуу ажиллана уу.

```
<html>
</html>
```

??нийг tag гэж нэрлэдэг б?г??д нээх, хаах гэсэн хоёр т?р?лтэй. Бидний дээр харж байгаачлан хаах tag ??сгэхийн тулд нээх tag дээр / нэмдэг. Б?гд биш ихэнх tag хаах tag-тай байдаг. Ер нь tag-ийг browser-т х?рв??лэлтэнд хэрэглэгддэг жижиг хэмжээний текст гэж ойлгож болно. Browser <html> болон </html> гэсэн tag-уудын хоорондох б?х з?йлийг HTML document гэж ойлгоод энэ т?р?лд нь тохируулан х?рв??лдэг. Browser ??р ??р tag-уудыг ??р ??р??р х?рв??лдэг.

HTML document б?рд **head** tag хосоороо байх ёстой.

```
<html>
<head>
</head>
</html>
```

М?н **title** tag гэж байх ба энэ нь ихэвчлэн гарчиг хийхэд хэрэглэгдэнэ.

```
<html>
<head>
<title></title>
</head>
</html>
```

Харин web-ийн маань ихэнх хэсгийг эзлэх текстийн хэсгийг бид **body** tag дотор хийдэг. ??нийг жишээгээр авч ?звэл:

```
<html>
<head>
<title>My first HTML page!</title>
</head>
<body>
Hello HTML world!
</body>
</html>
```

Одоо ??нийгээ хадгална. Гэхдээ text document хэлбэрээр биш html document хэлбэрээр хадгалах хэрэгтэй ш???. ? ?рийн анхны хуудсандаа **page1.html** гэсэн нэр ?г??д хатуу дискнийхээ хаа нэгтээ нэг шинэ folder-т хадгалаарай.

Notepad-ынхаа **File** цэснээс **Save As** командыг сонгоно. Save As dialog box гарч ирэх ба энд бид шинэ хавтас ??сгэж болно. Ингэхдээ Save As dialog box-ын New Folder гэсэн icon дээр дарах хэрэгтэй.

New Folder icon –ы харагдах хэлбэр--> .

Шинэ folder-оо ??рийн х?ссэнээрээ нэрлэ. Ингээд ??ссэн шинэ хавтасаа нээхийн тулд хоёр удаа дарах хэрэгтэй. Одоо л бид шинэ folder-тоо ??рийн анхны web хуудсаа хийх боломжтой боллоо. ??ний тулд **File name:** гэсэн хэсэгт **page1.html** гэж бичээд **Save as type:** хэсгийг **All Files(*.*)** болгох хэрэгтэй. Одоо бол Enter дараад боллоо.

Хэрвээ чи Internet Explorer хэрэглэдэг бол бидний файл доорх хэлбэртэй харагдах ёстой.



page1.html

??н дээрээ хоёр дарахад ??рийн хийсэн web хуудсаа харж чадна. Одоо ??рийн файлаа **.html** ?рг?тг?лтэй хадгалсан эсэхээ ахин нэг шалгаарай. Дараагийн хийх з?йл бол ??рийн web хуудсан дээр зарим нэг з?йлсийг нэмэх явдал юм.

1. ? ?рийн хадгалсан html файлаа олоод icon дээр нь хоёр дараарай. Эсвэл...
2. browser дээрх **File** цэснээсээ **Open File** командыг сонгон файлаа олж сонгох замаар нээж болно. (**page1.html**).

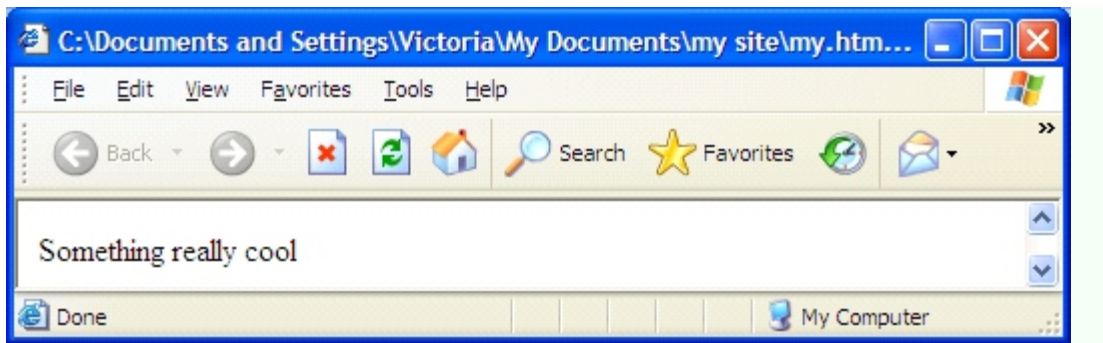
Энэ хичээлийн хамгийн чухал гурван з?йлийг ахин тодруулья.

1. Бидний хийсэн з?йл бол ерд??л ганц гарчиг б?хий хуудас. Гэхдээ энэ нь web хуудсанд байх ёстой з?йлсиний хамгийн эхнийх нь.
2. Гарчиг бол browser цонхны хамгийн дээд хэсэгт байрладаг з?йл билээ.
3. Search engine-ууд web хуудсуудыг rank-лахдаа чиний web дэх б?х з?йлсийн дотроос **гарчигийг** анхаарч ?здэг гэдгийг анхаарах хэрэгтэй. Иймээс гарчиг аль болох богино бас тухайн хуудасныхаа агуулгыг б?рэн гаргаж байх ёстой.

ХОЁРДУГААР ХИЧЭЭЛ

Бичлэгээ арай багасгах ??днээс одоо би з?вх?н **<body>** tag доторх з?йлсийг харуулах болно. **<html>**, **<head>** болон **<title>** tag-уудыг орхино. Гэхдээ эдгээрийг чи бас орхиж болохг?й ш??. Something really cool. гэж бичье.

```
<body>  
Something really cool  
</body>
```

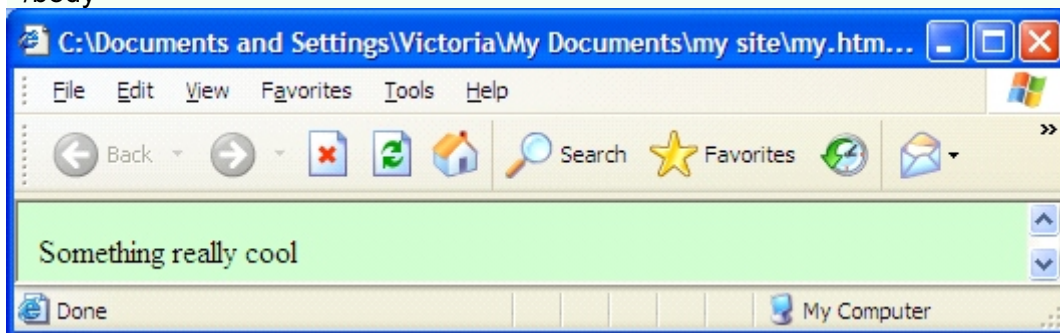


АНХААРУУЛГА: Файлдаа ??рчл?лт хийх б?рдээ ??нийгээ хадгалаад browser дээрээ байгаа Refresh/Reload товчийг дар. Заримдаа **refresh товч** даргах нь ?рд?н?гд?гг?й. Энэ тохиолдолд:

Internet Explorer хэрэглэгчид: CTRL дарангаа Refresh дар.
Netscape/FireFox хэрэглэгчид: SHIFT дарангаа Reload дар.

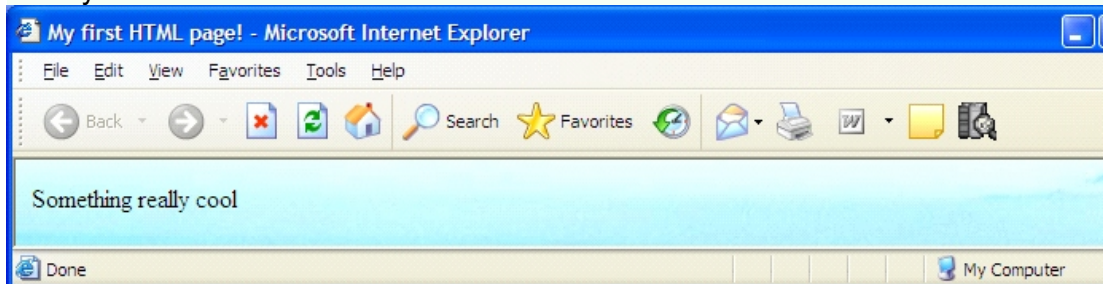
Одоо хийх з?йлс маань суурь ?нг? ??рчилж сурах.

```
<body bgcolor="#cffff">  
Something really cool  
</body>
```



#cffff бол цайвар ногоон ?нг?. Бид дэвсгэрийн ?нгийг ??рчл?х??с гадна дэвсгэрт нь зураг оруулж болно. Энэ ?ед суурь болох зураг маань бидний html файльтай нэг хавтасанд байх ёстой.

```
<body background="bluehills.jpg">  
Something really cool  
</body>
```



Зургийг харуулахын тулд browser т??ний байрлалыг мэдэж байх ёстой. Одоохондоо бид зургийг HTML document (**page1.html**)-тай нэг хавтасанд хийж байна.

Хэрвээ зургийн хэмжээ жижиг байвал зургийг олон дахин хэвлэж, дэлгэцийг д??ргэдэг.

АСУУЛТ: Зарим хуудаснууд дээр дэвсгэр зураг нь х?д?лг??нг?й, харин хуудасны бусад элемент??д ??н дээг??р г?йдэг. ??нийг яаж хийх вэ?

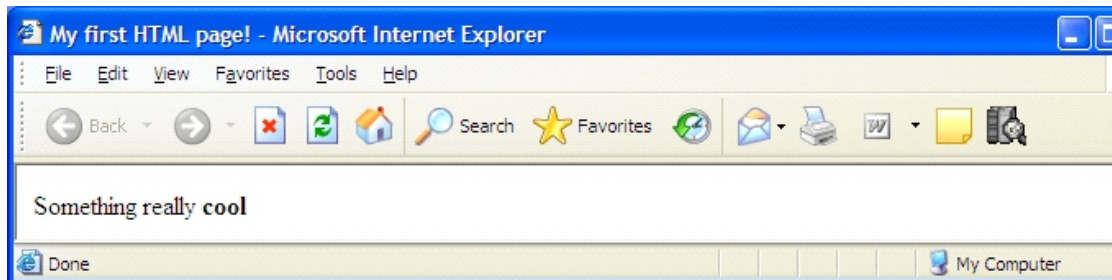
ХАРИУЛТ: Энэ их амархан... з?гээр л `style="background-attachment:fixed"`-ийг `<body>` tag дээр нэм.

```
<body background="mybackground.gif" style="background-attachment:fixed">
```

ГУРАВДУГААР ХИЧЭЭЛ

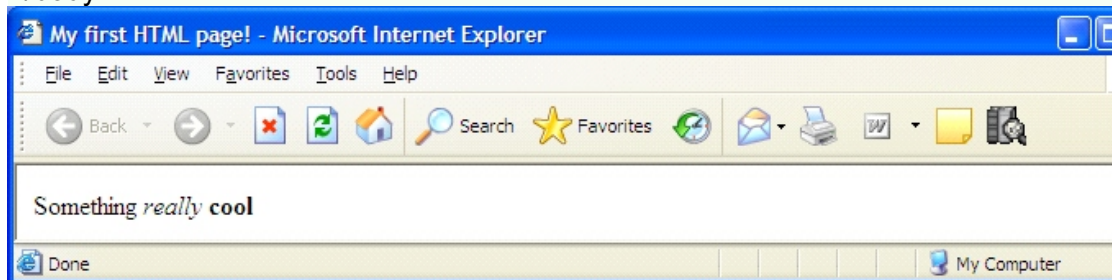
Бид текстийн хэлбэрийг ??рчилж, ??р??р хэлбэл **bold** болгож чадна.

```
<body>
Something really <b>cool</b>
</body>
```



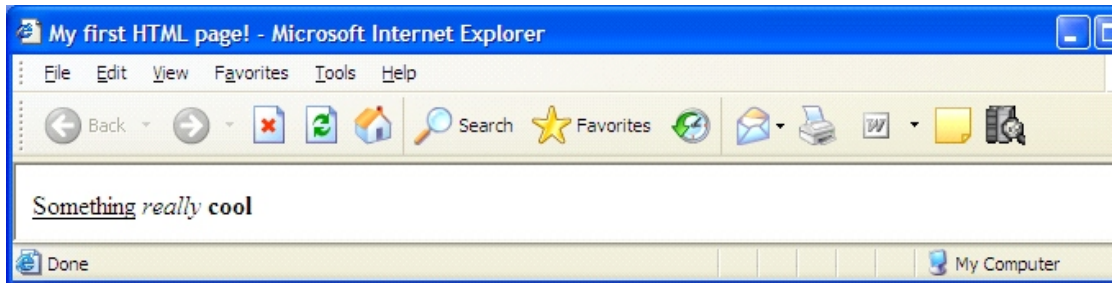
Энд `` эхэлсэн газраас б?гдийг bold болгоод `` дээр ??нийгээ зогсоо бид browser-т команд ?гч байгаа юм. ? ?р??р хэлбэл, бид browser-т `` доторх б?х элементийг bold болгон ?з??л гэж зааварчилж байна. *Italics*-ийн хувьдм?н ижил зарчим ?йлчилнэ.

```
<body>
Something <i>really</i> <b>cool</b>
</body>
```



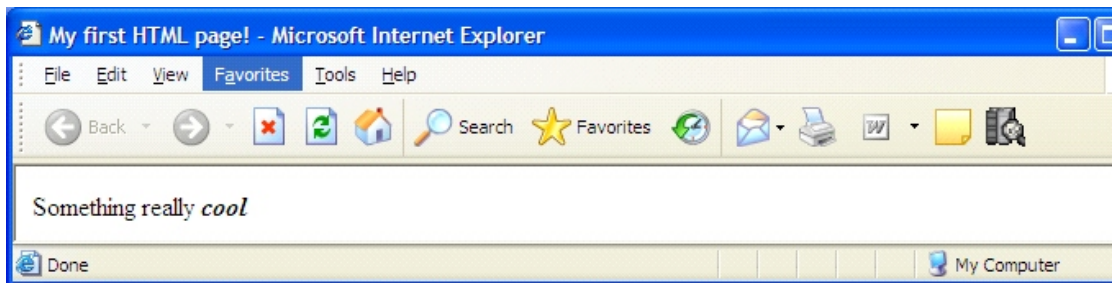
...underlining.м?н адил.

```
<body>
<u>Something</u> <i>really</i> <b>cool</b>
</body>
```



Бид дээрх tag-уудыг м?н хослуулан хэрэглэж болно.

```
<body>
Something really <i><b>cool</b></i>
</body>
```



Энд хослолыг з?в хэрэглэсэн байна. Гэвч зарим тохиолдолд х?м??с tag-ийн хослолыг буруу хэрэглэн, browser-т буруу команд ?гд?г. Тодруулан авч ?ье.

```
<this><that>cool</this></that><-- Давхардсан tag.... буруу
```



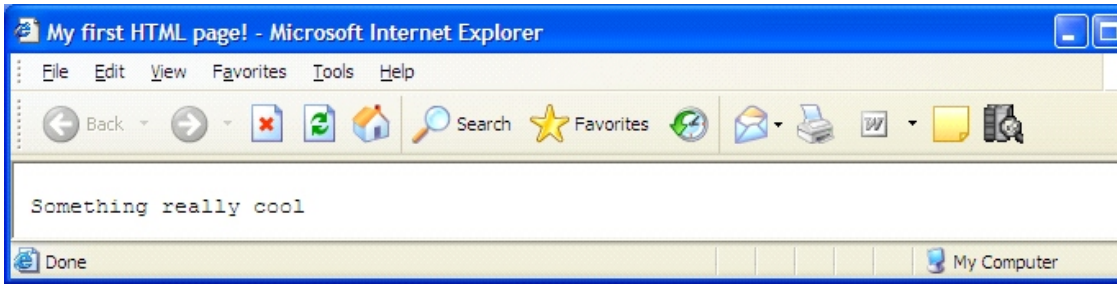
```
<this><that>cool</that></this><-- Холбогдсон tag.... з?в
```



Д? Р? ВД?ГЭЭР ХИЧЭЭЛ

Text effect-ийн хамгийн тохиромжтой хэлбэр бол mono-spaced font буюу Typewriter Text юм.

```
<body>
<tt>Something really cool</tt>
</body>
```

Эсэг болгон хэвтээтэнхлэгийн дагуу ижил хэмжээ авна.

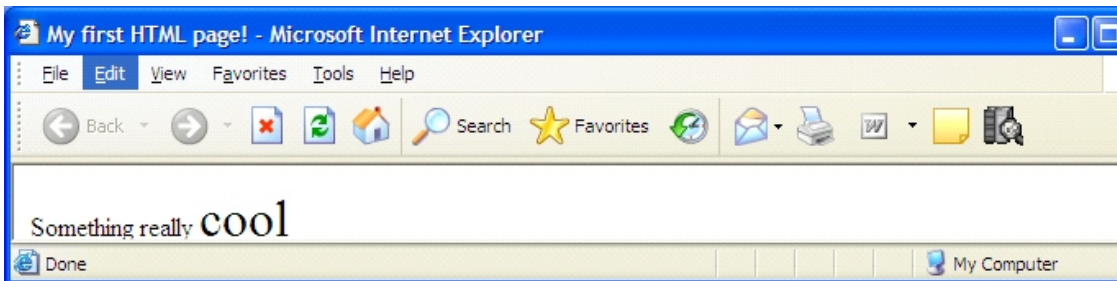
Энэ бол жирийн фонд ->
 iiiiiiiii
 oooooooooo
 mmmmmmmmm

Харин энэ бол monospaced type ->
 iiiiiiiii
 oooooooooo
 mmmmmmmmm

Фондын хэмжээг ?рчл?хийн тулд бид **** tag-ийг хэрэглэнэ.

??н дээрээ **size** attribute-ийг нь зааж ?гн?.

```
<body>
Something really <font size="6">cool</font>
</body>
```



Фондын 7 хэмжээг бид хэрэглэж болно.

teeny tiny	small	regular	extra medium	large	real big &	yelling!
1	2	3	4	5	6	7

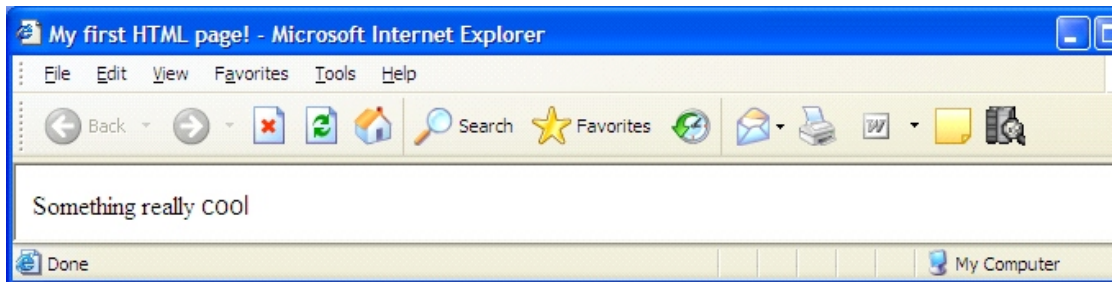
Энд хоёр з?йлийг тодруулах шаардлагатай. Нэгд, **<tag>** нь browser-т ямар нэг з?йл ?рчл?х гэж буйг заадаг. Харин ??н доторх **attribute** нь browser-т ??нийг хэрхэн ?рчл?хийг зааж ?гд?г.

Хоёр дах з?йл нь default утгуудын тухай юм. Бидний мэдэж буйгаар бол *default утга* нь бид ямар нэг з?йлийг ??р??р хийхийг зааж ?г??г?й ?ед browser-ийн авч хэрэглэдэг утгууд юм. ??нд дурьдаж болох жишээ нь фондын хэмжээ юм. Default фондын хэмжээ нь 3 (ихэнх тохиолдолд)

байдаг. Хэрэв чи фондын хэмжээг зааж ?г??г?й бол энэ утга шууд 3 болох ба энэ хэмжээгээрээ дэлгэцэн дээр д?рслэгддэг.

Browser б?р default фондын хэмжээс??дтэй байдаг. ??нд – фондын нэр, хэмжээ, ?нг? гэх мэт. Ихэнх browser дээр эдгээрийн default нь Times New Roman буюу Verdana, 12pt (энэ нь бидний тохиолдолд 3 болдог) ба ?нг? нь хар байна. Иймээс бид фондын нэр, ?нг? зэргийг м?н зааж ?гч болно.

```
<body>
Something really <font face="arial">cool</font>
</body>
```



Бидний хуудсандаа хэрэглэсэн фондыг ?эж буй хэрэглэгч ??рт нь байгаа тохиолдолд з?вх?н харж чаддаг. ??р??р хэлбэл, бидний б?тээсэн хуудсыг INMERNET-ээр дамжуулан ?эж буй хэрэглэгчийн компьютер дээр бидний хэрэглэсэн фонд байхг?й бол тэд з?вх?н default фондыг харна. Иймээс фонд хэрэглэхдээ их анхааралтай хандах хэрэгтэй. Arial болон Comic Sans MS фондууд нь Windows ?йлдлийн системтэй цуг т?гээгддэг учраас их т?гээмэл байдаг. Гэвч бид эрсдэлээ бууруулж ?г?хийн тулд нэгээс олон тооны фондыг нэр заан хэрэглэж болно.

```
<font face="arial,helvetica,lucida sans">Hidee Ho</font>
```

Дээрхийг тайлбарлавал, browser эхлээд arial-ийг хайна, хэрэв олвол т??нийгээ ашиглана. Харин олохг?й бол Helvetica-г хайна. ??нийгээ ч м?н адил олохг?й бол lucida sans-ийг хайна. ??нийг ч м?н адил олохг?й бол ??рийн default font-ийг ашиглана.

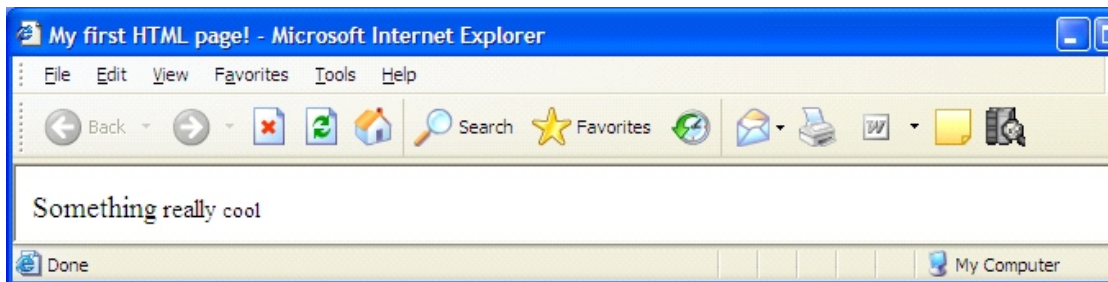
Ихэнх компьютерт байдаг буюу хамгийн т?гээмэл хэрэглэгддэг фондуудыг дор ?з??лэв.

Arial	Comic Sans MS	Lucida Console
Arial Black	Courier New	Tahoma
Arial Narrow	Georgia	Times New Roman
Bookman Old Style	Impact	Trebuchet MS
Century Gothic		Verdana

ТАВДУГААР ХИЧЭЭЛ

Одоо бид syntax-ийн тухай авч үзнэ. **attribute="value"** хэлбэртэй зүйл энд харагдвал энэ утга заавал "" дотор байх ёстой гэсэн үг. Syntax бол маш чухал бөгөөд нэг бяцхан syntax-ийн алдаа бидний маш урт хугацааны хэдлэмгийг бэхэлд нь баллаж болно. Хамгийн эхэнд хэлэх зүйл бол HTML бол INsensitive. Үүнийг хэлбэл, бид ****, **** хоёрыг ээлжлэн хэрэглэж болно. Гэхдээ бөхнийг цэгцтэй байлгахын тулд бөхдийг жижиг үсгээр бичсэн нь дээр байхаа. Сонирхолтой хэсэг үүгээ буцаад орьё. Үүнийг том жижгээр харагдуулахын тулд бид **big** tag болон **small** tag-ийг хэрэглэнэ.

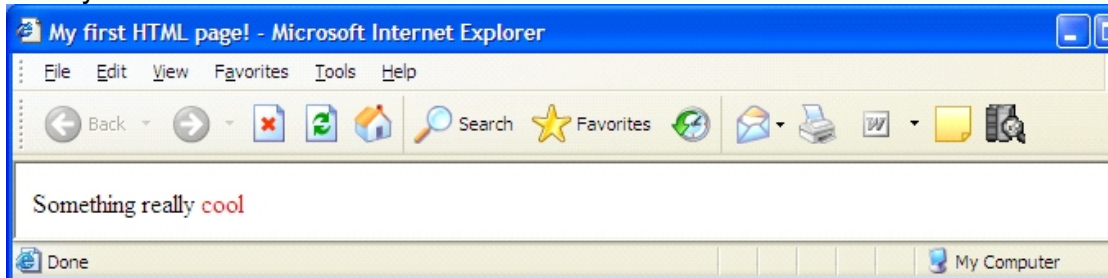
```
<body>
<big>Something</big> really <small>cool</small>
</body>
```



big tag нь текстийг нэг хэмжээгээр томруулах ба, **small** нь нэг хэмжээгээр жижигрүүлнэ.

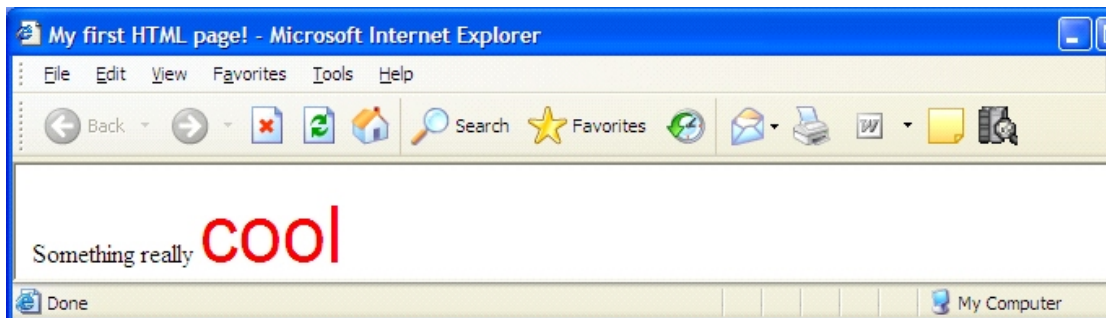
Үүн дурьдсанчлан бид фондын үүнийг үүрчилж болно.

```
<body>
Something really <font color="#ff0000">cool</font>
</body>
```



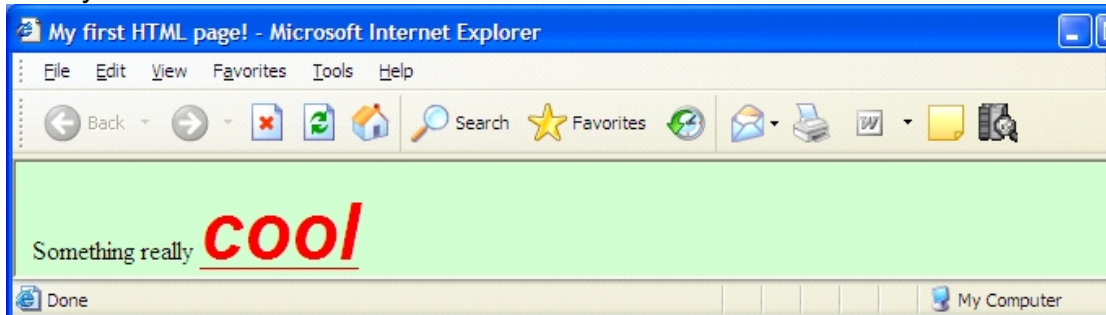
Мөн бид **** tag дотор нэгээс олон **attribute** хэрэглэж болно.

```
<body>
Something really <font color="#ff0000" face="arial" size="7">cool</font>
</body>
```



Цаашлаад:

```
<body bgcolor="#cffff">
Something really <u><i><b><font color="#ff0000" face="arial"
size="7">cool</font></b></i></u>
</body>
```



Энд ахиад хэлэхэд олон тооны tag хэрэглэх ?ед эдгээрийг давхардуулж болохг?й, з?в байрлуулах хэрэгтэй.

```
<tag3><tag2><tag1>Hooha !</tag1></tag2></tag3>
```

Ихэнх тохиолдолд аль tag эхнийх байх нь тийм ч чухал биш. Жишээ нь, хэрэв чи аль нэг ?гийг **red, bold** болгохыг х?сч байгаа бол чи tag-ийн дарааллыг ??рийн х?ссэнээр ?гч болно.

```
<tag2><tag1><tag3>Hooha !</tag3></tag1></tag2>
```

Харин доорх тохиолдлийг давтаж болохг?й.

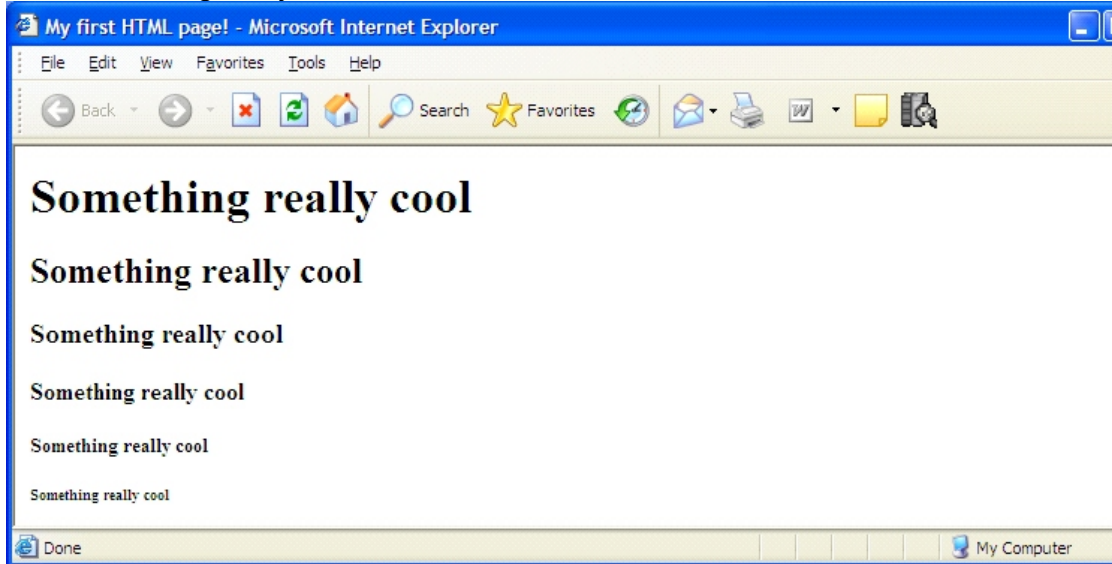
```
<tag3><tag2><tag1>Hooha !</tag3></tag1></tag2>
```

ЗУРГААДУГААР ХИЧЭЭЛ

Заавал дурьдах ёстой бас нэг tag байна. Энэ нь хэсгийн гарчиг буюу heading юм. Эдгээр нь 1-6 гэсэн хэмжээтэй байдаг.

```
<h1>Something really cool</h1>
<h2>Something really cool</h2>
<h3>Something really cool</h3>
<h4>Something really cool</h4>
<h5>Something really cool</h5>
```

```
<h6>Something really cool</h6>
```



Энд хэрэглэж болох нэг зүйл бол **align** бөгөөд бид үүний тусламжтайгаар текстээ эмхэлж чадна.

```
<h2 align="left">Something really cool</h2>  
<h2 align="center">Something really cool</h2>  
<h2 align="right">Something really cool</h2>
```



Browser текстийн үнг, link-ны үнг, active link-ны үнг, айлчилсан link-ны үнг зэрэгт үүнийн default утгуудтай байдаг. Эдгээрийг үзүүлвэл:

Background-ихэвчлэн цагаан

Text –ихэвчлэн хар

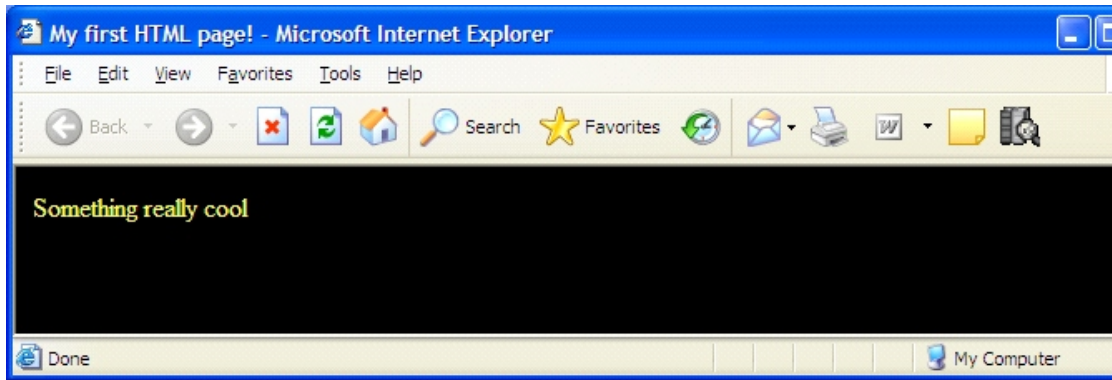
[Link- ихэвчлэн х?х](#)

[Айлчилсан link-ихэвчлэн чернилэн ягаан](#)

[Active link-ихэвчлэн улаан байдаг](#)

Бид б?хэл хуудасны хувьд эдгээр default-ийг **<body>** tag дотор үүрчилж болно.

```
<body bgcolor="#000000" text="#ffff66" link="#00ff33" vlink="#00bb33"  
alink="#00ddff">  
Something really cool  
</body>
```

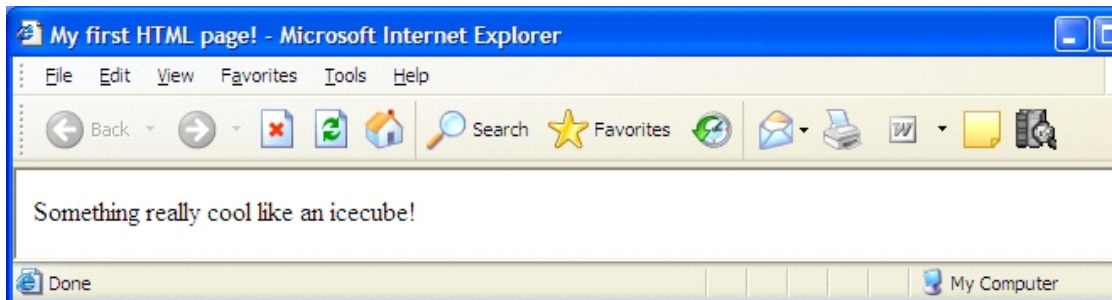



Одоо бид ??рийн хуудсан дахь текстийг х?ссэнээрээ ??рчилж чадахаар боллоо. Бид одоо **ТОМ УЛААН ?Г** эсвэл **жижиг bold ?г** бичиж чадна.

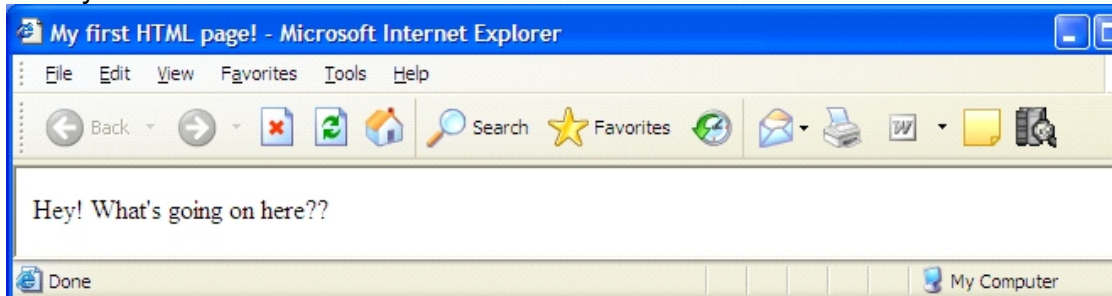
Бид ??рсдийн browser-ийн View/Document Source цэснээс ?зэж буй хуудасныхаа HTML кодыг харж болно. Иймээс янз б?рийн хуудаснуудын кодыг харж, сонирхолтой санагдсан з?йлсийхээ кодыг дуурайн хийж ?зээрэй.

ДОЛДУГААР ХИЧЭЭЛ

```
<body>  
Something really cool  
like an icecube!  
</body>
```

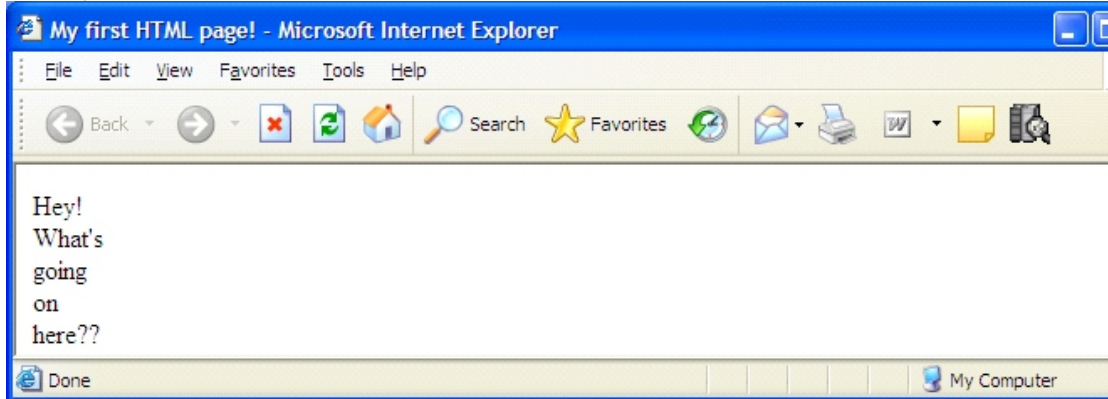


```
<body>  
Hey!  
What's  
going  
on  
here??  
</body>
```



Browser дээрх шиг форматыг ойлгохгүй. Иймээс бид ойлгох аргаар нь буюу код бичиж үгэх хэрэгтэй. Хэрвээ шинэ мөрөөс эхлэхийг хүсч байвал line break авах хэрэгтэй.

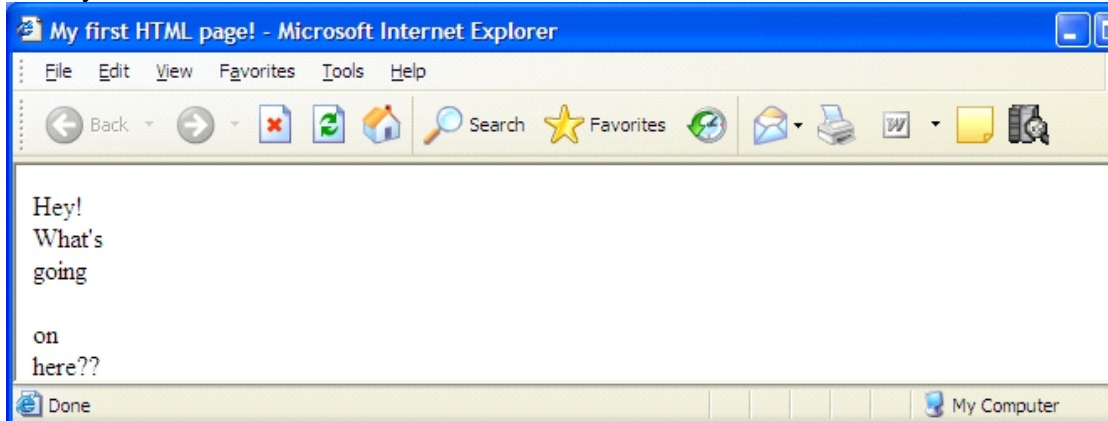
```
<body>
Hey!<br>
What's<br>
going<br>
on<br>
here??
</body>
```



**
**- шинэ мөрөөс эхэл гэсэн үг.

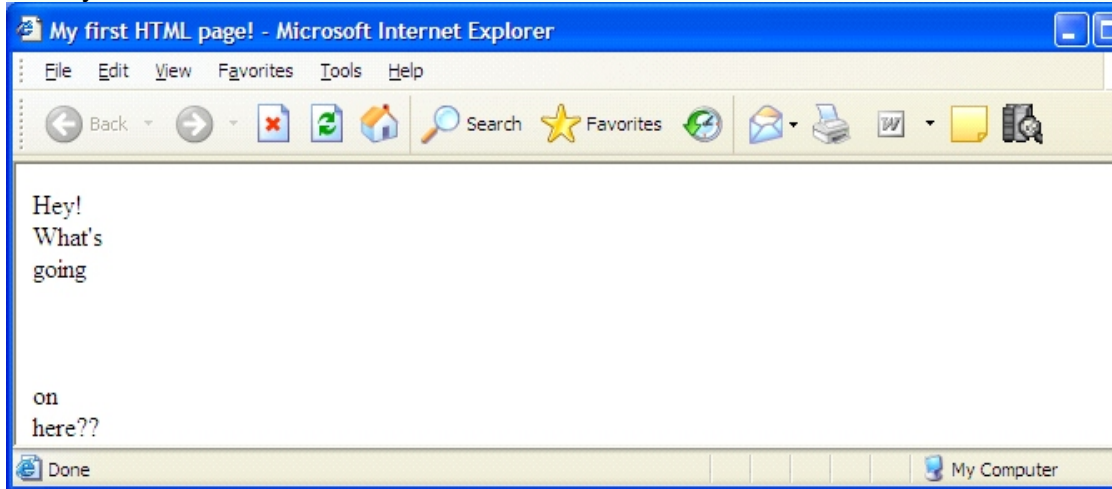
Энд бид **
**-ийг мөн мөр алгасахад хэрэглэж болно.

```
<body>
Hey!<br>
What's<br>
going<br>
<br>
on<br>
here??
</body>
```



Бүүр олон мөр алгасая гэвэл:

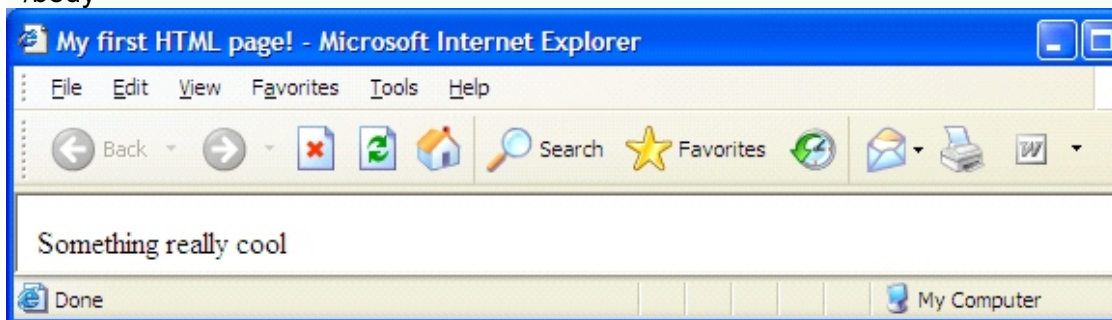
```
<body>
Hey!<br>
What's<br>
going<br>
<br>
<br>
on<br>
here??
</body>
```



НАЙМДУГААР ХИЧЭЭЛ

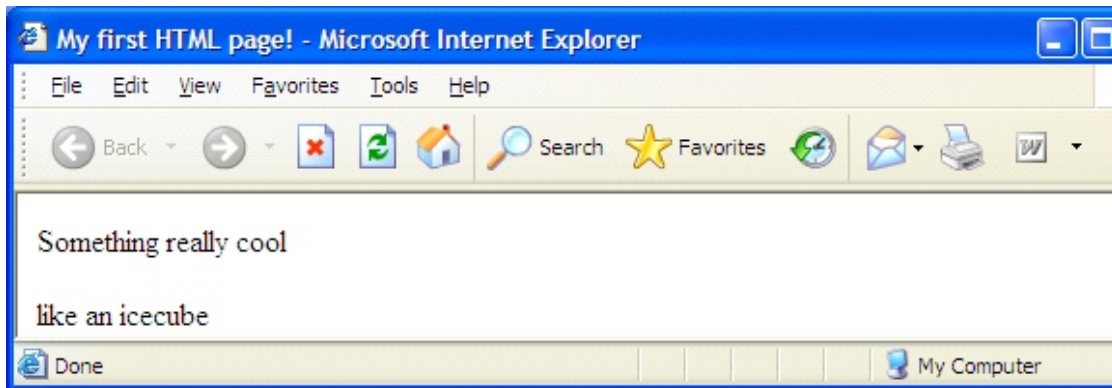
Одоо бид paragraph tag ʘзнэ. Paragraph бол догол мʘрʘʘр тусгаарлагдсан текстийн блок гэж хэлж болно.

```
<body>
<p>Something really cool</p>
</body>
```



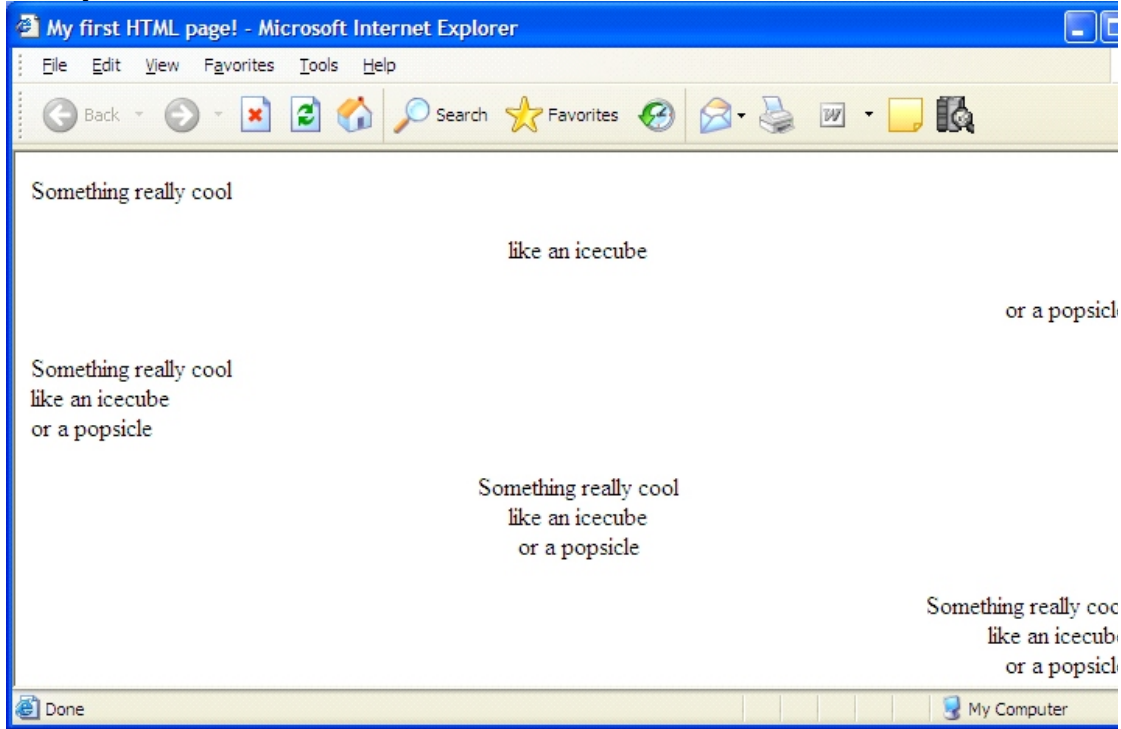
Ихэнх browser-т шинэ paragraph эхэлнэ гэдэг нь мʘр алгасна гэж ойлгогддог.

```
<body>
<p>Something really cool</p>
<p>like an icecube</p>
</body>
```



М?н **<p>** tag-ийг **align** хийхэд хэрэглэж болно.

```
<body>
<p align="left">Something really cool</p>
<p align="center">like an icecube</p>
<p align="right">or a popsicle</p>
  <p align="left">
    Something really cool<br>
    like an icecube<br>
    or a popsicle</p>
  <p align="center">
    Something really cool<br>
    like an icecube<br>
    or a popsicle</p>
  <p align="right">
    Something really cool<br>
    like an icecube<br>
    or a popsicle</p>
</body>
```



Текстийг голлуулах нэг энгийн арга бол **<center>** tag хэрэглэх явдал юм. Center tag-уудын хооронд байгаа б?х з?йл голлодог.

< < бага
> > их
& &
" "

Бид эдгээрийг байнга хэрэглэх шаардлагаг?й. Э?гээр л зарим нэг тэмдэгт??дийг Browser ойлгохг?й байгаа ?ед хэрэглэнэ.

АРАВДУГААР ХИЧЭЭЛ

Одоо ??рсдийн хуудсандаа зураг нэмж ?ье. Хуудсанд оруулж байгаа зураг хуудасны файльтай нэг хавтсанд байх ёстой гэдгийг урьд хэлсэн билээ. Бид доорх зургийг **chef.jpg** нэрээр **page1.html**-ийг хадгалсан хавтсанд хуулья.

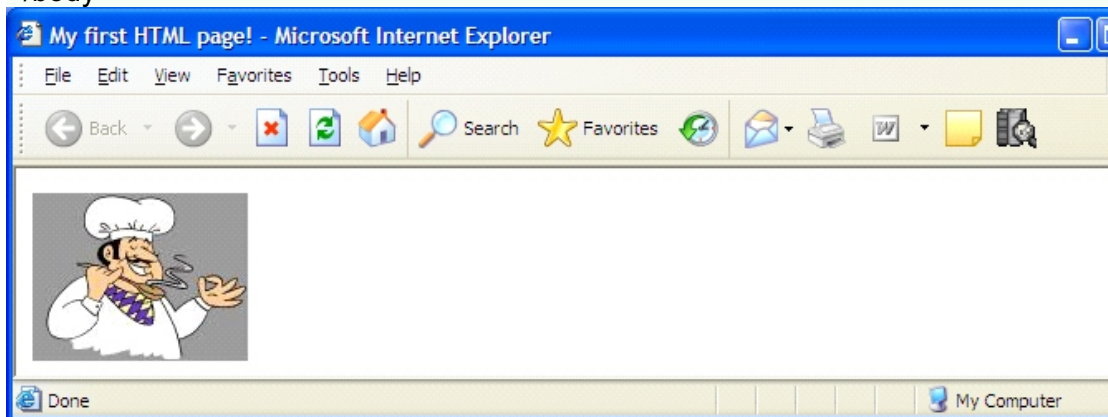


Бид зураг оруулахдаа **** tag-ийг хэрэглэнэ.

```
<body>  
<img>  
</body>
```

Бид ??нээс гадна файлын нэрийг хэмжээтэй нь оруулж ?г?х ёстой.

```
<body>  
  
</body>
```



Энд бид з?вх?н зургийн нэр биш м?н байрлалыг нь зааж ?гч болно. Дээрх жишээн дээр бол **src="chef.gif"** гэсэн нь browser **chef.gif** гэсэн нэртэй зургийг html document байгаа хавтас дотор хай гэсэн ?г юм. Доорх диаграммыг харна уу.



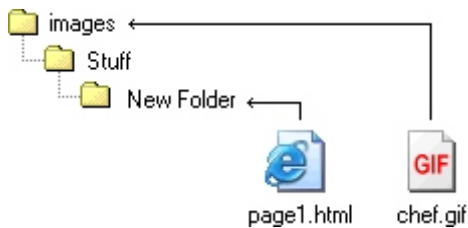
`src="chef.gif"` зураг-г ??рийг нь дуудаж байгаа html document байгаа хавтасан дотроос хайна.



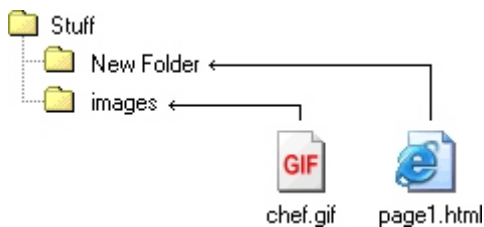
`src="images/chef.gif"` зураг-г ??рийг нь дуудаж байгаа html document байгаа хавтаснаас нэг хавтасан доор байх хавтаснаас хайна.



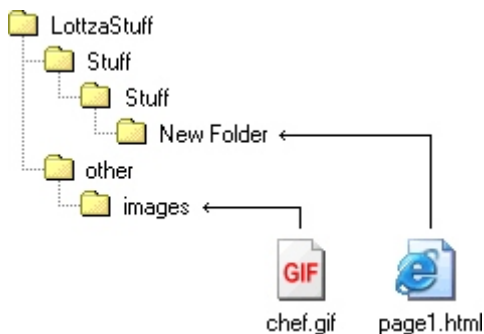
`src="../chef.gif"` зураг-г ??рийг нь дуудаж байгаа html document байгаа хавтаснаас нэг хавтасан дээр байх хавтаснаас хайна.



`src="../../chef.gif"` зураг-г ??рийг нь дуудаж байгаа html document байгаа хавтаснаас хоёр хавтасан дээр байх хавтаснаас хайна.



`src="../images/chef.gif"` зургийг зэргэлдээ байрлалб?хий хавтаснаас хайна.



`src="../../../other/images/chef.gif"` Хар даа.

М?н ??р server дээр байгаа зургийг ??рийн хуудсан дээр байрлуулж болно. ??ний тулдз?вх?н тухайн зургийн байгаа URL-ийг оруулахадболно. Жишээ нь:

<http://www.yourdomain.net/LottzaStuff/images/chef.gif>

АРВАННЭГД?ГЭЭР ХИЧЭЭЛ

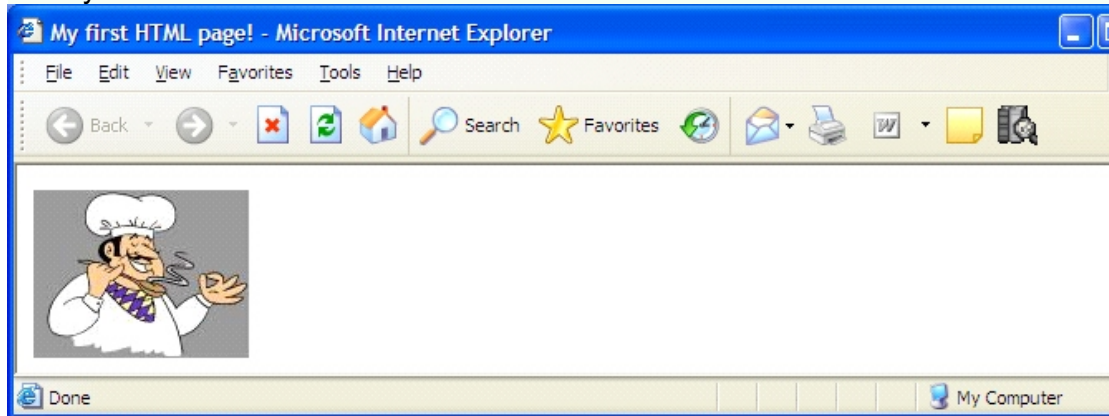
 м?н alt attribute-тай байж болно.

```

```

alt бол хэрэглэгч ямар нэг шалтгааны улмаас зураг ???лдэгг?й browser хэрэглэж байх ?ед зургийг орлуулах хувилбар болно гэсэн ?г.

```
<body>  
  
</body>
```

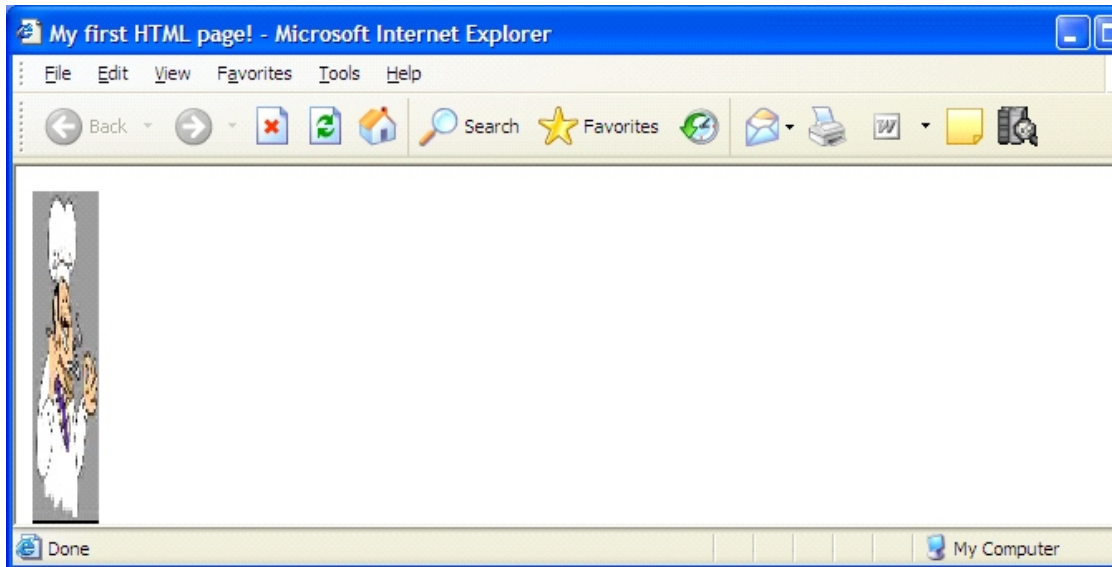


Бидний харж байгаачлан, browser зургийн хэмжээг ???р?? тодорхойлдог. ?нэхээр тийм юм бол хэмжээг заавал онцолж ?зэх хэрэг байна уу. З?вх?н ??ний улмаас browser зурганд зориулан санах ой н??ц?лс?н байдаг. Харин ??ний дараа хуудасны ?лдсэн хэсгийг ачаалдаг. Хуудас б?хэлдээ ачаалагдаж дууссаны дараа browser ахин зургандсзориулсан зайг д??ргэдэг. Харин хэмжээс ?г?гд??г?й бол зурганд х?рэх ?ед browser хуудсыг ачаалахаа зогсоож, зургаа ачаалдаг, дараа нь хуудсаа ?ргэлжл??лэн ачаалдаг.

```
<body>  
  
</body>
```



```
<body>  
  
</body>
```

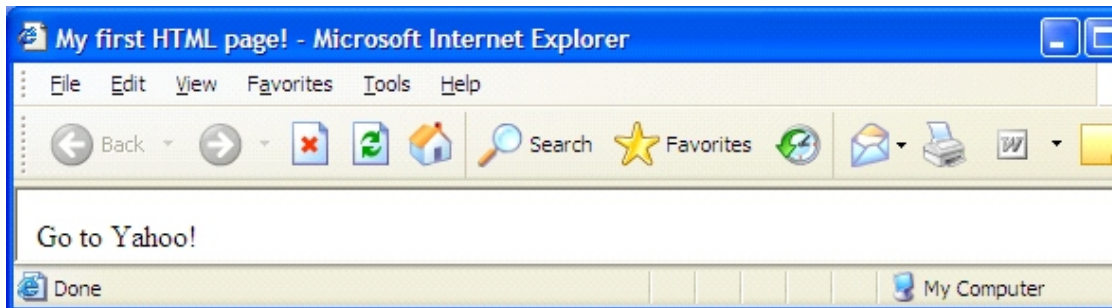


Бид ??рийн х?ссэн хэмжээгээ оруулж ?гч болохыг харлаа.

АРВАНХОЁРДУГААР ХИЧЭЭЛ

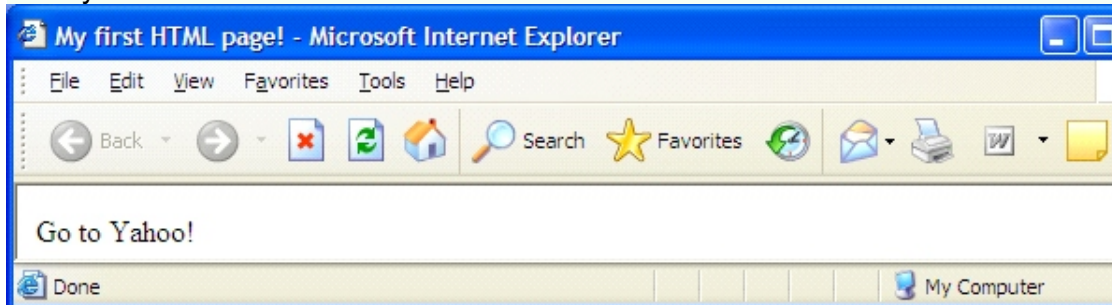
Одоо link хэрхэн хийх тухай авч ?зье. Эхлээд [Yahoo](#).-д link хийе.

```
<body>  
Go to Yahoo!  
</body>
```



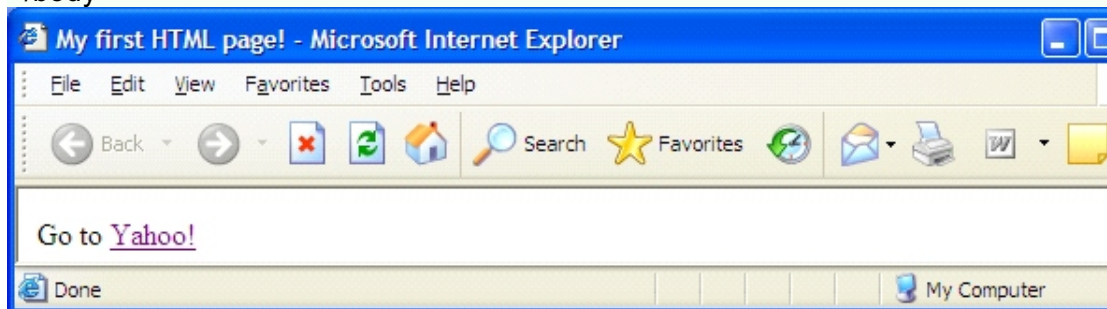
Одоо ??р нэг tag нэмье.

```
<body>  
Go to <a>Yahoo!</a>  
</body>
```



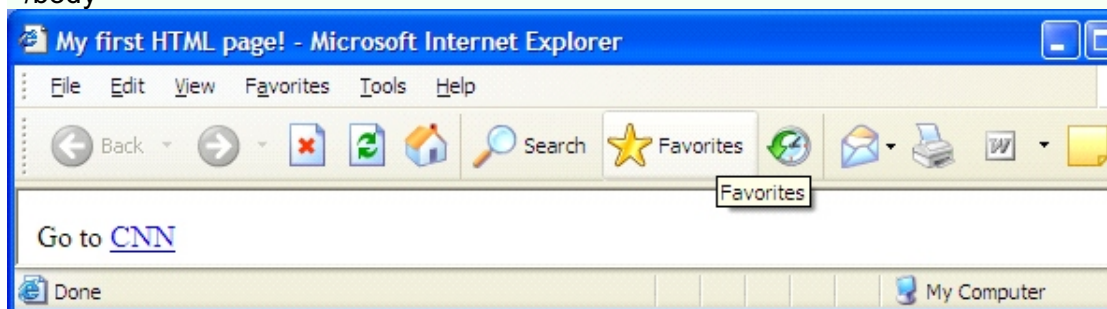
Одоо URL нэмээд боллоо. URL нь Uniform Resource Locator-ыг товчилсон ?г б?г??д энэ нь з?гээр л INTERNET хаяг. Харин бид одоо **href="url"** нэмж ?гн?. Yahoo-н URL **http://www.yahoo.com/** учраас бидний хийх link доорх хэлбэртэй болно.

```
<body>  
Go to <a href="http://www.yahoo.com/">Yahoo!</a>  
</body>
```



Ахиад нэгийг хийгээд ?ье.

```
<body>  
Go to <a href="http://www.cnn.com/">CNN</a>  
</body>
```



PHP гэж юу вэ?

PHP is a server-side, cross-platform, HTML embedded scripting language.
PHP бол сэрвер талын, платформ хамааралгүй, HTML холилдсон, вэб хэл.

Дээрх тодорхойлолтыг тайлбарлья.

1. Сэрвер талын -- Бүх PHP кодууд сэрвер дээр байрлах бөгөөд сэрвер дээр өгөгдөл гүйцэтгэгдэнэ.
2. Платформ хамааралгүй -- PHP кодыг Unix, Windows NT, Macintosh болон бусад үйлдлийн системтэй сэрвер дээр ажиллуулж болно.
3. HTML холилдсон -- PHP кодыг HTML кодын дотор хольж болно.
4. Вэб хэл -- Вэб хэл нь програмын хэлээс ялгаатай. Вэбийн хэлээр бичсэн програм нь браузерын туслалцаатайгаар ажилладаг бол програмын хэлээр бичсэн програм өөрөө биеэ дааж ажилладаг.

Тэгэхээр PHP бусад вэб хэлээс юугаараа давуутай вэ? Яагаад бид нар цагаа гарздаж энэ хэлийн тухай уншиж байна? Вэб барихад танд байгаа бусад сонголтуудаас дурдвал HTML, CGI, ASP, JSP ба CFML.

1. HTML -- Hyper Text Markup Language
2. CGI -- Common Gateway Interface
3. ASP -- Active Server Pages
4. JSP -- Java Server Pages
5. CFML -- Cold Fusion Markup Language

Яагаад PHP? Яагаад гэхээр бусдаасаа хурдан, найдвартай ажилладаг, сурахад хялбар, олон үйлдэл гүйцэтгэдэг, ёрөнхийдөө "супер". Тэгэхээр бусад хэлний тухай тайлбарлахгүй бол хангалтгүй сонсогдож байна.

CGI код бичихийн тулд вэбмастер Perl, Python эсвэл C зэрэг хэл мэддэг байх шаардлагатай. Дээрх хэлүүд бол вэб хэл биш, програмын хэл. Програмын хэл сурахад цаг, хөдөлмөр, мөнгө их гарздана. Өрдөө л вэб програм бичихийн тулд бүтэн програмын хэл сурах бол илүү үйлдэл. Хэрвээ вэбээс програмчлал руу шилжих зорилготой байгаа нөхцөлд бол ондоо хэрэг.

HTML хэл бол вэбийг бүтээх хэл болохоос ямар нэг үйлдэл гүйцэтгэх чадваргүй. Иймээс зөвхөн STATIC хуудас хийхэд хэрэглэнэ.

ASP код бичихийн тулд вэбмастер Visual Basic програмын хэлийн VB Script хэл эзэмшсэн байх шаардлагатай.

JSP бичихэд JAVA програмын хэл мэддэг байх хэрэгтэй

CFML Cold Fusion бол PHP -д дөхөж ирнэ. Сурахад хялбар, HTML холилдсон вэб хэл.

Тэгэхээр зарим хүн JavaScript-н талаар бодож байж магадгүй. JavaScript хэл бол сэрвер талын бус, хэрэглэгчийн талын вэб хэл. Иймээс бидэнд хэрэгтэй гэж бодохгүй байна. Яагаад гэхээр бид нар гол нь сэрвер талын вэб програм бичих талаар хойшдоо хэлэлцэх болно.

PHP -ын жинхэнэ ид шид нь датабэйстэй холбогдож дата буюу мэдээлэлтэй харьцах. Бусад сэрвер талын вэб хэлвүдээс олон терлийн датабэйстэй холбогдож чаддагаараа онцлог.

Доор хэдхэнийг дурдвал:

MySQL

Oracle

Access

SQL

PostgreSQL

mSQL

SyBASE

...

Бусад жишээнээс дурдвал ASP хуудсууд Microsoft Access-тэй харьцдаг ба 2-уулангийг нь ард Microsoft байж байхад PHP бараг бүх терлийн датабэйс програмтай холбогдож чаддагаараа давуу.

Лекц 1. PHP БҮЛЭГ

Right Environment

PHP Apache мэтийн Web server-тэй хослон ажилладаг. Web server PHP script-ын хэсэлт хэлээн аваад түүнийг PHP interpreter хэрвүүлдэг. Ажиллуулсны дараах гарах үр дүнд Web server-т очих ба server энэ үр дүнд client browser руу дамжуулна. PHP script буюу – бидний script янзбүрийн тооцоо хийж, хэрэглэгчийн үүргдийг боловсруулж, үүргдийн сантай ажиллаж, файл уншиж, хуулах зэрэг олон үйлдэлүүцэтгэж байдаг. Үүрүүр хэлбэл, бусад программчлалын хэлний хийж чаддаг бүх зүйлсийг бид түүрийн PHP script дээрээ хийж чадна.

Бид PHP хэрэглэхийн тулд тохирох development environment-ыг шаарддаг. Бид PHP 5-ыг татан авч суулгаж болно

Эхлэе

Бид цаашаа явахаас үүдэн нэг маш чухал зүйлийг ойлгох хэрэгтэй. HTML код гаргахын тулд заавал код бичихийг шаарддаг CGI script-ээс ялгаатай нь PHP бидэнд PHP кодыг жирийн HTML хуудсанд оруулан нэгтгэж, энэ нэгтгэгдсэн PHP кодоо web хуудасыг request-лэсэн үед боловсруулан хэрэглэгч рүү илгээдэг.

Энэ нэгтгэгдсэн PHP командуудыг тусгай нээх, хаах tag дотор хийдэг. Жишээ:

```
<?php  
... PHP code ...  
?>
```

Доор PHP болон HTML-ыг хэрхэн нэгтгэн хэрэглэж болохыг үзүүлээ.

```
<html>  
<head></head>  
<body>
```

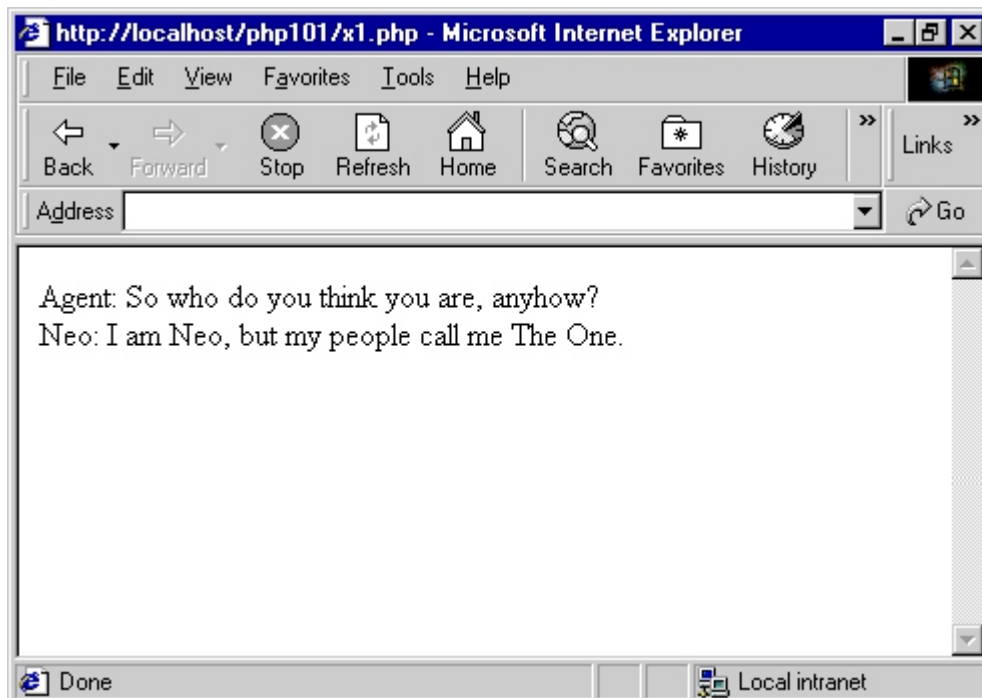
Agent: So who do you think you are, anyhow?

```
<br />
```

```
<?php  
// print output  
echo 'Neo: I am Neo, but my people call me The One.';  
?>
```

```
</body>  
</html>
```

Энэ script-ыг үүрийн Web server-ын document root-дээ .php үргэтгэлтэйгээр хадгал. Тэгээд үүнийгээ Browser дээр ачаал. Үр дүн нь дараах байдалтай харагдах ёстой.



HTML source:

```
<html>
<head></head>
<body>
```

Agent: So who do you think you are, anyhow?

```
<br />
```

Neo: I am Neo, but my people call me The One.

```
</body>
```

```
</html>
```

Бид дээрх script-ийг request хийх үед Apache энэ хэсэлтийг хүлээн авч бидний хэсэлт PHP рүү илгээгддэг. Дараа нь PHP script-ыг задлан уншиж, <?php...?> хоорондох кодыг гүйцэтгэн, кодын гаралтанд үр дүнг үгдэг. Энэ үр дүн server-т буцан ирж, хэрэглэгч рүү илгээгдэнэ. Гаралт зөв HTML учраас browser үүнийг хэрэглэгчид үзүүлэх боломжтой болно.

Энэ script-ийг анхааралтай харвал PHP-ын синтаксын үндсэн зарчмуудыг олж харна. PHP statement бөр semi-colon-р тгсдэг. PHP шинээр суралцагчдын их гаргадаг алдаа бол энэ semi-colon-ыг орхидог явдал юм. Гэхдээ PHP block-ын сүүлийн мөрийг дуусгахад semi-colon хэрэггүй. Энэ үргийг PHP closing tag гүйцэтгэдэг. Доорх жишээг авч үзье.

```
<?php
```

```
// print output
```

```
echo 'Neo: I am Neo, but my people call me The One.'
```

```
?>
```

Мөн бусад программчлалын хэлний адилаар бид кодон дээрээ comment нэмж болно. PHP дээр нэг мөрийн болон олон мөрийн гэсэн хоёр төрлийн comment хийж болдог.

```
<?php
```

```
// энэ бол нэг мөрийн comment
```

```
/* энэ бол
```

?>

PHP tag доторх хоосон м?рн??дийг комплятор х?рв??лдэгг?й. М?н tag-ийн гаднах б?х з?йлс х?рв??лэгдэхг?й б?г??д тэр хэвээрээ буцаагддаг. Эндээс з?вх?н tag дотор бичигдсэн кодууд х?рв??лэгдэж, ажилладаг гэж хэлж болно.

Хувьсагчийн тухай

Хувьсагчид бол б?х программчлалын хэлэн дээр хамгийн гол з?йл байдаг. PHP дээр ч м?н адил. Хувьсагчийг бид тоон болон тоон биш ?г?гдлийг аль алийг нь хадгалж болох з?йл гэж ойлгож болно. Энэ хувьсагчийн утга програм ажиллах явцад ??рчл?гд?ж болдог. М?н хувьсагчдийг хооронд нь харьцуулж бас энэ харьцуулалтын ?р д?нг хэрэглэн ямар нэг ??рэг г?йцэтгэх боломжтой.

PHP-ийн ?г?гдлийн т?рл??д: integer, floating point number, string болон массив. Ихэнх программчлалын хэлэн дээр хувьсагчдийн т?рлийг хэрэглэхээсээ ?мн? зарлаж ?г?х ёстой байдаг бол PHP дээр байдал арай ??р байдаг. PHP хувьсагчийн т?рлийг хадгалж буй ?г?гд?л болон агуулгаас нь хамааруулаад автоматаар тодорхойлдог.

Хувьсагч б?р нэртэй байдаг. Харин PHP дээр хувьсагчийн нэрний ?мн? (\$) тэмдэгт байдаг ба ?сэг, доогуур зураасаар эхэлж, тоо, ?сэг, болон доогуур зураас араас нь байж болдог. Жишээ нь, \$popeye, \$one болон \$INCOME. М?н PHP хувьсагч case sensitive. Жижиг жишээ авч ?ье.

```
<html>
<head></head>
<body>
```

```
Agent: So who do you think you are, anyhow?
<br />
```

```
<?php
// define variables
$name = 'Neo';
$rank = 'Anomaly';
$serialNumber = 1;

// print output
echo "Neo: I am <b>$name</b>, the <b>$rank</b>. You can call me by my serial
number, <b>$serialNumber</b>.";
```

?>

</body>

</html>

Хувьсагчид утга онооход = тэмдэгийг хэрэглэнэ. Жишээ нь:

<?php

\$age = \$dob + 15;

?>

Бид нэг зэрэг хэд хэдэн хувьсагчид утга оноож болно. Нэг зэрэг гурван хувьсагчид утга оноосон доорх жишээг авч үзье.

<?php

\$angle1 = \$angle2 = \$angle3 = 60;

?>

PHP-ын гэгдлийн тэрлээдээс үргүн хэрэглэгддэгийг нь дор жишээгээр үзүүлээ. **Boolean:** үнэн, худал гэсэн хоёр утга авдаг.

<?php

\$auth = true;

?>

Integer: Бүхэл тоо.

<?php

\$age = 99;

?>

Floating-point: энэ нь ерөнхийдүү бутархай тоо.

<?php

?>

String: тэмдэгт??дийн дараалал. "hello" эсвэл "abracadabra". String утгуудыг (""), ("")-ын алинд ч хийж болдог. Давхар хашилтан доторх String-ийг автоматаар задлан шалгаж, тусгай тэмдэгт буюу хувьсагчийн нэр байгаа эсэхийг нь шалгадаг. Хэрвээ энэ шалгалт батлагдвал тохирох утгаар сольдог. Жишээ:

<?php

```
$identity = 'James Bond';
$scar = 'BMW';
// this would contain the string "James Bond drives a BMW"
$sentence = "$identity drives a $scar";
echo $sentence;
```

?>

Дэлгэрэнг?й мэдээллийг

<http://www.php.net/manual/en/language.types.php>.

Хувьсагчдийг хэрвээ барилгын блок гэж ?звэл операторууд энэ блокуудыг нэгтгэн барьдаг з?йл нь юм. PHP-д арифметик ?йлдл??д, string дээр хийгдэх ?йлдл??д, харьцуулалт, логик ?йлдл??д байдаг.

```
<html>
<head>
</head>
<body>
```

<?php

```
// set quantity
$quantity = 1000;
// set original and current unit price
$origPrice = 100;
$currPrice = 25;
// calculate difference in price
$diffPrice = $currPrice - $origPrice;
// calculate percentage change in price
$diffPricePercent = (($currPrice - $origPrice) * 100)/$origPrice
?>
```

```

<table border="1" cellpadding="5" cellspacing="0">
<tr>
<td>Quantity</td>
<td>Cost price</td>
<td>Current price</td>
<td>Absolute change in price</td>
<td>Percent change in price</td>
</tr>
<tr>
<td><?php echo $quantity ?></td>
<td><?php echo $origPrice ?></td>
<td><?php echo $currPrice ?></td>
<td><?php echo $diffPrice ?></td>
<td><?php echo $diffPricePercent ?>%</td>
</tr>
</table>

</body>
</html>

```

М?н арифметик ?йлдлийг, утга оноох ?йлдэлтэй цуг г?йцэтгэж болно.

```
<?php
```

```
// this...
```

```
$a = 5;
```

```
$a = $a + 10;
```

```
// ... is the same as this
```

```
$a = 5;
```

```
$a += 10;
```

```
?>
```

Харин string-??дийг нэмэхдээ (.)-ийг хэрэглэнэ.

```
<?php
```

```
// set up some string variables
```

```
$a = 'the';
```

```
$b = 'games';
```

```

$c = 'begin';
$d = 'now';

// combine them using the concatenation operator
// this returns 'the games begin now<br />'
$statement = $a.' '.$b.' '.$c.' '.$d.'<br />';
print $statement;

// and this returns 'begin the games now!'
$command = $c.' '.$a.' '.$b.' '.$d.'!';
print $command;

```

```
?>
```

Энд ч м?н адил утга оноох ?йлдлийг нэмэх ?йлдэлтэй цуг хийж болно.

```
<?php
```

```

// define string
$str = 'the';

// add and assign
$str .= 'n';

// str now contains "then"
echo $str;

?>

```

Формын тухай

? мн? бид PHP-ын ?ндсэн ойлголтуудыг авсан бол одоо гол давуу тал болох хэрэглэгчийн оролтыг автоматаар WEB-р авч, ??нийг PHP хувьсагчид хувиргах талаар авч ?знэ.

Form...

Форм бол WEB-тэй харилцах хамгийн энгийн, т?ргэн арга м?н. Формын тусламжтайгаар хэрэглэгчдээс саналыг нь авах, сан б?рд??лэх зэрэг олон з?йл хийх боломжтой болдог. PHP Web-based формоос ??сэх ?г?гдлийг боловсруулах процессыг хялбарчилдаг. Доорх жишээ нь хоёр скриптээс б?рддэг. Нэг нь

HTML форм агуулж байгаа бол (form.htm) н?г?? нь формыг боловсруулах логикийг агуулсан байгаа. (message.php). form.htm-ын код:

```
<html>
<head></head>
<body>
<form action="message.php" method="post">
Enter your message: <input type="text" name="msg"
size="30">
<input type="submit" value="Send">
</form>
</body>
</html>
```

? ?ний хамгийн чухал хэсэг нь <form> tag

```
<form action="message.php" method="post">
...
</form>
```

<form> tag –ын "action" attribute нь формд оруулсан ?г?гдлийг боловсруулах server-side script (message.php)-ыг зааж ?гч байна. "method" attribute мэдээллийг хэрхэн дамжуулахыг зааж байгаа.

...Функц

Одоо ?мн?х жишээний хоёр дах хэсэг message.php script-ыг авч ?зье. Энэ script хэрэглэгчийн оруулсан ?г?гдлийг уншиж цаашдын ?йлдлийг г?йцэтгэдэг.

message.php:

```
<html>
<head></head>
<body>

<?php
// retrieve form data
$input = $_POST['msg'];
```

```
$u=$input*3;
// use it
echo "You said: <i>$u</i>";
?>
```

```
</body>
</html>
```

form.htm-ын оролтонд ямар нэг текст бичээд ("Boo"), submit хийвэл формыг боловсруулах message.php-ийг уншиж, бичсэн зүйлийг харуулна. ("You said: Boo"). PHP script-д форм орсон бол энэ форм доторх хувьсагчууд автоматаар script-д хэрэглэгдэхэд бэлтгэгддэг. PHP-ийг PHP container хувьсагч: \$_POST-г илгээхэд гэдэг. \$_POST container дотор бид формын хувьсагчийн нэрийг хэрэглэн утгыг нь авч болно.

PHP-д мөн формын утгыг авахад GET method хэрэглэж болдог. Энэ үед бидний үргэлж ёстой зүйлс бол үүгээр л "method" attribute-ийг "get", болгоод үүгэд л авахдаа \$_POST-ын оронд \$_GET-ийг хэрэглэхэд болно.

Одоо бидний үндэслэл шалгах буюу давталтын оператор үзэх цаг. PHP-ийг мөн харьцуулах үйлдлийг эхэлж үзье.

```
<?php
/* define some variables */
$mean = 9;
$median = 10;
$mode = 9;

// less-than operator
// returns true if left side is less than right
// returns true here
$result = ($mean < $median);
print "result is $result<br />";
```

```
// greater-than operator
// returns true if left side is greater than right
// returns false here
$result = ($mean > $median);
print "result is $result<br />";

// less-than-or-equal-to operator
// returns true if left side is less than or equal to
right
// returns false here
$result = ($median <= $mode);
print "result is $result<br />";

// greater-than-or-equal-to operator
// returns true if left side is greater than or equal
to right
// returns true here
$result = ($median >= $mode);
print "result is $result<br />";

// equality operator
// returns true if left side is equal to right
// returns true here
$result = ($mean == $mode);
print "result is $result<br />";

// not-equal-to operator
// returns true if left side is not equal to right
// returns false here
$result = ($mean != $mode);
print "result is $result<br />";

// inequality operator
// returns true if left side is not equal to right
// returns false here
$result = ($mean <> $mode);
print "result is $result";
```

```
?>
```

Харьцуулах ?йлдлийн ?р д?н з?вх?н хоёр байдаг. ? нэн (1) эсвэл худал (0). PHP 4.0 дээр шинэ харьцуулах ?йлдэл гарч ирсэн. Энэ нь тэнц?? эсэхийг болон т?рлийг нь шалгадаг === оператор юм. ? ?нийг доорх жишээгээр ?з??лье.

```
<?php
```

```
/* define two variables */
$str = '10';
$int = 10;

/* returns true, since both variables contain the
same value */
$result = ($str == $int);
print "result is $result<br />";

/* returns false, since the variables are not of the
same type even though they have the same value */
$result = ($str === $int);
print "result is $result<br />";
/* returns true, since the variables are the same
type and value */
$anotherInt = 10;
$result = ($anotherInt === $int);
print "result is $result";
```

```
?>
```

PHP-д м?н логик AND, логик OR, логик XOR, логик NOT гэсэн ?йлдл??д байдаг. Доорх жишээгээр тайлбарлая.

```
<?php
```

```
/* define some variables */
$auth = 1;
$status = 1;
```

```

$role = 4;

/* logical AND returns true if all conditions are
true */
// returns true
$result = (($auth == 1) && ($status != 0));
print "result is $result<br />";

/* logical OR returns true if any condition is true
*/
// returns true
$result = (($status == 1) || ($role <= 2));
print "result is $result<br />";

/* logical NOT returns true if the condition is false
and vice-versa */
// returns false
$result = !($status == 1);
print "result is $result<br />";

/* logical XOR returns true if either of two
conditions are true, or returns false if both
conditions are true */
// returns false
$result = (($status == 1) xor ($auth == 1));
print "result is $result<br />";

?>

```

? мн? дурьдсанчлан, н?хц?л шалгах оператор ?г?гдс?н н?хц?л ?нэн эсэхийг шалгаж, ??ний ?р д?нгээс хамааруулан ??р ??р ?йлдэл хийдэг.

```

if (condition) {
    do this!
}

```

за одоо if() statement-ыг формд яаж хэрэглэх талаар авч ?ье. Доорх жишээнд хэрэглэгчээс насаа оруулахыг х?снэ.

```
<html>
<head></head>
<body>
<form action="ageist.php" method="post">
Enter your age: <input name="age" size="2">
</form>
</body>
</html>
```

Нас нь 21-ээс их байгаа эсэхээс хамаараад ageist.php script ??p ??p агуулга б?хий текстийг харуулна.

```
<html>
<head></head>
<body>

<?php
// retrieve form data
$age = $_POST['age'];
// check entered value and branch
if ($age >= 21) {
    echo 'Come on in, we have alcohol and music
awaiting you!';
}
if ($age < 21) {
    echo "You're too young for this club, come back
when you're a little older";
}
?>

</body>
</html>
```

if () statement-ээс гадна PHP-д if-else-ийг хэрэглэж болно.

Бичигдэх хэлбэр

```
if (condition) {  
    do this!  
}  
else {  
    do this!  
}
```

Жишээ:

```
<html>  
<head></head>  
<body>  
  
<?php  
// retrieve form data  
$age = $_POST['age'];  
// check entered value and branch  
if ($age >= 21) {  
    echo 'Come on in, we have alcohol and music  
awaiting you!';  
}  
else {  
    echo "You're too young for this club, come back  
when you're a little older";  
}  
?>  
</body>  
</html>
```

Жишээ 2.

```
<?php  
  
if ($numTries > 10) {  
    $msg = 'Blocking your account...';  
}  
else {  
    $msg = 'Welcome!';  
}
```

```
?>
```

Жишээ 3.

```
<?php
```

```
if ($day == 'Thursday') {  
    if ($time == '0800') {  
        if ($country == 'UK') {  
            $meal = 'bacon and eggs';  
        }  
    }  
}
```

```
?>
```

? нийг арай хялбараар бичвэл:

```
<?php
```

```
if ($day == 'Thursday' && $time == '0800' && $country  
== 'UK') {  
    $meal = 'bacon and eggs';  
}
```

```
?>
```

```
<html>
```

```
<head></head>
```

```
<body>
```

```
<h2>Today's Special</h2>
```

```
<p>
```

```
<form method="get" action="cooking.php">
```

```
<select name="day">
```

```
<option value="1">Monday/Wednesday
```

```
<option value="2">Tuesday/Thursday
```

```
<option value="3">Friday/Sunday
```

```
<option value="4">Saturday
```



```
</select>
<input type="submit" value="Send">
</form>
</body>
</html>
```

Дээрх код зөвхөн долоо хоногийн өдөр сонгох боломж олгоно. Харин цаадах ажлыгнь PHP script `cooking.php` хийдэг.

```
<html>
<head></head>
<body>
<?php
// get form selection
$day = $_GET['day'];
// check value and select appropriate item
if ($day == 1) {
    $special = 'Chicken in oyster sauce';
}
elseif ($day == 2) {
    $special = 'French onion soup';
}
elseif ($day == 3) {
    $special = 'Pork chops with mashed potatoes and
green salad';
}
else {
    $special = 'Fish and chips';
}
?>
<h2>Today's special is:</h2>
<?php echo $special; ?>
</body>
</html>
```

Switch & case

PHP-ийн `if-else()`-ийн адил `??рэг г?йцэтгэдэг нэг statement` байдаг нь `switch-case()` `statement` юм.

```
switch (сонголтын -хувьсагч) {
    case эхний н?хц?л биелвэл:
        энэ ?йлдлийг г?йцэтгэ!
    case дараагийн н?хц?л биелвэл:
        энэ ?йлдлийг г?йцэтгэ!
    ... ГЭХ МЭГ...
}
```

Сонголтын хувьсагчийн утгаас хамаарч тохирох `case()` `block` г?йцэтгэгдэнэ. `default` `block` байж болох ба энэ нь сонголтын хувьсагчийн утга байгаа `case()` н?хц?лтэй тохирохг?й ?ед ажилладаг. ??нийг ил?? ойлгомжтой болгохын тулд дараах жишээг авч ?ье.

```
<html>
<head></head>
<body>
<?php
$day = $_GET['day'];
switch ($day) {
    case 1:
        $special = 'Chicken in oyster sauce';
        break;
    case 2:
        $special = 'French onion soup';
        break;
    case 3:
        $special = 'Pork chops with mashed potatoes
and green salad';
        break;
    default:
        $special = 'Fish and chips';
        break; } ?>
<h2>Today's special is:</h2>
<?php echo $special ?>
```

```
</body>
</html>
```

Энд хэд хэдэн анхаарах зүйл байдаг.

- `break` тгх??р ?г. Энэ нь `switch()` statement block-оос удирдлагыг дараагийн м?р р?? шилж??лэхэд хэрэглэгддэг.
- `default` тгх??р ?гийг бид `switch()` р?? дамжуулсан хувьсагч блокийн н?хц?лтэй тохирохг?й ?ед хэрэглэнэ. ??р??р хэлбэл н?хц?л??дийн аль нь ч биелэхг?й ?ед `default`-ын блок ажиллана гэсэн ?г.

`case()` block б?рийн ард `break` байх ёстой. Хэрвээ ??нийг мартвал PHP б?х `case()` block-ыг ажиллуулдаг.

? мн? хийсэн дасгал дээр бид HTML формоо нэг файл дээр, харин энэ формыг боловсруулах ?йлдлийг тусдаа ??р PHP script-ээр хийж байсан. Харин одоо н?хц?лт давталтыг ашиглаад бид эдгээрийгээ нэгтгэнэ.

? ?ний тулд хийх ёстой зүйл гэвэл `form submit control`-доо нэг оноож, энэ нэрээ формоос ?г?гд?л авах `$_POST container`-ийн хувьсагчаар тохируулж ?г?х хэрэгтэй. ? ?ний дараа бид формоос ирэх ?г?гдлийг боловсруулах боломжтой болно. Одоо жишээ авч ?зье.

```
<html>
<head></head>
<body>
<?php
/* if the "submit" variable does not exist, the form has not
been submitted - display initial page */
if (!isset($_POST['submit'])) {
?>

    <form action="<?php echo $_SERVER['PHP_SELF'];
?>" method="post">
    Enter your age: <input name="age" size="2">
    <input type="submit" name="submit" value="Go">
    </form>
```

```

<?php
}
else {
/* if the "submit" variable exists, the form has been
submitted - look for and process form data */
// display result
$age = $_POST['age'];
if ($age >= 21) {
echo 'Come on in, we have alcohol and music
awaiting you!';
}
else {
echo 'You are too young for this club, come
back when you are a little older';
}
}
?>
</body>
</html>

```

Дээр харсанчлан энэ script хоёр page-ээс б?рдэж байна. Нэг нь хоосон формын хэсэг б?г??д н?г?? нь submit button дарсны дараа ?р д?нг харуулж байна. Аль хуудсыг харуулахаа шийдэхийн тулд script эхлээд \$_POST['submit'] хувьсагч байгаа эсэхийг шалгана. Хэрвээ ??нийг олохг?й бол формыг submit хийгээг?й байна гэж ?зээд эхний хуудсыг харуулна. Харин форм submit хийгдсэний дараа энэ script дуудагдан оролтоо боловсруулдаг.

Бидний submit button "name" attribute-д ?г?гдс?н нэртэй байх. Харин \$_SERVER массив нь тусгай PHP хувьсагч б?г??д ??нд одоо ажиллаж байгаа скриптын нэр, зам зэргийн server мэдээлэл агуулагдаж байдаг.

Цикл

Цикл бол б?лэг ?йлдлийг олон дахин хийх шаардлага гарсан ?ед хэрэглэгддэг control structure юм. (давталтын тоог зааж ?гч болно, м?н н?хц?л хэрхэн биелж байгаагаас хамаарч ??рчл?гд?н?)

Одоо нэгэнт цикл ?зэж байгаа учир бид PHP-ийн автоматаар нэмэх болон хасах операторуудыг ?зэх шаардлагатай. Автоматаар нэмэх оператор бол хувьсагчийн

утгыг 1-ээр нэмэгдүүлдэг PHP оператор юм. Үүнийг хоёр нэмэх тэмдэгтээр (++) дүрсэлдэг. Жишээ авч үзье.

```
<?php
$total = 10;
// increment it
$total++;
// $total is now 11
echo $total;
?>
```

`$total++` нь `$total = $total + 1` -тай утга нэг.

Мөн үүнтэй адилаар автоматаар хасах буюу хорогдуулагч оператор байдаг нь (--) гэж тэмдэглэгдэнэ.

```
<?php
// define $total as 10
$total = 10;
// decrement it
$total--;
// $total is now 9
echo $total;
?>
```

Эдгээр операторууд цикл давталтуудад давталтын тоолуурыг гүйлгэх зорилгоор их хэрэглэгддэг.

Хамгийн энгийн цикл бол `while()` юм. Бичигдэх хэлбэр:

```
while (нэгж биелж байвал) {
    үүнийг гүйцэтгэ!
}
```

Энд, тодорхойлогдсон нэгж биелж байвал системийн хаалтан дах PHP statement-үүд гүйцэтгэгдэнэ. Харин нэгж биелэхгүй болвол, цикл дуусч програмын удирдлага циклийн гаднах дараагийн statement-д шилждэг.

`while()` циклыг илэрхийлж чадах жишээг авч үзье.

```
<html>
<head></head>
<body>
<form action="squares.php" method="POST">
Print all the squares between 1 and <input
type="text" name="limit" size="4" maxlength="4">
<input type="submit" name="submit" value="Go">
</form>
</body>
</html>
```

Энэ нь хэрэглэгчээс ?г?гд?л буюу тоо авах энгийн форм б?г??д формд ?г?гд?л оруулвал PHP script энэ тоог авч 1-ээс эхлээд гараас ?г?гдс?н тоо х?ртэлх б?х тооны квадратыг хэвлэнэ.

```
<html>
<head></head>
<body>

<?php
// set variables from form input
$upperLimit = $_POST['limit'];
$lowerLimit = 1;
// keep printing squares until lower limit = upper
limit
while ($lowerLimit <= $upperLimit) {
    echo ($lowerLimit * $lowerLimit). '&nbsp;';
    $lowerLimit++;
}
// print end marker
echo 'END';
?>

</body>
</html>
```

Энэ script нь while () циклыг тоолуурын утга 1-ээс эхлэн \$lowerLimit болон \$upperLimit хоёрын утга тэнцэх х?ртэл ажиллуулдна.

`while()` цикл заагдсан н?хц?л биелж байх ?ед б?лэг ?йлдлийг г?йцэтгэдэг. Хэрвээ хамгийн эхний давталт дээр н?хц?л биелж байвал яах вэ? ?мн?х жишээн дээр, формд бид 0 гэж оруулвал `while()` цикл нэг удаа ч гэсэн ажиллахг?й. Тэгэхээр бид б?лэг ?йлдлийг ядаж нэг удаа ч гэсэн ажиллуулъя гэвэл `do-while()` циклыг хэрэглэж болох нь. Б?тэц:

```
г?йцэтгэ {
    ??нийг г?йцэтгэ!
} while (н?хц?л)
```

тэгэхээр `while()` болон `do-while()`-ын ялгааг гаргах жишээ авч ?зье.

```
<?php
$x = 100;
while ($x == 700) {
    echo "Running...";
    break; }
?>
```

Энэ тохиолдолд, бд хичнээн ч удаа энэ PHP script-ыг ажиллуулсан ямар ч ?р д?н гарахг?й. Учир нь `$x`-ын утга 700-тай тэнц?? биш. Харин дараах script-ыг ашиглавал:

```
<?php
$x = 100;
// do-while loop
do {
    echo "Running...";
    break;
} while ($x == 700);
?>
```

Бид нэг м?р гаралтанд харах болно. Тэгэхээр одоо ?мн?х PHP script-ыг ямар утга оруулсанаас ?л хамаараад ядаж нэг удаа ажилладаг болговол:

```
<html>
<head></head>
<body>
<?php
```

```

// set variables from form input
$upperLimit = $_POST['limit'];
$lowerLimit = 1;
// keep printing squares until lower limit = upper
limit
do {
    echo ($lowerLimit * $lowerLimit) . '&nbsp;';
    $lowerLimit++;
} while ($lowerLimit <= $upperLimit);
// print end marker
echo ' END';
?>
</body>
</html>

```

while () болон do-while () циклүүд нь биелж байх үед давтагдаад ажиллаж байх нь. Харин бэлгэ үйлдлийг тодорхой удаа гүйцэтгэхээр бол яах вэ? Энд бид for-ыг хэрэглэж болно. Харагдах байдал:

```

for (тоолуурын эхний утга; нхцл; сгийн утга) {
    гүйцэтгэ!
}

```

энд цикл давтагдахаас үүн нхцл шалгагддаг. Хэрвээ нхцл биелвэл цикл давтагдан, тоолуурын утга нэгээр нэмэгддэг. Харин нхцл биелэхгүй бол цикл дуусч, дараагийн мрэнд удирдлага шилждэг. Жишээ:

```

<html>
<head>
<basefont face="Arial">
</head>
<body>

<?php
// define the number
$number = 13;
// use a for loop to calculate tables for that number
for ($x = 1; $x <= 10; $x++) {

```



```

echo "$number x $x = " . ($number * $x) . "<br />";
}
?>
</body>
</html>

```

Энд хамгийн т?р??нд хийх з?йл бол ?рж??лэх х?рдэнд хэрэглэх тоо юм. Жишээн дээр 13-ийг хэрэглэсэн байгаа.

Дараагийн анхаарах з?йл бол, for () циклын тоолуурын хувьсагч юм. ? ?нийг \$x гээд эхний утгыг нь 1, 10-аас ил??г?й удаа давталт явах ёстой гэж?зсэн. Энэ дотор бид ?мн? ?зсэн автоматаар ?с?х операторыг хэрэглэж болно. Цикл дотор тоолуур маань ?рж??лэх х?рд ?з??лэхийн тулд ?г?гдс?н тоогоор ?рж??лэгдсэн. Харин ?р д?нг дэлгэцэн дээр хэвлэхийн тулд echo () -г хэрэглэсэн.

```

<html>
<head></head>
<body>
<form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
Enter number of rows <input name="rows" type="text"
size="4"> and columns <input name="columns"
type="text" size="4"> <input type="submit"
name="submit" value="Draw Table">
</form>

```

```

<?php
if (isset($_POST['submit'])) {
    echo "<table width = 90% border = '1' cellspacing
= '5' cellpadding = '0'>";
    // set variables from form input
    $rows = $_POST['rows'];
    $columns = $_POST['columns'];
    // loop to create rows
    for ($r = 1; $r <= $rows; $r++) {

```

```
        echo "<tr>";
        // loop to create columns
        for ($c = 1; $c <= $columns;$c++) {
            echo "<td>&nbsp;</td> ";
        } echo "</tr> ";
    }
    echo "</table> ";
}
```

```
?>
```

```
</body>
</html>
```

Дээрх жишээ нь х?снэгтийг динамик хэлбэрээр ??сгэнэ.

Массивын тухай

Одоог х?ртэл бид з?вх?н ганц утга б?хий хувьсагчдийг ?зсэн.

```
<?php
$i = 5;
?>
```

Массив бол олон утгыг нэг хувьсагчид хадгалж болдог комплекс хувьсагч юм.

```
<?php
// define an array
$pizzaToppings = array('onion', 'tomato', 'cheese',
'anchovies', 'ham', 'pepperoni');
print_r($pizzaToppings);

?>
```

Энд \$pizzaToppings массиван хувьсагч б?г??д дотроо 'onion', 'tomato', 'cheese', 'anchovies', 'ham', 'pepperoni' гэсэн утгуудыг агуулсан байна.

print_r() бол массивын доторхийг харах боломж олгодог функц юм. ??нийгхэвлэхээс ил?? debug хийхэд хэрэглэхэдтохиромжтой байдаг. Массивын элементэд хандахдаа индексээр нь дамжуулдаг бахамгийн эхний элементийн

индекс 0 байна. Тэгэхээр дээрх жишээний 'onion' гэсэн элементэд хандахын тулд бид \$pizzaToppings[0], харин 'anchovies'-т хандахдаа \$pizzaToppings[3] гэх шаардлагатай болох нь ээ. Эндээс харахад массивын хувьсагчийн нэр, араас нь д?рв?лжин хаалтанд индекс гэсэн зарчмаар элементэд хандахаар байна.

PHP дээр бид индексийн оронд т?лх??р хэрэглэж болдог. Энэ ?ед арай ??р т?рлийн массив ??сэх ба энд хэрэглэгдэх т?лх??р??д давтагдахг?й, массивын з?вх?н нэг элементэд хамаардаг.

```
<?php
```

```
// define an array
$fruits = array('red' => 'apple', 'yellow' =>
'banana', 'purple' => 'plum', 'green' => 'grape');
print_r($fruits);
```

```
?>
```

\$fruits бол д?рв?н т?лх??р-утга гэсэн хослолоос б?рдэх массивын хувьсагч юм. (=> тэмдэгт нь т?лх??р болон т??ний утгын хоорондын холбоог заадаг) 'banana' гэсэн утганд хандахын тулд бид \$fruits['yellow'] гэж тэмдэглэх хэрэгтэй. Харин 'grape' буюу усан ?зэм гэсэн утганд хандахын тулд \$fruits['green'] тэмдэглэгээг хийнэ.

Энэ т?рлийн массивыг "hash" буюу "харилцан хамааралтай массив" гэж нэрлэдэг.

Массив зарлах хамгийн энгийн арга бол array() функц хэрэглэх.

```
<?php
```

```
$pasta = array('spaghetti', 'penne', 'macaroni');
?>
```

Массивт утга оноохдоо элемент б?рийг индексээр нь заан утга оноож болно.

Жишээ:

```
<?php
```

```
$pasta[0] = 'spaghetti';  
$pasta[1] = 'penne';  
$pasta[2] = 'macaroni';
```

```
?>
```

Хэрэв тоо биш т?лх??рэн индекс хэрэглэхийг х?свэл:

```
<?php  
$menu['breakfast'] = 'bacon and eggs';  
$menu['lunch'] = 'roast beef';  
$menu['dinner'] = 'lasagna';  
?>
```

Элемент нэмэх зарчим бас адилхан. Жишээ нь бид \$pizzaToppings массивт 'green olives' гэсэн элемент нэмье гэвэл:

```
<?php  
$pizzaToppings[3] = 'green olives';  
?>
```

Харин массивын элементэд ??рчл?лт хийхийг х?свэл тухайн хувьсагчид шинэ утгаа шууд оноож болно. Бидний ?зэж буй жишээн дээр 'ham'-ийг 'chicken' болгох шаардлага гарлаа гэвэл доорх загварыг дагаж болно.

```
<?php
```

```
$pizzaToppings[4] = 'chicken';
```

```
?>
```

Т?лх??р ашиглаж байгаа ?ед ч энэ нь адилхан.

```
<?php
```

```
$menu['lunch'] = 'steak with mashed potatoes';
```

?>

Байгаа массивын сүүлд элемент нэмэхдээ бид `array_push()` функцийг хэрэглэнэ.

```
<?php
// define an array
$pasta = array('spaghetti', 'penne',
               'macaroni');
// add an element to the end
array_push($pasta, 'tagliatelle');
print_r($pasta);
?>
```

Энгийн эсрэгээр массивын хамгийн сүүлийн элементийг устгахдаа `array_pop()` функцийг хэрэглэж болно.

```
<?php
// define an array
$pasta = array('spaghetti', 'penne', 'macaroni');
// remove an element from the end
array_pop($pasta);
print_r($pasta);
?>
```

Харин массивын эхнээс элемент хасъя гэвэл `array_shift()` функцийг хэрэглэнэ.

```
<?php
// define an array
$pasta = array('spaghetti', 'penne', 'macaroni');
// take an element off the top
array_shift($pasta);
print_r($pasta);
```

```
?>
```

`array_unshift()` функц массивын эхэнд элемент нэмэхэд хэрэглэгдэнэ.

```
<?php
```

```
// define an array
$pasta = array('spaghetti', 'penne', 'macaroni');
// add an element to the beginning
array_unshift($pasta, 'tagliatelle');
print_r($pasta);
```

```
?>
```

`array_push()` болон `array_unshift()` функцуудыг харин харилцан хамааралтай массив дээр хэрэглэх боломжгүй учраас энэ тэргийн массивт нэмэхдээ `$arr[$key] = $value` хэлбэрийг хэрэглэж болно. `explode()` функц нь string-ийг хэрэглэгчийн тодорхойлсон хязгаарлагчийн дагуу жижиг хэсгүүдэд хуваан, энэ хэсгүүдэд массив хэлбэртэй буцаадаг.

```
<?php
```

```
// define CSV string
$str = 'red, blue, green, yellow';
// split into individual words
$colors = explode(',', ' ', $str);
print_r($colors);
```

```
?>
```

Эсрэг тэргийг нь хийхийн тулд `implode()` функцыг хэрэглэнэ. Энэ функц хэрэглэгчийн тодорхойлсон нэгтгэлийн дагуу массивын элементүүдийг нэгтгэн нэг string үүсгэдэг. Үүнийг жишээгээр харвал:

```
<?php
```

```
// define array
$colors = array ('red', 'blue', 'green', 'yellow');
```

```
// join into single string with 'and' returns
'red and blue and green and yellow'
$str = implode(' and ', $colors);
print $str;

?>
```

Хамгийн эцэст нь доорх хоёр жишээ массивыг `sort()` болон `rsort()` функцуудыг хэрэглэн хэрхэн цагаан толгойн дараалал, `?с?х`, буурахаар эрэмбэлж болохыг `?з??лнэ`.

```
<?php
$pasta = array('spaghetti', 'penne', 'macaroni');
// returns the array sorted alphabetically
sort($pasta);
print_r($pasta);
print "<br />";
// returns the array sorted alphabetically in
reverse
rsort($pasta);
print_r($pasta);
?>
```

Давталт

Массивт элемент оруулах талаар `?злээ`. Одоо хэрхэн массиваас элемент авах талаар `?зье`. Энэ бол маш хялбар б?г??д бидний хийх ёстой з?йл бол индекс хэрэглээд тухайн элементэд хандах явдал юм. Жишээ авч `?зье`.

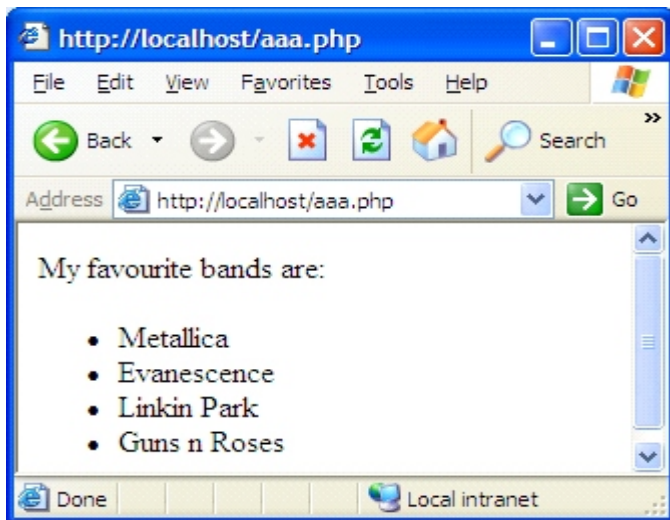
```
<html>
<head></head>
<body>
My favourite bands are:
<ul>

<?php
// define array
$artists = array('Metallica', 'Evanescence', 'Linkin
Park', 'Guns n Roses');
```

```
// loop over it and print array elements
for ($x = 0; $x < sizeof($artists); $x++) {
    echo '<li>'.$artists[$x];
}
?>

</ul>
</body>
</html>
```

Энэ script-ыг ажиллуулснаар бидэнд доорх цонх харагдах ёстой.



Дээрх тохиолдолд, эхлээд массивыг тодорхойлоод, нийгээ дэлгэцэнд харуулахын тулд `for()` давталт хэрэглэн, индексээр нь дуудаж хэвлэсэн байна. Энд шинээр `sizeof()` функцыг хэрэглэсэн байна. Энэ нь массивт хэрэглэгддэг функцуудын дотроос хамгийн чухал функцын нэг бөгөөд массивын хэмжээг буцаадаг. Ихэвчлэн циклыг массив дахь элементийн тоогоор гүйлгэх зорилгоор хэрэглэнэ.

Харин хамааралт массивыг хэрэглэж байгаа бол `array_keys()` болон `array_values()` функцуудыг ашиглаж болох юм.

```
<?php
```

```
// define an array
$menu = array('breakfast' => 'bacon and eggs',
    'lunch' => 'roast beef', 'dinner' => 'lasagna');
```



```
/* returns the array ('breakfast', 'lunch', 'dinner')
with numeric indices */
$result = array_keys($menu);
print_r($result);
print "<br />";
```

```
/* returns the array ('bacon and eggs', 'roast beef',
'lasagna') with numeric indices */
$result = array_values($menu);
print_r($result);
```

?>

Массивын б?х элементийг задлан харуулах бас нэг арга байдаг нь `foreach()` цикл б?г??д энэ нь массиваар давталт явуулдаг.

```
foreach ($array as $temp) {
    do this!
}
```

`foreach()` цикл ??рт нь аргумент хэлбэрээр дамжуулагдсан массивын элемент б?рийн хувьд давтагддаг. `for()`-оос ялгаатай тал нь тоолуур буюу `sizeof()`-ын алийг ч хэрэглэдэгг?й. Учир нь энэ функц массивын байрлалыг автоматаар байнга тогтоож байдаг. Давталт явагдах б?рд, системийн хаалтан дахь `statement` г?йцэтгэгдэх ба тухайн ?ед сонгогдсон массивын элемент т?р зуурын циклийн хувьсагч болдог.

??нийг хэрхэн ажиллахыг ил?? сайн ойлгохын тулд ?мн?х жишээг `foreach()` цикл хэрэглэн ахин бичье.

```
<html>
<head></head>
<body>
My favourite bands are:
<ul>
<?php
```

```

// define array
$artists = array('Metallica', 'Evanescence', 'Linkin
Park', 'Guns n Roses');
// loop over it
// print array elements
foreach ($artists as $a) {
    echo '<li>'.$a;
}

?>
</ul>
</body>
</html>

```

Цикл давтагдах үед, одоогоор сонгогдсон байгаа массивын элементийг тэр зуурын хувьсагч \$a-д оноодог. Энэ хувьсагчийг циклын блок доторх statement-үүд хэрэглэдэг. foreach() цикл тоолуур хэрэглэдэггүй учраас үйлчилгээ бага шаардагддаг, стандарт for() циклыг бодвол уншихад хялбар байдаг. Мөн хамааралт массив дээр ч ямар нэг нэмэлт программ шаардалгүйгээр хэрэглэж болно.

PHP-д форм боловсруулахад массив болон циклийг хэрэглэх нь илүү эргэлтэй. Жишээ нь, бэлэг checkbox болон олон сонголттой list боловсруулах гэж байгаа бол бүүх сонгогдсон формын утгыг нэг хувьсагчид хадгалахад массивыг хэрэглэвэл боловсруулалтыг хялбарчилна. Үүнийг тайлбарлах нэг жишээ авч үзье.

```

<html>
<head></head>
<body>

<?php
// check for submit
if (!isset($_POST['submit'])) {
    // and display form
    ?>
    <form action="<?php echo $_SERVER['PHP_SELF'];
?>" method="POST">
    <input type="checkbox" name="artist[]" value="Bon

```

```

Jovi">Bon Jovi
    <input type="checkbox" name="artist[]"
value="N'Sync">N'Sync
    <input type="checkbox" name="artist[]"
value="Boyzone">Boyzone
    <input type="checkbox" name="artist[]"
value="Britney Spears">Britney Spears
    <input type="checkbox" name="artist[]"
value="Jethro Tull">Jethro Tull
    <input type="checkbox" name="artist[]"
value="Crosby, Stills & Nash">Crosby, Stills & Nash
    <input type="submit" name="submit"
value="Select">
    </form>

```

```

<?php
    }
else {
    // or display the selected artists
    // use a foreach loop to read and display array
elements
    if (is_array($_POST['artist'])) {
        echo 'You selected: <br />';
        foreach ($_POST['artist'] as $a) {
            echo "<i>$a</i><br />";
        }
    }
    else {
        echo 'Nothing selected';
    }
}
?>
</body>
</html>

```

Дээрх формыг submit хийх ?ед, PHP автоматаар массивын хувьсагчийг ??сгэж, сонгох цэстэйгээ холбоно. ??ний дараа массивыг foreach() циклээр боловсруулж, сонгогдсон элементийг эндээс татан болно. Энэ ?йлдлийг multi-select list дээр хэрэглэж болно.

Лекц №3

Энэ хичээлээр бид файльтай хэрхэн ажиллах, ?-н дотор файл олох, файлын агуулгыг унших, м-н файлд ?-гд-л нэмэх тухай авч ?-нэ. Энэ б-хнийг PHP-ийн файльтай ажилладаг, файлын атрибютитийг ?-рчилд-г, агуулгыг нь уншдаг API-ийн тусламжтайгаар хийнэ.

Эхний ээлжинд файл нээж, агуулгыг нь уншиж ?-ье. ?-рийн дискэн дээрх аль нэг файлыг авч ?-зэж болно. Бидэнд Испани омлетийн жор байгаа ба ?-нийгээ PHP скрипт харуулахыг х-сч байна. ?-нийг хийхэд дараах гурван алхамын дагуу явах хэрэгтэй.

- ❖ Файлаа нээж, ?-ндээ file handle оноох.
- ❖ Файльтай handle дамжуулан ажиллахын тулд ?-нийхээ агуулгыг PHP хувьсагчид оноох.
- ❖ Файлыг хаах.

Энэ ?-йлдлийг хийх PHP script-ийг доор ?-з-?ллээ.

```
<?php
```

```
// set file to read
$file =
'/usr/local/stuff/that/should/be/elsewhere/recipes/omelette.txt' or die('Could not open file!');
// open file
$fh = fopen($file, 'r') or die('Could not open file!');
// read file contents
$data = fread($fh, filesize($file)) or die('Could not read file!');
// close file
fclose($fh);
// print file contents
echo $data;
```

```
?>
```

Энэ скриптийг Web browser дээрээ дуудан ажиллуулвал, PHP файлын агуулгыг буцаах ёстой. Одоо ?мн?х гурван алхамаа дэлгэрэнг?й тайлбарлая.

Файл нээж, file handle оноох

PHP-д файлаас ?г?гд?л уншихын тулд file handle хэрэгтэй болдог. Энэ file handle-ийг файлын нэр, зам болон файлыг нээх горимыг тодорхойлдог *string* гэсэн хоёр аргумент б?хий `fopen()` функцээр г?йцэтгэж болно. `fopen()` функцыг дараах горимуудад хэрэглэж болно. ? ?нд:

'r' – файлыг унших горимонд нээнэ

'w' – файлыг бичих горимонд нээх ба файлын ?мн?х агуулгыг устгана

'a' – файлыг нэмэх горимонд нээх ба ?мн?х ?г?гдлийг хадгалдаг.

Файлтай ажиллаж, ?г?гдлийг нь PHP хувьсагчид оноох

Хэрвээ `fopen()` функц ажиллавал, файлтай цаашид ажиллахад хэрэглэгдэх `$fh` file handle-ыг буцаадаг. Энэ file handle-ийг `fread()` функц хэрэглэн, файлыг уншиж, ?г?гдлийг нь хувьсагчид оноодог. `fread()`-ын хоёр дахь аргумент бол уншигдах байтын тоо юм. ? ?нийг бид `filesize()` функцын тусламжтайгаар тодорхойлж болно.

Файлыг хаах

Файлыг хаах ?йлдэл заавал байх шаардлагаг?й учир нь код ажиллаад дуусахдаа автоматаар файлыг хаадаг. Гэвч файлыг `fclose()` ашиглан хаах нь тодорхой давуу талуудтай.

Бид ?мн? нь `die()` функцыг хэрэглэж байгааг?й. Энэ функц нь ихэвчлэн алдааг зохицуулах зорилгоор хэрэглэгддэг. Жишээ нь, файлын зам худлаа заагдсан буюу файлын PHP унших эрхг?й тохиолдолд `die()` скрипт боловсруулалтыг дуусгаж, хэрэглэгчийн сонгосон алдааны мэдээллийг ?з??лдэг.

```
<?php
```

```
// set file to read
```

```

$file =
'/usr/local/stuff/that/should/be/elsewhere/recipes/omelette.txt' or die('Could not read file!');
// read file into array
$data = file($file) or die('Could not read file!');
// loop through array and print each line
foreach ($data as $line) {
    echo $line;
}

?>

```

Дээрх жишээн дээр, `file()` команд файлыг нээж, массив хэлбэрээр уншин, файлыг нэг ?йлдлээр хаадаг. Файлын м?р б?р нь массивын нэг элементэд хамаарна. Ийм болохоор одоо файлын агуулгыг хэвлэхэд `foreach()` циклыг хэрэглээд бараг болно. Хэрвээ файлыг боловсруулахад массив хэрэглэхг?й гэвэл `file_get_contents()` функцыг хэрэглэж болно. Энэ функц нь файлыг б?хэлд нь нэг string-ээр ?з?лдэг.

```

<?php
// set file to read
$file =
'/usr/local/stuff/that/should/be/elsewhere/recipe/omelette.txt' ;
// read file into string
$data = file_get_contents($file) or die('Could not read file!');
// print contents
echo $data;
?>

```

PHP-д м?н скрипт р?? файл импортлох маш хялбар хоёр функц байдаг нь `include()` болон `require()` функцууд юм. `include()` болон `require()` функцуудын хэрэглээг ойлгохын тулд жишээ авч ?зье. Web site б?р дээр хуудасны дээд хэсэгт стандарт цэс, доод хэсэгт стандарт copyright мэдээлэл байдаг. Энэ хуудас б?р дээр дээрх х?л, толгойн мэдээллийг оруулах код бичихийн оронд PHP х?гж??лэгчид х?л, толгойн мэдээлэл б?хий файлуудыг тусд нь ??сгээд хуудаснуудын дээд, болон доод хэсэгт з?гээр импортлодог. Ингэснээр хуудасны зохиомжийг хэрэгж??лэхэд хялбар болно. Учир нь хуудас

б?р дээр дахин дахин засвар хийхийн оронд з?вх?н хоёрхон файл дээр жижиг ??рчл?лт хийгээд болоо. ? ?нийгээ жишээгээр авч ?зье. Толгойн мэдээлэл б?хий файл ??сгээд header.php гэж хадгалъя.

```
<html>
<head>
<title><?php echo $page['title'];?></title>
</head>
<body>
<!-- top men` u bar -->
<table width="90%" border="0" cellspacing="5"
cellpadding="5">
<tr>
<td><a href="#">Home</a></td>
<td><a href="#">Site Map</a></td>
<td><a href="#">Search</a></td>
<td><a href="#">Help</a></td>
</tr>
</table>
<!-- header ends -->
```

Одоо copyright мэдээлэл б?хий х?лний мэдээллээр хоёр дах файлыг ??сгэе. Тэгээд footer.php гэж нэрлэе.

```
<!-- footer begins -->
<br />
<center>Your usage of this site is subject to its
published <a href="tac.html">terms and
conditions</a>. Data is copyright Big Company Inc,
1995-<?php echo date("Y", mktime ()); ?></center>
</body>
</html>
```

Харин одоо бидний вэб хуудасны ?ндсэн агуулгыг м?н х?л, толгойн мэдээллийг тохиромжтой газар нь гаргах скриптээ бичье.

```
<?php
```

```

// create an array to set page-level variables
$page = array();
$page['title'] = 'Product Catalog';
/* once the file is imported, the variables set above
will become available to it */

// include the page header
include('header.php');

?>

<!-- HTML content here -->

<?php

// include the page footer
include('footer.php');

?>

```

Энэ скриптийг ажиллуулах үед, PHP автоматаар хэрхэл, толгойн мэдээлэл бүхий файлуудыг уншиж, HTML хэсэгтэй нэгтгэн бүхэл хуудсыг бидэнд харуулна.

Импортлогдож байгаа файл дотор PHP код хэрхэл бичиж болно. Файл анх уншигдах үед, parser `<?php...?>` tag байгаа эсэхийг тогтоогоод, байвал доторх кодыг нь автоматаар гүйцэтгэнэ.

PHP-д `require_once()`, `include_once()` функцууд байдаг нь үүнийг уншигдсан файлыг ахин уншихгүй байхад анхаарал тавьдаг. `require()` функцууд заагдсан файл олдохгүй бол алдааны мэдээлэл дамжуулж, скриптийн боловсруулалтыг зогсоодог бол `include()` функц анхааруулга дамжуулаад, скриптийн боловсруулалтыг үргэлжлүүлдэг.

За одоо файл руу бичих талаар авч үзье. Файл руу үргэлж бичих нь уншихтай тус тус. Файлыг нээж, file handle аваад, энэ file handle-ийг хэрэглэн үргэлжлүүлж файл руу бичнэ. Гэвч энд хоёр том ялгаа гарч ирдэг. Эхнийх нь, файлыг `fopen()` хэрэглэн нээхдээ бичих горимонд('w') нээх явдал бол хоёр дах

нь файлаас унших `fread()` функцын оронд бичих `fwrite()` функцыг хэрэглэх явдал юм.

```
<?php
```

```
// set file to write
$file = 'dump.txt';
// open file
$fh = fopen($file, 'w') or die('Could not open
file!');
// write to file
fwrite($fh, "Look, Ma, I wrote a file! ") or die('Could
not write to file');
// close file
fclose($fh);
```

```
?>
```