

**به نام خدا**

# **برنامه نویسی وب**

Web Programming

**ویژه دانشجویان**

تدوین : علی ثاقب موفق

## مقدمه و تاریخچه اینترنت ( World wide web )

تاریخچه مختصر اینترنت: اینترنت از دپارتمان دفاعی آمریکا شروع شد و ابتدا وزارت دفاع آمریکا آنرا تحت عنوان آرپانت در دهه ۱۹۶۰ با سرویس های اولیه پست الکترونیک و انتقال فایل ارائه داد و سپس این موضوع در دهه ۱۹۸۰ علاقه مندی تجاری و دانشگاهی یافت و نهایتا در سال ۱۹۸۹ سرویس اینترنتی WWW توسط آقای tim berners lee ایجاد شد. مرورگر وب netscape در سال ۱۹۹۴ و مرورگر ie در سال ۱۹۹۵ ارائه شد. بدنبال آن سایت Amazon.com در سال ۱۹۹۵ و گوگل در سال ۱۹۹۶ ژانویه شروع به کار کردند. اینترنت، حاصل اجتماع هدفمند رایانه های مستقل است به نحوی که هر رایانه یک اتصال به شبکه اینترنت از طریق ip ( internet protocol ) دارد. کامپیوترها به طور پویا به شبکه متصل شده و سپس شبکه را ترک می کنند و با توجه به وجود استانداردهای باز برای اتصال به شبکه های رایانه ای، همه می توانند جزئی از شبکه اینترنت بوده و یک سیستم اینترنتی جدید ایجاد کنند. همچنین با توجه به فقدان کنترل مرکزی برای اینترنت، هر کسی می تواند بسادگی از آن استفاده نماید.

**سازمانها و استانداردهای اینترنت:** برخی از استانداردهای موجود در این زمینه عبارتند از:

Tcp :Transmission Control Protocol

IETF : internet engineering task force

Icann : internet corporation for assigned name and number decided top level domain name

### استانداردهای وب :

W3c : world wide web consortium

**پروتکل اینترنتی ( IP ) :** پروتکل ساده برای ارسال داده بین کامپیوترها. هر دستگاه دارای یک آدرس ۳۲ بیتی است که از ۴ تا ۸ رقم مبنای ۲ تشکیل شده است. که معادل دهدهی آن مثل 145.10.34.3 می شود. برای بدست آوردن آدرس IP می توان از سایت Whatismyip.com و یا با استفاده از دستور ipconfig در خط فرمان ویندوز استفاده کرد.

**Tcp:** پروتکلی است که مالتی پلکسینگ را انجام می دهد و تحویل پیام بر روی IP را گارانتی می کند. در این پروتکل چند برنامه با استفاده از یک آدرس ip کار می کنند و در اینجا مولتی پلکسینگ ، اختصاص یک عدد داده شده به هر برنامه یا سرویس است که اختصارا پورت نامیده می شود. بعنوان مثال برای پورت Port 80

web browser و پورت ۴۴۳ برای secure browsing و پورت ۲۵ برای email است. بعضی از برنامه ها بجای tcp از پروتکل udp استفاده می کنند.

**سرور وب ( www )**: وب سرور، نرم افزاری است که برای درخواست های صفحه وب اینترنت، همواره گوش فرامی دهد. مثل Apache و IIS

**مرورگر وب یا Browser**: کار مرورگر این است که مستندات را از سرور وب گرفته و سپس نمایش می دهد. مثل: Google chrome , Opera , Apple safari , Ie , Mozilla firefox

**سرویس نام حرفی بجای آدرس IP ( domain name system - Dns )**: مجموعه ای از سرور ها است که نامهای نوشته شده را به آدرسهای عددی ip نگاشت می کنند. مثال [www.es.washintun.edu](http://www.es.washintun.edu) که معادل آدرس عددی و IP آن 128.208.3.88 است. بسیاری از سیستمها یک حافظه cache محلی دارند و تحت یک فایل میزبان به نام Host File که این آدرس ها را نگهداری می کنند.

Windows: c:\system32\drives\etc\hosts

Mac: /private/etc/hosts

Linux: /etc/hosts

**آدرس اینترنتی URL ( uniform resource locator )**: شناسه ای برای محل مستند روی یک وب سرویس می باشد. مثلا در آدرس URL مربوط به <http://www.aw-bc.com/info/regessepp> ، پروتکل آن Http و میزبان آن [www.aw-bc.com](http://www.aw-bc.com) و مسیر آن همان مسیر فولدري [/info/regessepp](http://www.aw-bc.com/info/regessepp) است. لذا بر اساس ورود یک url در مرورگر ، کارهای زیر به ترتیب انجام می شود:

- درخواست از سرور dns برای نام حرفی متناظر با آدرس IP [www.aw-bc.com](http://www.aw-bc.com)
- اتصال به آدرس ip مذکور در پورت ۸۰
- درخواست از سرور برای بدست آوردن فایل [/info/regessepp/index.html](http://www.aw-bc.com/info/regessepp/index.html) از مسیر مذکور
- نمایش صفحه نتیجه روی صفحه مرورگر

**ارجاع یا anchor**: با کمک آن، پرش به بخش داده شده از صفحه وب صورت می گیرد.

<http://www.testpad.com/download/index.html#download>

در اینجا ابتدا فایل index.html اخذ شده و سپس پرش به بخش صفحه با برچسب download صورت می گیرد.

**پورت port برای سرورهای وب :** پورت پیش فرض روی سرورهای وب ۸۰ است و برای پورت هایی غیر از پورت پیش فرض ۸۰ ، بایستی به شکل زیر عمل شود و شماره پورت عینا درج گردد:

`http://www.cs.washingtun.edu:8080/secret/money.txt`

**رشته پرس چو query string :** رشته پرس جو، شامل یک مجموعه پارامترهایی است که به برنامه وب فرستاده می شود و سپس برنامه وب بر اساس آن پارامتر ها عمل می کند. مثلا :

`http://www.google.com/search?q=miserable+failure and start = 10`

که در آن پارامتر q به miserable + failure ، مقداردهی و set می شود و همچنین پارامتر start به عدد ۱۰ مقداردهی و set می شود.

پروتکل http ( Hyper Text Transport Protocol ) : به مجموعه قوانین و دستوراتی گفته میشود که توسط سرور وب قابل تفسیر و اجراء است و این فرامین از یک مرور گر وب فرستاده می شود. برخی فرامین لازم را http مرورگر، به طور داخلی ارسال می نماید که عبارتند از:

Get : filename download

Post: filename : send a web from response

Put file name : upload filename

شبیه سازی یک مرورگر با یک پنجره ترمینال به ترتیب مراحل زیر است:

- telnet [www.cs.washingtun.edu](http://www.cs.washingtun.edu) 80
- trying 128.208.3.88
- connected to 128.208.3.88
- escape character is '^]
- get /index.html

**کدهای خطای http :** مرورگر با یک سند html همراه است . لذا وقتی چیزی غلط بشود سرور وب یک کد خطا بر می گرداند ( "error code" ). برخی از کدهای خطای رایج عبارتند از:

خطای به شماره های 301 – 303 با پیغام page has moved یعنی صفحه جایجا شده است.

خطای شماره ۴۰۳ به صورت you are forbidden to access to this page یعنی دسترسی به این صفحه برای شما منع شده است و لذا دسترسی به صفحه وجود ندارد.

خطای شماره ۴۰۴ به شرح page not found یعنی اینکه صفحه مورد نظر یافت نشد.

خطای ۵۰۰ و پیام internal server error یعنی خطای سرور داخلی وجود دارد.

کد ۲۰۰ و پیام ok

انواع رسانه اینترنتی mime : بعضی اوقات وقتی وب شامل منابعی در صفحه است ( sheet , icon و یا شی چند رسانه ای ) ما نوع داده را مشخص می کنیم:

|                          |       |
|--------------------------|-------|
| Text/html                | .html |
| Text/plain               | .txt  |
| Image/gif                | .gif  |
| Image/jpeg               | pg    |
| Video/quichtime          | .mov  |
| Application/octet.stream | .exe  |

فناوری و زبان های وب نویسی :

Html برای نوشتن صفحات وب

Css اطلاعات سبک برای صفحات وب

Php ایجاد صفحات پویا روی سرور وب

Java script ایجاد صفحات وب قابل برنامه نویسی و محاوره ای

Ajax دسترسی داده برای برنامه های کاربردی وب

Xml متا زبان برای سازماندهی داده

Sql تعامل با بانک اطلاعاتی

صفحات وب CSS : meta data

اطلاعت درباره صفحه شما ارائه می نماید. ( برای یک مرورگر و موتور جستجو و .... )

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

```
<meta name="description" content="author" websitefor building javaPrograms "/>
```

< meta name = “ keywords” content = “java, textbook”/>

اینها در بخش head صفحه xhtml قرار می گیرند. تگ meta اغلب دارای نام و محتوی است. یعنی content , name ، بعضی از تگ های meta از صفت http-equiv بجای نام استفاده می کنند. با استفاده از یک تگ meta برای content-type ، هشدار “tentatively valid” را دریافت نمی کنیم.

لیست تعریف **dd, dt , dl** :

<dl > ft

<dt newbie </dt>      <dd> one who does ..... </dd>

<dt own </dt>      <dd> to soundly ..... </dd>

<dt frag .... </dt>      <dd> a kil ..... </dd>

</dl>

در اینجا DI ، یک لیست از تعاریف واژه ها – بلوک را ارائه می دهد ، Dt ، ارائه هر واژه و dd تعریف آن است.

### برنامه نویسی تحت وب ( گام به گام ) :

لغت HTML مخفف hyper text markup language است و کار آن شامل :

۱\_توصیف محتوی و ساختار اطلاعات یک صفحه وب و البته نه شبیه یک کار نمایشی و presentation

۲\_محتوی متن را با تگ های باز و بسته احاطه می کند.

۳\_هر نام تگ tag عنصر یا element نامیده می شود.

مثال :

<element >      **content**      </element>

</p>      یک پاراگراف      <p>

۴\_در تفسیر برنامه وب، همه جاهای خالی به یک جای خالی تبدیل می شوند و بقیه چشم پوشی می شوند.

۵\_از نسخه استاندارد xhtml استفاده می کنیم.

ساختار یک صفحه **xhtml** :

۱\_ header صفحه را توصیف می کند و body محتوی صفحه را

۲\_ صفحه html با پسوند .html ذخیره می شود.

```
<html>
  <head>
    اطلاعاتی درباره صفحه
  </head>
  <body>
    محتوی صفحه
  </body>
</html>
```

عنوان صفحه <title>:

```
</title>      عنوان      < title>
```

این دستور درون بخش head از برنامه وب قرار میگیرد و پس از اجراء در title bar مرورگر نمایش و در قسمت بالای صفحه مرورگر نمایش می یابد که در صورت باز بودن TAB های مرورگر مختلف متمایز باشد.

پاراگراف و ایجاد آن با استفاده از تگ <p>

```
<p>
    متن
    متن
</p>
```

تگ پاراگراف ، داخل بدنه و body صفحه قرار می گیرد.

عنوان یا heading <h1>, <h2>, .....<h6> :

عناوین برای جداسازی نواحی اصلی از صفحه و بلوک استفاده می شوند که h1 فونت بزرگتر است.

</h1> دانشگاه امیرکبیر <h1>

</h2> دپارتمان مهندسی کامپیوتر <h2>

</h3> رشته فناوری اطلاعات <h3>

این دستورات Tag داخل body می آید.

قانون خط افقی یا **horizontal rule** : برای رسم یک خط افقی برای تفکیک بخش های یک صفحه است.

</p> اولین پاراگراف <p>

<hr/>

</p> دومین پاراگراف <p>

بلوک و عناصر داخلی آن: بلوک یا Block شامل بخش کاملی از محتوی است که شامل لیست، پاراگراف و جدول است و مرورگر بین عناصر بلاک جای خالی قرار می دهد.

تگ های **html** و چیزهای بیشتری درباره آنها: بعضی tag ها می توانند شامل اطلاعات اضافی به نام صفات attribute باشند.

<a href = "page2.html" > next page </a>

<hr/> بعضی از تگ ها ( tag ) محتوی ندارند و در یک تگ باز و بسته می شوند . مثل :

اینها داخل بدنه صفحه وب و یا به عبارتی داخل تگ body قرار می گیرند.

پیوند ها و link به صفحات دیگر: استفاده از تگ <a>

<p>

Search

<a href = "http://www.google.com"> google </a>

</p>



از href برای مشخص کردن مقصد url استفاده می کنیم. که می تواند متعلق به سایتی دیگر و یا به طور نسبی تعریف شود. مثل صفحه دیگری روی این سایت باشد. پسوند ها و anchor ها عنصر و inline هستند که داخل بلوک قرار می گیرند . مثل h1 , p

**تصاویر <img>** : درج یک تصویر گرافیکی داخل صفحه

```
<img src = "images/gone/a,jpg" alt "picture" />
```

Src یک تصویر url را مشخص می کند. Xhtml همچنین نیاز به صفت alt دارد که در صورت نبودن تصویر آنرا نمایش دهد. که در اینجا در صورت نبودن تصویر یک قاب خالی با نوشته picture نمایش می یابد.

**رد کردن یک خط <br> line break** : الزام به رد کردن یک خط در وسط یک بلوک ( inline )

```
<p> teddy said it was a hat , <br/> so I put it on </p>
```

```
<p> now 'daddy': say in' , <br /> where are you </p>
```

نکته : br بایستی بلافاصله با /> بسته می شود. همچنین اینکه Br نباید برای چیداسازی پاراگراف استفاده شده و با ایجاد جای خالی چند بار در یک ردیف استفاده گردد.

**عناصر عبارتی یا phrase elements** : <em> , <strong>

تگ Em : معمولا برای تاکید جهت متن و کلمه ای استفاده می شود و آنرا به صورت ایتالیک می نویسد.

تگ Strong : متن را با تاکید بیشتر ظاهر می نماید و معمولا با حروف پررنگ و bold آنرا نشان می دهد

```
<p> Html is <em> really </em> <strong> really </strong> fun! </P>
```

**لیست نامرتب یا unordered list** : <ul> , <li>

تگ Ul مبین یک لیست از آیتم هایی با bullet است و ui آیتم های تکی مربوط به آن است.

```
<ul>
```

```
<li> simpsons:
```

```
<ul>
```

```
<li> homer </li>
```

```
<li> merge </li>
```

```
</ul>
  </li>
  <li> family;
<ul>
  <li> peter </li>
  <li> lois </li>
</ul>
</li>
</ul>
```

مثال دیگر:

```
<ul>
  <li> no shoes </li>
  <li> no shirt </li>
  <li> no problem </li>
</ul>
```

**لیست مرتب: <ol> ordered list**

نشان دهنده آیتم های لیستی شماره گذاری شده است:

```
<p> business model </p>
```

```
<ol>
  <li> customers </li>
  <li> consumers </li>
  <li> owners </li>
</ol>
```

کوئیشن : quotations : block quote

کو تیشن طولانی :

```

<blockquote>
<p>
    Is a example For qoutation
    Statement
</p>
</blockquote>

```

کو تیشن در خط **<q> : inline quotations**

```

<p> sample inline quotation <q> Qoute </q> </p>

```

اساس و پایه **CSS ( Basic css )** :

گروه بندی سبک ها **:grouping styles**

```

<html>
<head>
<style>
P, h1 {
Color : green ;
}
H2 {
    Blockground-color : yellow;
}
<style>
/* .....*/
/* .....this is comment .....*/
P {

```

```
Color : red;
Background-color = a gun;
}
```

خواص css برای font :

Font-family, Font-size, Font-style ,Font-weight, Text-align, Text-decoration

روش‌های تولید styles:

<p>

```
<font face = "arial"> welcome to iran. </font>
You will win <b> in life </b> , <i> ever </i> , <u> ever </u>
<font size = "+4" color = "red"> our </font>
```

**CSS: <link>**

<head>

```
<link href = "filename" type = "text/css" rel = "stylesheet"/>
```

</head>

CSS ظاهر و بستر اطلاعات صفحه وب را توصیف می‌کند که در صفحه html می‌آید و یا اینکه در یک فایل CSS جدا قرار می‌گیرد.

نحو و syntax در قانون کلی کد نویسی CSS :

```
Selector {
    Property : value;
    Property : value;
    Property : value;
    .....
    Property : value;
}
```

```
P {
  Font-family : sans-serif
  Color = red;
}
```

یک فایل css شامل یک یا چند rules است. هر rule با یک selector شروع می شود که عناصر html را مشخص می کند و سپس سبک properties را برای آنها بکار می برد. Selector ، \* همه عناصر را انتخاب می کند.

```
P {
  Color : red;
  Background-color : yellow;
}
```

```
H2 { color : rgb ( 128,0,196 ); }
```

```
P {
  Font-family : Garamond, "times new roman" , serif;
}
```

یک بخش کوتاه از کد کامپیوتر آورده شده است. (معمولا در فونت با طول ثابت ارجاع دارد)

```
<p>
  The <code> u1 </code> and <code> ol </code> .
</p>
```

متن از قبل فرمت شده <pre> :

```
<pre>
  Steve jobs speaks loudly
</pre>
```

سبک های گروه بندی یا **grouping styles** :

```
P, h1, h2 {
    Color : green;
}
H2 { background-color : yellow ; }
```

توضیحات CSS :

```
/* comments */
/* this
    is multi line comments */
P { color:red
    Background-color : aqua;
}
```

Text-decoration

```
P {
    Text-decoration : underline;
}
Body {
    Font-size : 16px;
}
```

اعتبار دهنده CSS :

```
<p>
    <a href = "http ://www.google.com"
    <img src = "abc.jpg"
    Alt = "valid css" /> </a>
</p>
```

```
Body {
    Background-image:url ( "images/draft.jpg" );
}
```

Php نمایش عناصر بلوک بعنوان inline :

```
<ul id = "topmenu">
    <li> item1 </li>
    <li> item2 </li>
    <li> item3 </li>
</ul>
```

```
#top menu li { Display : inline; Border : 2px solid gray; Margin-right : 1em; }
```

خروجی

```
Item1      item2      item3
```

خاصیت رویت پذیری ( **visibility property** ) :

این خاصیت تنظیم می کند که آیا یک عنصر باید روی صفحه نمایش یابد و می تواند قابل رویت یا مخفی باشد.

```
p.secret {
    visibility : hidden;
}
<?php>
    Print "hello world";
?>
```

بعضی url ها , برنامه هایی را که وب سرور باید آنها را اجرا کند را مشخص می کنند و سپس خروجی آنها برای شما ارسال می کند .

<http://webster.cs.washington.edu/quote2.php>

url بالا به سرور Webster.cs.washington.edu می گوید تا برنامه quote.php را اجرا و نتایج را برای مرورگر برگرداند. صفحات server – side برنامه های نوشته شده ای هستند که با استفاده از یکی از زبانها و framework های برنامه نویسی وب نوشته شده اند:

مثال perl , python asp.net ruby on rails , java/jsp , php

وب سرور شامل نرم افزاری است که اجازه می دهد که آن برنامه ها اجرا شوند و خروجی آنها بعنوان پاسخ به درخواست های وب برگرداند. ما از php برای برنامه نویسی server-side استفاده می کنیم و php یک زبان scripting لبه سرور است که برای ایجاد صفحات وب دینامیک استفاده می شود و محتویات مختلف وابسته به زمینه را فراهم می کند. با سایر سرویس ها ، بانک اطلاعاتی و email و غیره اینترفیس دارد . تعیین هویت کاربران و پردازش اطلاعات انجام می شود. کد php در کد xhtml می تواند قرار گیرد. وقتی که مرورگر درخواست یک فایل html. ( محتوی ثابت ) می دهد، سرور صرفا آن فایل را می فرستد. در حالیکه موقعیکه مرورگر درخواست یک فایل php. ( محتوی پویا ) می دهد، سرور آنها می خواند و هر کد و برنامه script داخل آن را اجرا می کند. گزینه هایی زیادی برای زبان های بخش server-side وجود دارد. مثل asp.net , ruby on rails , Jsp و غیره.

### مزایا برنامه نویسی php :

۱\_ رایگان بودن، free and open source و اینکه هرکسی می تواند یک سرور php-embeded را به طور مجانی اجرا کند.

۲\_ سازگاری : توسط بسیاری از وب سرورها پشتیبانی می شود.

۳\_ سادگی: تعداد زیادی از عملکردهای داخلی و نحو آشنایی دارد.

Lots of built-in functionality ; familiar syntax

۴\_ دسترس پذیری : روی سرورهای uw و بسیاری از Webster های تجاری نصب شده است

( dante, Webster ) بعنوان مثال محتوی زیر می توانند در یک فایل hello.php باشد.

```
<?php
    Print    " hello world";
?>
```

یک فایل با کد php عموماً با <?php شروع و با >? خاتمه می یابد. جملات و توابع تعیین نوع و غیره در بین این ها قرار می گیرد. نمی توان صفحه php. را روی هارد درایو رویت نمود و چیزی از کد منبع php



را نمی توان دید. اگر فایل را به وب سرور Upload کنید. php-enable درخواست فایل php. برنامه را اجرا خواهد کرد و به شما خروجی بر می گرداند.

## نحو اصلی php ( php basic syntax ) :

```
Print "hello world\n"
```

```
Print "escape\"chars\"are the same as in java\n";
```

## متغیرها variables :

```
$name = expression;
```

```
$username = "pinkheat"
```

```
$age = 16;
```

```
$drinking-age = $age + 5;
```

```
$this-class-rocks = TRUE;
```

نام ها، حساس به حروف بزرگ و کوچک هستند و لغت های چندگانه با - جدا می شوند. نام ها همیشه با \$ شروع می شود. در کاربرد و تعیین نوع همیشه به طور واضح توسط انتساب تعیین میشود. ( نوع نوشته نمی شود )

نوع های اساسی TYPE : int , float , Boolean , string , array, NULL , object

Php ، نوع ها را به طور اتوماتیک به هم تبدیل می کند. تبدیل اتوماتیک string به int , تبدیل int به

```
$age = (int) "21"; , type-cast with(type) , float
```

عملگرهای حسابی عبارتند از : += -= \*= /= %= + \* - / % ++ -- = += -= \*= /= %=

بسیاری از اپراتورها به طور اتوماتیک نوع ها را به یکدیگر تبدیل می کنند مثل 5 + "7" is 12

توضیح comment :

```
# single_line comment
```

```
// Single_line Comment
```

```
/*
```

## Multi\_line comments

\*/

شبيهه java اما # اجازه داده می شود. کدهای php از # بجای // برای توضیح استفاده می کند. مثال:

String type

```
$favorite_food = "Ethiopia";
```

```
Print $ favorite_foof[2];
```

```
5 + " 2 trutle dove" ==> 7
```

```
5 . "2 trutle dove" ===> 52 turtle dove
```

```
$age = 16
```

```
Print "you are $age year old.\n";
```

رشته داخل " " تفسیر و interpret می شود. و متغیرها ، مقدارشان درج می شود.

For loop ( same as java )

```
For ( initialization; condition; update) {
```

```
    Statements;
```

```
}
```

```
For ( $ i=0; $ i < 10; $ i++) {
```

```
    Print " $i squared is " , $i * $i . ".\n";
```

```
}
```

کدهای بین تگ <?php در ابتدا و تگ >? در انتها بعنوان کد php اجرا می شود. سایر محتویات بعنوان html خالص خروجی خواهند بود.

عملیات ریاضی :

```
$a = 3;
```

```
$b = 4;
```

```
$c = sqrt( pow($a , 2 ) + pow( $b, 2 );
```

نحو و syntax روش فراخوانی در php و پارامترها و return آن شبیه جاوا است.

نوع های float , int :

```
$a = 7/2;           #float : 3.5
$b = (int) $a;      # int : 3
$c = round($a );    # float : 4.0
$d = "123" ;        # string : "123"
$e = (int) $d;      # int : 123;
```

کلید واژه Int برای اعداد صحیح است و float برای حقیقی، عمل تقسیم بین مقادیر صحیح ممکن است مقداری ممیزی و float تولید کند.

نوع رشته ای:

```
$favorite_food = "Ethiopian";
Print $favorite-food[2];           #h
$favorite_food = $favorite_food, "cuisine";
Print $favorite_food;              # Ethiopian cuisine
```

ایندکس ها از صفر است . نوع char نداریم و هر حرفی یک string است و اتصال رشته ها با نقطه یا period است نه +

توابع رشته ای:

```
#index 0123456789012345
$name = "Stefani hatcher";
$length = strlen($name);           #16
$cmp = strcmp($name, "brian le");  # > 0
Index = strpos($name, "e" );       #2
First = substr($name , 9, 5 );      # hatch
Name = strtoupper ( $name);        #STETANTE HATCHER
```

تابع تبدیل به حروف کوچک یعنی `strtolower` هم داریم.

جملات `if/else`:

```
If ( condition)  {  
    Statements;  
} elseif ( condition)  {  
    Statements;  
} else  {  
    Statements;  
}
```

حلقه `while`: در اینجا کلمات کلیدی `break`, `continue` مشابه جاوا رفتار می کنند.

```
While ( condition )  {  
    Statements;  
}
```

```
Do {  
    Statements;  
} while ( condition);
```

NULL:

```
$name = "Victoria";
```

```
$name = NULL;
```

```
If ( isset($name))  {  
    Print "this line isn't going to be reached\n";  
}
```

مقداری NULL است که هیچ مقداری نداشته باشد یا اینکه NULL داشته و یا با تابع `unset` حذف شده باشد.

متغیر NULL با تابع `ISSET` تست می شود و برنامه بالا خروجی نخواهد داشت.

## آرایه ها : #create

```
$name = array();  
$name = array(value0, value1, ..... valueN);  
$name[index] #get element value  
Name[index] = value; #set element value  
Name[] = value; #append  
$a = array();  
$a[0] = 23;  
$a2 = array( "some" , "string" , "in" , "an" , "array");  
$a2[] = "ooh!"; # add string to end( at index 5)
```

برای الصاق از [] بدون مشخص کردن ایندکس استفاده می کنیم و نوع عنصر مشخص نمی شود.

## حلقه ForEach

```
Foreach ( $array as $variable )  
{  
    .....  
}  
$stooges = array ("larry", "moe", "curly", "shemp" );  
Foreach( $ stooges as $stooge) {  
    Print "moe slaps $stooge )  
    Print "moe slaps $ stooge\n"; #even himself  
}
```

این دستور معادل مجموعه دستورات زیر است:

```
For( $i=0; $i<count($stooges); $i++ )
```

```
{
    Print "moe slaps { [$stooges[@i]]\n"; }
```

چاپ تگ های html در php , یک سبک توصیه نشده:

```
<?php
    Print "<!doc....."\n";
    Print "<head> \n";
    .....
    For ( $i = 1; $i <=10; $i++) {
        Print "<p> I can count to $i <p> \n";
    }
?>
```

بلوک های php expression php : یک قطعه کوچک از php که یک مقدار عبارت را در html ارزیابی و سپس قرار می دهد .

```
< ? expression ? >
```

```
<h2> the answer is < ? = 6*7 ? > </h2>
```

```
<? = expression ?> <? Php print expression ?>
```

مثالی از بلوک expression :

```
<!doctype html public "-//wsc//dtd xhtml 1.1 //en"
```

```
http://www.w3.org/tr/xhtml11/dtd/xhtml11.ftf>
```

```
<html xmlns = http://www.w3.org/1999/xhtml>
```

```
<head> <title> cse 190 m : embedded php </title> </head>
```

```
<body>
```

```
<?php
```

```
For ( $i = 99; $i >=1; $i-- ) {
```

```

?>
<p> <? $i ?> bottles , <br/>
<? = $i ?> bottles , <br/>
Take one down , pass it around, <br/>
< ? = $i = 1 ?> bottles..... </p>
<?php
}
?>
<body>
</html>

```

### خطاهای رایج:

```

<body>
    <p> watch how....:
<?php
    For ( $i = 1; $i <= 10; $i++)    {
?>
<? $i ?>
</p>
</body>
</html>

```

در اینجا تگ های `</body>` , `</html>` داخل حلقه `for` هستند که هرگز بسته نمی شوند و اشتباه است. اگر فراموش شود که آکولاد بسته گردد, یک خطا از نوع `$end` غیر منتظره دریافت می شود. همچنین مقادیر پارامتر پیش فرض عبارتند از:

Function print\_separated ( \$str \$separator = “,” )

```
{
    If( strlen($str) > 0 )      {
    Print $str[0];
    For ( $i = 1; $i < strlen ( $str); $i++ )  {
        Print $separator, $str[$i];
    }
}
}
```

Print\_separated(“hello”); #h,e,l,l,o

Print\_separated( “hello , “-“); #h-e-l-l-o

اگر مقداری بعنوان پارامتر ارسال شود بعنوان پیش فرض استفاده می شود.

**محدوده متغیرها global , local :**

```
$school = “UW”
```

.....

```
Function downgrade() {
```

```
Global $school;
```

```
$syffic = “Tacoma”;
```

```
$school = “$school $ suffix”;
```

```
Print “$school\n”;
```

```
}
```

متغیر های تعیین نوع شده در داخل یک تابع برای آن تابع local هستند و متغیرهای خارج آن global هستند و اگر تابعی بخواهد از متغیر global استفاده نماید باید جمله global را داشته باشد.

رشته های پرس جو و پارامترها:



url ? name = value & name = value...

<http://www.google.com/search? Q = olive>

http:// example.com/student-login.php? username = stepp

&id = 123457

**Query string** : مجموعه پارامترهایی که از یک مرورگر به وب سرور بعنوان پارامتر ارسال می شوند و اغلب با قراردادن جفت نام/مقدار پاس می شوند. بعنوان مثال در پایان url در کد بالا پارامتر username دارای مقدار stepp و sid دارای مقدار ۱۲۳۴۵۶۷ است. کد php روی سرور می تواند مقدار پارامتر را تست و بکار گیرد و یک روشی برای کد php برای تولید خروجی مختلف بر اساس مقادیر پاس شده توسط کاربر است.

### پارامترهای Query : \$\_REQUEST

```
$username = $_REQUEST("username");
```

```
$id_number = (int) $_REQUEST["id"];
```

```
$seats_meat = FALSE;
```

```
If ( isset ( $_REQUEST["meat"] )) {
```

```
    $seats_meat = TRUE;
```

```
}
```

تابع \$\_REQUEST مقادیر پارامتر را بعنوان رشته بر می گرداند. تست می کند که آیا یک پارامتر داده شده با isset پاس شده است.

ورودی و خروجی php : توابع I/O ، PHP : فایل File-get-content("foo.txt")

```
"hello\n
```

```
How are\n
```

```
You\n
```

```
\n
```

```
I am fine\n"
```

فایل File( "foo.txt")

```

Array( "hello\n"    #0
      "how are\n"    #1
      "you?\n"      #2
      "\n" ,        #3
      " I am fine \n" #4
)

```

محتوی Foo.txt :

```

Hello
How are
You?
I am fine

```

فایل, خطوط یک فایل را بعنوان یک آرایه بر می گرداند. `file_get_contents` ، محتوی کامل یک فایل را بعنوان رشته بر می گرداند.

فایل کامل یک **reading/writing** :

```

# reverse a file

$text = file_get_contents( "poem.txt");
$text = strrev($text);
File-put_contents("poen.text" , $text);

```

`File_get_contents` محتویات کامل یک فایل را بعنوان رشته بر می گرداند و اگر فایلی نباشد پیغام هشدار دریافت می شود. `File_put_contents` یک رشته را داخل فایل می نویسد و هر محتوی قبلی را جایگزین می کند.

الصاق به فایل:

```

#add a line to a file

$new_text = "P.S[ly,gtg ttyl ?n";

```

```
File_put_contents ( "poem.txt", $new_text, FILE_APPEND );
```

File\_put\_contents بایک پارامتر سوم اختیاری فراخوانی میشود. ( Append(adds to the end بجای جایگزینی با محتوی قبلی.

تابع فایل :

#display lines of file as a bulleted list:

```
$lines = file("todolist.txt");
Foreach ( $line as $line )      {
?>
    <li>    <? = $line ?>    </li>
<?php
}
```

فایل file , خطوط یک فایل را بعنوان یک آرایه از رشته ها بر می گرداند. هر رشته با \n خاتمه می یابد. برای کوتاه کردن \n هر خط، از پارامتر دوم اختیاری استفاده کنید.

```
$lines = file ( "todolist.txt" , FILE_IGNORE_NEW_LINES );
```

مقدمات form :

بسیاری از صفحات جذاب وب با داده ها و data در ارتباط هستند مثل google , facebook , youtube که می توانند فرمت های مختلف داده را بگیرد. چند رسانه ای text , html , xml و بسیاری از آنها به ما امکان می دهند تا به داده های آنها دسترسی یابیم. حتی به ما اجازه می دهند تا داده جدید را submit کند. بسیاری از برنامه های وب سمت server-side پارامترها را می پذیرد و اجرای آنها را راهنمایی می کند .

فرم های form: html : یک گروهی از کنترل های ui که اطلاعات را از کاربر می گیرد و اطلاعات را به سرور وب ارسال می کند. اطلاعات به سرور بعنوان query\_string ارسال می شود. Javascript برای ایجاد کنترل محاوره ای ایجاد شده است .

```
<form action = "destination url ">
```

Form controls

</form>

در اینجا نیازمند صفت **action** است که از صفحه که این فرم را پردازش کند به **url** داده می شود. وقتی فرم پر شده و **submitted** می شود . داده به **url , action** ارسال می شود. یک صفحه ممکن است در صورت لزوم شامل فرم هایی باشد. مثال:

```
< form action = "http://www.google.com/search" . >
```

<div>

Lets search google:

```
<input name = "q"/>
```

```
<input type = "submit"/>
```

</div>

</form>

lets search google: ||| submit ||| : خروجی آن :

**<input>** کنترل های فرم

```
<!... 'q' happen to be name of google', requested prer...>
```

```
<input type = "text name = "q" value = "Colbert ropert"/>
```

```
<input type = "submit" value = "rooyah!"/>
```

Colbert report||| booya!||| : خروجی :

عنصر ورودی برای ایجاد بسیاری از کنترل های UI استفاده می شود.

**<input>** فیلدهای متن ورودی

```
<input type = "text size = "10" maxlength ="8"/> NETID <br/>
```

```
<input type ="password" sixe ="16" /> password
```

```
<input type = "submit" value = "log in ">/
```

||| NETID : خروجی :

||| password ||| log in

صفات ورودی شامل value , size , readonly , maxlength , disabled هستند.

Text boxes <textarea>

```
<textarea rows = "4" cols = "20">
```

Type your comments here,

```
</textarea>
```

**Checkboxes < input >** گزینه های yes/no که می تواند check , unchecked شود.

```
<input type = "checkbox" name = "lettuce" /> lettuce
```

```
<input type = "checkbox" name = "tomato" checked = "checked" /> tomato
```

```
<input type = "checkbox" name = "pickles" /> pickles
```

خروجی شامل سه گزینه lettuce , tomato , pickles خواهد بود. وقتی به سرور فرستاده می شود هر checked box مقادیری را ارسال می کند:

<http://webster.cs.washington.edu/params.php? Tomato = on & pickles = on>

استفاده از checked = "checked" در html برای تیک کردن box در ابتدای کار.

### Radio buttons < input>

```
<input type = "radio" name = "cc" value = "visa" checked = "checked" /> visa
```

```
<input type = "radio" name = "cc" value = "mastercard" /> mastercard
```

```
<input type = "radio" name = "cc" value = "amex" /> American express
```

خروجی شامل سه گزینه visa , mastercard , American express خواهد بود و بایستی یک مقداری را مشخص کند برای هر یک که فرستاده می شود.

**برچسب : text : < label >**

```
<label><input type = "radio" name = "cc" value = "visa" checked = "checked" />  
visa</label>
```

```
<label><input type = "radio" name = "cc" value = "master card"/> master card
</label>
```

```
<label> <INPUT TYPE = "RADIO" NAME = "CC" VALUE = "AMEX" /> American
express </label>
```

که سه گزینه خروجی خواهد داشت.

همراه با text کنترل به طوریکه بتوان click کرد text را برای فعال سازی کنترل با checkbox و radiobutton استفاده می شود. عنصر lable می تواند با قوانین سبک CSS هدف گذاری شود.

**<select> , <option> Drop-down list منوی انتخاب ها :**

```
<select name = "favorite character "
<option> jerry </option>
<option> George </option>
<option> selected = "selected"> Kramer </option>
<option> Elaine </option>
</select>
```

صفحه selected وقتی یکی انتخاب می شود set می گردد.

For list using <select>

```
<select name = "favoritecharacter[]" " size = '3" multiple = "multiple" >
    <option> jerry </option>
    <option> George </option>
    <option> Elaine </option>
    <option selected = "selected" newman </option>
</select>
```

خروجی:

Jerry

George

Elaine

||||| submit

صفت `multiple` اختیاری اجازه می دهد انتخاب آیتم های چند گانه را با `shift` یا `ctrl _ click` . نام اپراتور باید با `[]` همراه باشد اگر شما انتخاب های چند گانه دارید. `Tag` های انتخابی می تواند انتخاب شود تا در ابتدا به `selected` در ابتدا `set` شود.

Option group : `<optgroup>`

```
<select name = "favorite character">
```

```
  <optgroup lable = "major character">
```

```
    <option> jerry </option>
```

```
    <option> George </option>
```

```
    .....
```

```
</optgroup>
```

```
<optgroup> lable = "minor character">
```

```
  <option> newman </option>
```

```
  <option> susan</option>
```

```
</optgroup>
```

```
</select>
```

خروجی به صورت `Submit` `|||||` `jerry` `|||||||` خواهد بود .

آرایه های **"superglobal"** : آرایه های `superglobal` ( متغیرهای `global` ) شامل اطلاعاتی است درباره درخواست جاری سرور و غیره.

Array

پارامترها به درخواست `get , put` ارسال می شود : `$_GET , $_post`

پارامترها به هر نوع درخواست ارسال می شود : `$_REQUEST`

اطلاعات در باره وب سرور : `$_SERVICE, $_ENV`

فایل بارگزاری شده با درخواست وب : `$_FILES`

`$_session, $_COOKIE` : کوکی برای شناسایی کاربر استفاده شده است ( بعدا دیده می شود). اینها انواع مخصوص از آرایه هایی هستند که association array نامیده می شود.

```
<?php
```

```
    $base = $_REQUEST["base"];
```

```
    $exp  = $_REQUEST["exponent"];
```

```
    $result = pow ($base , $exp );
```

```
?>
```

```
<? = $base ?> ^ <? = $exp ?> = <? = $result?>
```

<http://example.com/example.php?base = 3 & exponent = 4>

خروجی آن می شود:  $3^4 = 81$

مثالی از **css html**:

```
<!doctype html>
```

```
<html>
```

```
    <head>
```

```
        <style>
```

```
            H1 {
```

```
                Color : blue;
```

```
                Font_family : verdana;
```

```
                Font_size : 300%;
```

```
            }
```

```
        P {
```

```
            Color:red;
```



```

    Font_family : courier;
    Font_size: 160%;
  }
</style>
</head>
<body>
  <h1>
    This is a heading </h1>
  <p> this is paragraph </p>
</body>
</html>

```

سبک خارجی مورد استفاده یا URL می تواند برای صفحه وب استفاده شود:

```

<link rel = "stylesheet" href = https://www.w3schools.com/html/styles.css>
<link rel = "stylesheet" href = "/html/styles.css">

```

مثال:

|                   |                                |
|-------------------|--------------------------------|
| <table>           | برای تعریف جدول                |
| <tr>              | برای تعریف ردیف جدول           |
| <td>              | برای داده جدول                 |
| <th>              | برای تعریف عنوان جدول          |
| <caption>         | برای تعریف caption :           |
| <border_collapse> | برای کولپس محدوده سلول:        |
| padding           | برای اضافه کردن پدینگ به سلول: |
| text_align        | تنظیم متن سلول:                |

```

<table style = "width:100%" >
  <tr>
    <th>      firstname  </th>
    <th>      lastname  </th>
    <th>      age        </th>
  </tr>
  <tr>
    <td> jill  </td>
    <td> smith </td>
    <td> 50 </td>
  </tr>
  <tr>
    <td> eve </td>
    <td> Jackson </td>
    <td> 94 </td>
  </tr>
</table>

```

.....

```

Table , th , td {
Border: 1px solid black;
}

```

عنصر <div> معمولاً به عنوان container برای سایر عناصر html استفاده می شود.

```

<div style = "background_color:black; color :white;padding:20px;">
<h2> London is the capital city of England.. </p>

```

</div>

I frame syntax < I frame src="url"> </iframe>

<iframe src = "demo-iftame.htm "height = "200" width = "300"> </iframe>

**Javascript** : صفحات وب را پویا تر و محاوره ای می سازد.

<!doctype html>

<html>

<body>

<h1> my first javascript </h1>

<button type = "button" on click =  
document.getelementByld('demo').innerHTML = Date()>

Click me to display Date and time, </button>

<p id = "demo"> </p>

</body>

</html>

تگ <script> برای تعریف یک اسکریپت قسمت client تعریف استفاده شده است.

مثال:

<script>

Document.getElementById("demo").innerHTML = "hello javascript!";

</script>

Document.getElementById("image").src = "picture.gif";

<!doctype HTML>

<html>

<body>

<script>

```

Function light(sw)      {
    Var pic = 0;
    If ( sw == 0 )      {
        Pic = "pic_bulloff.gif"
    } else              {
        Pic = "pic_bubon.gif";
    }
    Document.getElementById('myImage').src = pic;
}
</script>
<img id = "myimage" src = "pic_bulboff.gif" width = "100" height = "180">
<p>
<button type = "button" on click = "light(1)"> light on </BUTTON>
<button type = "button" on click = "light(0)"> light off </button>
</p>
</body>
</html>

```

HTML file path

|  |                                   |
|--|-----------------------------------|
| در همان فولدر صفحه اصلی قرار دارد          | picture <img src = "picture.jpg"> |
| در فولدر images قرار گرفته و در فولدر جاری | <img src = "images/picture.jpg">  |
| یک سطح بالاتر                              | <img src = "../picture.jpg">      |

مسیر مطلق:

مثال:

```
<img src = http://www.w3schools.com/images/picture.jpg alt = "mountain">
```

## طراحی وب responsive :

باعث می شود که صفحات وب شما روی همه دستگاه ها خوب به نظر آید . دسک تاپ ، تبلت ، تلفن و از CSS و html استفاده می کند.

```
<form>
```

```
First name : <br>
```

```
<input type = "text" name = "firstname"> <br>
```

```
Lastname : <br>
```

```
<input type = "text" name = "lastname">
```

```
</form>
```

radio button input

```
<form>
```

```
<input type "radio" name = "gender" value = "male" checked
```

```
<input type "radio" name = "gender" value = "female"> female <br>
```

```
<input type = "radio" name = "other" value = "other" <br>
```

```
</form>
```

**Client-side script** : بعد از اینکه صفحه از سرور فرستاده می شود، برنامه در مرورگر اجرا می شود. اغلب این کد ( برنامه ) تغییر و دستکاری می کند صفحه یا پاسخ به عمل کاربر را. کد php قبلا به ما اجازه صفحات پویا را می داد. چرا از اسکریپت نویسی قسمت client استفاده می کنیم؟ مزایای اسکریپت نویسی سمت client شامل قابلیت استفاده : اصلاح صفحه بدون داشتن به سمت سرور. کارایی: کوچک و سریع برای تغییرات صفحه بدون انتظار برای سرور

**Event-driven** : پاسخ به عمل کاربر مشابه کلیک و فشردن کلید.

**مزایای برنامه نویسی php سمت سرور :**

امنیت: دسترسی به داده خصوصی سرور و client نمی تواند که منبع را ببیند .

سازگاری و **Compatibility** : سازگاری کامل وجود دارد و موضوعی برای مباحث مرورگری نیست.

یک زبان برنامه نویسی سبک و برای صفحات وب محاوره ای استفاده می شود و دارای ویژگیهای زیر است:

-درج متن پویا در html

-واکنش به رویدادها

-بدست آوردن اطلاعات درباره کامپیوتر کاربر

-انجام محاسبات روی کامپیوتر کاربر

-استاندارد وب

-کد تفسیر و Interpret می شود و کامپایل Compile نمی شود javascript VS. java

- قوانین و نحو راحت و نوع داده کمتر ، متغیرها نیاز به تعیین نوع ندارند . خطاها اغلب ساکت هستند.

- ساختار کلیدی function است بجای class

**تشابهات زبان های javascript , php :** هر دو تفسیر interpret می شوند و کامپایل نمی شوند. هر دو نحو ساده ای دارند و هر دو حساس به حروف بزرگ و کوچک هستند.

**تفاوت ها:** js در واقع object oriented تر است و js روی اینترفیس کاربر تمرکز دارد. JS روی مرورگر client اجرا می شود و کد php روی سرور وب اجرا می شود.

پیوند به یک فایل javascript :

```
< script src = "filename" type = "text/javascript"> </script>
```

```
< script src = "example.js" type = "text/javascript"> </script>
```

تگ اسکریپت باید در head فایل html قرار می گیرد. کد اسکریپت در فایل جداگانه بعنوان js ذخیره می شود. کد JS می تواند مستقیما در body یا head فایل html قرار گیرد ( مثل CSS ). اما این یک سبک خوب نیست ( باید متوجه باشید که ارائه و رفتار جدا باشد )

```
Alert ("message");
```

```
Alert ("IE6 deleted suck mode enabled");
```

نوع متغیرها:

```
Var name = expression;
```

```
Var clientname = "connie client";
```

```
Var age = 32;
```

```
Var weight = 127.8
```

متغیرها با کلمه کلیدی var تعیین نوع می شوند. نوع ها مشخص نمی شود اما JS دارای نوع هستند.

```
Var wncrollment = 99;
```

اعداد صحیح و نوع , یکسان هستند و اپراتورها شامل + - / % ++ -- \* / =

**comment** توضیحات و

```
// single comment
```

```
/* multi comments */
```

**string type** نوع رشته

```
Var s = "connie client";
```

```
Var fname = s.substring(0, s.indexOf(""));
```

```
Var len = s.lenght;
```

```
Var s2 = 'meli merchant';
```

**تبدیل بین اعداد و رشته ها:**

```
Var count = 10;
```

```
Var s1 = "" + count; //۱۰
```

```
Var s2 = count + "bananas. Ah!"; //۱۰
```

```
Var n1 = parseInt( 42 is answer); //۴۲
```

```
Var n2 = parseFloat ( "booyah"); //na
```

**دسترسی حروف رشته string :**

```
Var firstletter = s[0];
```

```
Var firstletter = s.charAt[0];
```

```
Var lastletter = s.charAt(s.length - 1);
```

مشابه java ، for loop :

```
For ( initialization; condition;update) {  
    Statement;  
}
```

```
Var sum = 0;
```

```
For ( var l = 0; l < 100; i++) {
```

```
    Sum = sum + l;
```

```
}
```

```
Var s1 = "hello";
```

```
VAR S2 = " ";
```

```
FOR ( VAR l=0; l < S.LENGHT; l++) {
```

```
    S2 += S1.CHARaT(l) + S1.CHARaT(l);
```

```
}
```

```
//S2 store.....
```

### کد و برنامه های نمونه برنامه نویسی وب

**HTML :**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Page Title</title>
```

```
</head>
```

```
<body>
```



```
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

یک برنامه نمونه

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```

برنامه ای با صفات :

```
<!DOCTYPE html>
<html>
<body>
  <h2>The title attribute</h2>
  <p title="I'm a tooltip">
    Mouse over this paragraph, to display the title attribute as a tooltip.
  </p>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <a href=https://www.w3schools.com>This is a link</a>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <h1>About W3Schools</h1>
```

```
<p title=About W3Schools>
```

```
    You cannot omit quotes around an attribute value  
    if the value contains spaces.
```

```
</p>
```

```
<p><b>
```

```
    If you move the mouse over the paragraph above,  
    your browser will only display the first word from the title.
```

```
</b></p>
```

```
</body>
```

```
</html>
```

برنامه هایی با موضوع عنوان و Heading :

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <h1>This is heading 1</h1>
```

```
    <p>This is some text.</p>
```

```
    <hr>
```

```
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
</body>
</html>
```

برنامه ای در باره پاراگراف نویسی در وب:

```
<!DOCTYPE html>
<html>
<body>
  <p>
    This paragraph
    contains a lot of lines
    in the source code,
    but the browser
    ignores it.
  </p>
  <p>
    This paragraph
    contains a lot of spaces
    in the source code,
    but the browser
    ignores it.
```

```
</p>
<p>
    The number of lines in a paragraph depends on the size of the browser
    window. If you resize the browser window, the number of lines in this
    paragraph will change.
</p>
</body>
</html>
```

برنامه درباره سبک و Style:

```
<!DOCTYPE html>
<html>
<body style="background-color:powderblue;">
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

برنامه فرمت دهی به متن:

```
<!DOCTYPE html>
<html>
<body>
    <p>This text is normal.</p>
    <p><b>This text is bold.</b></p>
    <p><strong>This text is strong.</strong></p>
    <p><em>This text is emphasized.</em></p>
```

<h2>HTML <small>Small</small> Formatting</h2>

<p>My favorite color is <del>blue</del> red.</p>

<p>This is <sub>subscripted</sub> text.</p>

برنامه استفاده اختصارات لغات و نحوه درج آدرس و عنوان کار :Cite

<!DOCTYPE html>

<html>

<body>

<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

<p>Marking up abbreviations can give useful information to browsers, translation systems and search-engines.</p>

</body>

</html>

.....

<!DOCTYPE html>

<html>

<body>

<p>The HTML address element defines contact information (author/owner) of a document or article.</p>

<address>

Written by John Doe.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

```
</address>
```

```
</body>
```

```
</html>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>The HTML cite element defines the title of a work.</p>
```

```
<p>Browsers usually display cite elements in italic.</p>
```

```
    
```

```
    <p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

```
</body>
```

```
</html>
```

کد برنامه رایانه ای:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <p>Some programming code:</p>
```

```
<code>
```

```
var x = 5;<br>
```

```
var y = 6;<br>
```

```
document.getElementById("demo").innerHTML = x + y;  
</code>  
</body>  
</html>
```

کد درباره کامنت :

```
<!DOCTYPE html>  
<html>  
<body>  
    <!-- This is a comment -->  
    <p>This is a paragraph.</p>  
    <!-- Comments are not displayed in the browser -->  
</body>  
</html>
```

کد برنامه مربوط به استفاده از رنگ ها :

```
<!DOCTYPE html>  
<html>  
<body>  
    <h2 style="background-color:red">  
Background-color set by using red  
</h2>  
<h2 style="background-color:orange">  
Background-color set by using orange  
</h2>
```



```
<h2 style="background-color:yellow">
```

Background-color set by using yellow

```
</h2>
```

```
<h2 style="background-color:blue;color:white">
```

Background-color set by using blue

```
</h2>
```

```
<h2 style="background-color:cyan">
```

Background-color set by using cyan

```
</h2>
```

```
</body>
```

```
</html>
```

برنامه CSS HTML و CSS داخلی و CSS خارجی:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <h1 style="color:blue;">This is a Blue Heading</h1>
```

```
</body>
```

```
</html>
```

.....

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {background-color: powderblue;}
```

```
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

.....

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

برنام با پیوند و link :

```
<!DOCTYPE html>
<html>
<body>
```

```
<a href="https://www.w3schools.com/html/" target="_blank">Visit our HTML tutorial!</a>
```

```
<p>If you set the target attribute to "_blank", the link will open in a new browser window or tab.</p>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
  <p>The image is a link. You can click on it.</p>
```

```
  <a href="default.asp">
```

```
    
```

```
  </a>
```

```
  <p>We have added "border:0" to prevent IE9 (and earlier) from displaying a border around the image.</p>
```

```
</body>
```

```
</html>
```

برنامه و استفاده از تصاویر:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
  <h2>Spectacular Mountain</h2>
```

```

        
</body>
</html>
.....
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
</head>
<body>
<p>It is better to use the style attribute to set the width and height of an image
    (instead of using the width and height attributes), because it prevents
    internal or external styles sheets to change the original size of an
    image:</p>
<p>Using the style attribute:</p>

<p>Using the width and height attributes:</p>

</body>
</html>

```

```

.....
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
</head>
<body>
<p>It is better to use the style attribute to set the width and height of an image
    (instead of using the width and height attributes), because it prevents
internal or external styles sheets to change the original size of an image:</p>
<p>Using the style attribute:</p>

<p>Using the width and height attributes:</p>

</body>
</html>

```

برنامه برای کاربردهای جدول:

```

<!DOCTYPE html>
<html>
<body>
<table style="width:100%">

```

```
<tr>
  <th>Firstname</th>
  <th>Lastname</th>
  <th>Age</th>
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>
</body>
</html>
```

برنامه برای لیست:

```
<!DOCTYPE html>
<html>
<body>
  <h2>Unordered List without Bullets</h2>
  <ul style="list-style-type:none">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
  </ul>
```

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

برنامه بلوک:

```
<!DOCTYPE html>
<html>
<body>
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the
  United Kingdom, with a metropolitan area of over 13 million
  inhabitants.</p>
  <p>Standing on the River Thames, London has been a major settlement
  for two millennia, its history going back to its founding by the Romans, who
  named it Londinium.</p>
</div>
```

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

برنامه با استفاده از class : سبک های مشابه برای عناصر صفحه با نام کلاس یکسان برای آنها:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.cities {
  background-color: black;
  color: white;
  margin: 20px 0 20px 0;
  padding: 20px;
}
</style>
</head>

<body>
<div class="cities">
```

```

<h2>London</h2>
<p>London is the capital of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
<p>Standing on the River Thames, London has been a major settlement for
two millennia, its history going back to its founding by the Romans, who
named it Londinium.</p>
</div>
<div class="cities">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
<p>Situated on the Seine River, it is at the heart of the Île-de-France region,
also known as the région parisienne.</p>
<p>Within its metropolitan area is one of the largest population centers in
Europe, with over 12 million inhabitants.</p>
</div>
<div class="cities">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and
the most populous metropolitan area in the world.</p>
<p>It is the seat of the Japanese government and the Imperial Palace, and
the home of the Japanese Imperial Family.</p>
<p>The Tokyo prefecture is part of the world's most populous metropolitan
area with 38 million people and the world's largest urban economy.</p>
</div>
</body>
</html>

```

استفاده از frame :

```

<!DOCTYPE html>
<html>
<body>
<iframe height="300px" width="100%" src="demo_iframe.htm"
name="iframe_a"></iframe>

```



```

<p><a href="https://www.w3schools.com"
target="iframe_a">W3Schools.com</a></p>
<p>When the target of a link matches the name of an iframe, the link will
open in the iframe.</p>
</body>
</html>

```

جاوا اسکرپت نویسی:

```

<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
</body>
</html>

```

```

.....
<!DOCTYPE html>
<html>
<body>
<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function myFunction() {
  document.getElementById("demo").style.fontSize = "25px";
  document.getElementById("demo").style.color = "red";
}
</script>
<button type="button" onclick="myFunction()">Click Me!</button>
</body>
</html>
.....

```

```

<!DOCTYPE html>
<html>
<body>
<script>
function light(sw) {
    var pic;
    if (sw == 0) {
        pic = "pic_bulboff.gif"
    } else {
        pic = "pic_bulbon.gif"
    }
    document.getElementById('myImage').src = pic;
}
</script>

<p>
<button type="button" onclick="light(1)">Light On</button>
<button type="button" onclick="light(0)">Light Off</button>
</p>

</body>
</html>

```

برنامه مسیره‌های فایل:

```

<!DOCTYPE html>
<html>
<body>
<h2>Using a Relative File Path</h2>


</body>
</html>

```

کد واکنشی و Responsive :

```

<!DOCTYPE html>
<html lang="en-us">

```

```
<head>
<style>
.city {
  float: left;
  margin: 10px;
  padding: 10px;
  max-width: 300px;
  height: 300px;
  border: 1px solid black;
}
</style>
</head>
<body>
<h1>Responsive Web Design Demo</h1>
<h2>Resize this responsive page!</h2>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
  <p>It is the most populous city in the United Kingdom, with a
metropolitan area of over 13 million inhabitants.</p>
</div>
<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
  <p>The Paris area is one of the largest population centers in Europe, with
more than 12 million inhabitants.</p>
</div>
<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
  <p>It is the center of the Greater Tokyo Area, and the most populous
metropolitan area in the world.</p>
</div>
```

```

<div class="city">
  <h2>New York</h2>
  <p>The City of New York is the most populous city in the United
States.</p>
  <p>New York is an important center for international diplomacy and has
been described as the cultural and financial capital of the world.</p>
</div>

</body>
</html>

```

فرم های Html ی و عناصر فرم:

```

<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>
<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>

</body>
</html>

```

```

.....
<!DOCTYPE html>
<html>
<body>
<form>

```

```

    First name:<br>
    <input type="text" name="firstname">
    <br>
    Last name:<br>
    <input type="text" name="lastname">
</form>
<p>Note that the form itself is not visible.</p>
<p>Also note that the default width of a text input field is 20
characters.</p>

</body>
</html>
.....
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  <select name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
  <br><br>
  <input type="submit">
</form>

</body>
</html>

```

نوع های ورودی در HTML :

```

<!DOCTYPE html>
<html>
<body>

```

```
<form action="/action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>
```

<p>If you click "Submit", the form-data will be sent to a page called "/action\_page.php".</p>

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

کد ویدئویی با HTML :

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```

برنامه پخش صدا با کد HTML :

```
<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
```

```

    <source src="horse.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
  </audio>
</body>
</html>

```

برنامه plug in :

```

<!DOCTYPE html>
<html>
<body>

<object width="400" height="50" data="bookmark.swf"></object>

</body>
</html>
.....

```

کد موقعیت یابی:

```

<!DOCTYPE html>
<html>
<body>
<p>Click the button to get your coordinates.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}
function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}

```

```
    }  
  </script>  
  
  </body>  
</html>
```

## برنامه هایی با CSS

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
  body {  
    background-color: lightblue;  
  }  
  h1 {  
    color: white;  
    text-align: center;  
  }  
  p {  
    font-family: verdana;  
    font-size: 20px;  
  }  
</style>  
</head>  
<body>  
  
<h1>My First CSS Example</h1>
```



```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  text-align: center;
```

```
  color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Every paragraph will be affected by the style.</p>
```

```
<p id="para1">Me too!</p>
```

```
<p>And me!</p>
```

```
</body>
```

```
</html>
```

```
.....
```

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
h1 {
  background-color: green;
}
div {
  background-color: lightblue;
}
p {
  background-color: yellow;
}
</style>
</head>
<body>
<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
</body>
</html>
```

.....

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
div {
width: 100%;
height: 400px;
background-image: url('img_flowers.jpg');
background-repeat: no-repeat;
background-size: contain;
border: 1px solid red;
}
</style>
</head>
<body>

<p>Resize the browser window to see the effect.</p>
<div></div>
</body>
</html>
```

## جاوا اسکریپت

جاوا اسکریپت برنامه نویسی برای تعیین رفتار صفحات وب است. در حالیکه HTML برای تعریف محتوی صفحات وب و CSS برای مخص کردن زمینه و بستر صفحات وب است.

```
<!DOCTYPE html>
<html>
<body>
<h1>What Can JavaScript Do?</h1>
<p>JavaScript can change HTML attributes.</p>
<p>In this case JavaScript changes the src (source) attribute of an image.</p>
<button
onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the
light</button>

<button
onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the
light</button>
</body>
</html>
```

.....

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>
```

```
<h1>JavaScript in Head</h1>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

```
.....
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<p>My first paragraph.</p>
<script>
window.alert(5 + 6);
</script>
</body>
</html>
```

```
.....
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript is Case Sensitive</h1>
<p>Try change lastName to lastname.</p>

<p id="demo"></p>
```

```

<script>
var lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
document.getElementById("demo").innerHTML = lastName;
</script>
</body>
</html>
.....
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>
<p>JavaScript code blocks are written between { and }</p>
<button type="button" onclick="myFunction()">Click Me!</button>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
function myFunction() {
    document.getElementById("demo1").innerHTML = "Hello Dolly.";
    document.getElementById("demo2").innerHTML = "How are you?";
}
</script>
</body>

```

```
</html>
```

```
..... variables .....
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var carName1 = "Volvo XC60";
```

```
var carName2 = 'Volvo XC60';
```

```
var answer1 = "It's alright";
```

```
var answer2 = "He is called 'Johnny'";
```

```
var answer3 = 'He is called "Johnny"';
```

```
document.getElementById("demo").innerHTML =
```

```
carName1 + "<br>" +
```

```
carName2 + "<br>" +
```

```
answer1 + "<br>" +
```

```
answer2 + "<br>" +
```

```
answer3;
```

```
</script>
```

```
</body>
```

```
</html>
```

```
..... Functions .....
```

```
<!DOCTYPE html>
```

```
<html>
<body>
<h1>JavaScript Functions</h1>
<p>This example calls a function which performs a calculation, and returns the
result:</p>
<p id="demo"></p>
<script>
function myFunction(p1, p2) {
    return p1 * p2;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>
</body>
</html>
```

.....Date Formats .....

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript ISO Dates</h1>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
new Date("2015");
</script>
</body>
```



</html>

..... Best Practics .....

<!DOCTYPE html>

<html>

<body>

<p>Remove the comment (at the beginning of the lines) to test each case:</p>

<p id="demo"></p>

<script>

var x = 5;

//x = 5 + 7; // x.valueOf() is 12, typeof x is a number

//x = 5 + "7"; // x.valueOf() is 57, typeof x is a string

//x = "5" + 7; // x.valueOf() is 57, typeof x is a string

//x = 5 - 7; // x.valueOf() is -2, typeof x is a number

//x = 5 - "7"; // x.valueOf() is -2, typeof x is a number

//x = "5" - 7; // x.valueOf() is -2, typeof x is a number

//x = 5 - "x"; // x.valueOf() is NaN, typeof x is a number

document.getElementById("demo").innerHTML = x.valueOf() + " " + typeof x;

</script>

</body>

</html>

..... Forms API Input .....

<!DOCTYPE html>

<html>

```

<body>
<p>Enter a number and click OK:</p>
<input id="id1" type="number" min="100" max="300" required>
<button onclick="myFunction()">OK</button>
<p>If the number is less than 100 or greater than 300, an error message will be
displayed.</p>
<p id="demo"></p>
<script>
function myFunction() {
    var inpObj = document.getElementById("id1");
    if (inpObj.checkValidity() == false) {
        document.getElementById("demo").innerHTML = inpObj.validationMessage;
    } else {
        document.getElementById("demo").innerHTML = "Input OK";
    }
}
</script>
</body>
</html>

```

..... Input Forms .....

```

<!DOCTYPE html>
<html>
<body>

<form action="/action_page_post.php" method="post">

```

```
<input type="text" name="fname" required>
<input type="submit" value="Submit">
</form>
<p>If you click submit, without filling out the text field,
your browser will display an error message.</p>
</body>
</html>
```

..... Java Functions .....

```
<!DOCTYPE html>
<html>
<body>
<p>Setting a default value to a function parameter.</p>
<p id="demo"></p>
<script>
function myFunction(x, y) {
  if (y === undefined) {
    y = 0;
  }
  return x * y;
}
document.getElementById("demo").innerHTML = myFunction(4);
</script>
</body>
</html>
```

## برنامه نویسی با PHP

```
!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";

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

.....

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";

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

..... variables.....

```
<!DOCTYPE html>
<html>
<body>
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";

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

.....Print .....

```
<!DOCTYPE html>
<html>
<body>
<?php
echo "<h2>PHP is Fun!</h2>";

echo "Hello world!<br>";

echo "I'm about to learn PHP!<br>";

echo "This ", "string ", "was ", "made ", "with multiple parameters.";

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

.....String.....

```
<!DOCTYPE html>
<html>
<body>
<?php
$txt = "W3Schools.com";

echo "I love $txt!";

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

..... If then else .....

```
<!DOCTYPE html>
<html>
<body>
<?php
$t = date("H");

if ($t < "20") {

    echo "Have a good day!";

}

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

..... switch case .....

```

<!DOCTYPE html>
<html>
<body>
<?php
$favcolor = "red";
switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":\
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
</body>
</html>

```

..... While loop .....

```

<!DOCTYPE html>
<html>
<body>
<?php
$x = 1;
while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
</body>
</html>

```

..... function .....

```

<!DOCTYPE html>
<html>

```

```

<body>
<?php
function writeMsg() {
    echo "Hello world!";
}
writeMsg();
?>
</body>
</html>

```

.....PHP Form Handling .....

```

<!DOCTYPE HTML>
<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>

```

.....PHP Form URL, Email .....

```

<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {

```

```

$name = test_input($_POST["name"]);
// check if name only contains letters and whitespace
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
    $nameErr = "Only letters and white space allowed";
}
}

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}

if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid
    if (!preg_match("/\b(?:(:?https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?~_!|:.,;]*[-a-z0-9+&@#\/%?~_!|:.,;]*/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}

if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = test_input($_POST["comment"]);

```



```

}
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = test_input($_POST["gender"]);
}
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field.</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF
"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    E-mail: <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Website: <input type="text" name="website">
    <span class="error"><?php echo $websiteErr;?></span>
    <br><br>
    Comment: <textarea name="comment" rows="5" cols="40"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female
    <input type="radio" name="gender" value="male">Male
    <span class="error">* <?php echo $genderErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

```

```
<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>
</body>
</html>
```

**Reference:**

- Web Programming step by step: written by Ms. Jessica miller
- [www.w3schools.com](http://www.w3schools.com)