



# مبانی طراحی سایت

تألیف محمدحسین معین زاده

به نام خدا

## فصل اول HTML

HTML چیست ؟ مخفف Hyper Text Markup Language است ، به معنی زبان نشانه گذاری استاندارد برای ایجاد صفحات وب.

مثال اول:

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

اعلان تعریف **<!DOCTYPE html>** می کند که این سند یک سند HTML5 است

**<html>** عنصر اصلی یک صفحه HTML است

**<head>** این عنصر حاوی اطلاعات متا در مورد صفحه HTML است

**<title>** این عنصر عنوانی را برای صفحه HTML مشخص می کند (که در نوار عنوان مرورگر یا در برگه صفحه نشان داده شده است)

**<body>** این عنصر بدنه سند را تعریف می کند و محفظه ای برای تمام محتویات قابل مشاهده است، مانند سرفصل ها، پاراگراف ها، تصاویر، لینک ها، جداول، لیست ها و غیره.

**<h1>** عنصر یک عنوان بزرگ را تعریف می کند

**<p>** عنصر یک پاراگراف را تعریف می کند

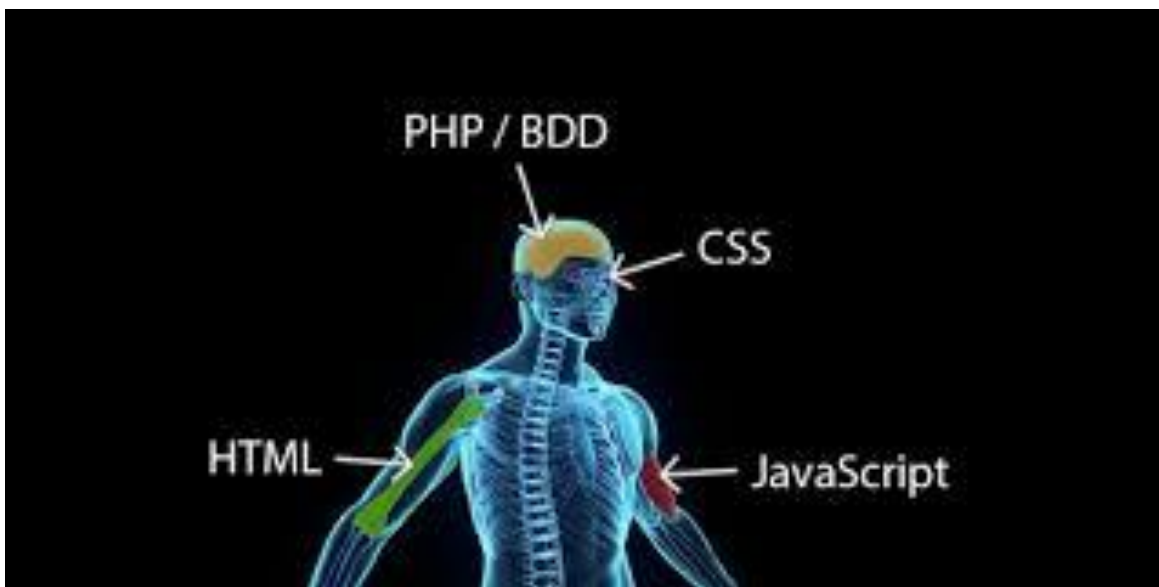
## ویرایشگر های HTML

صفحات وب را می توان با استفاده از ویرایشگرهای حرفه ای HTML ایجاد و اصلاح کرد.

با این حال، برای یادگیری HTML یک ویرایشگر متن ساده مانند Notepad (PC) یا TextEdit (Mac) را توصیه می کنیم.

ما معتقدیم که استفاده از یک ویرایشگر متن ساده راه خوبی برای یادگیری HTML است.

## اصول های پایه



### اسناد HTML

همه اسناد HTML باید با یک اعلان نوع سند شروع شوند. `<!DOCTYPE html>` :

خود سند HTML با شروع `<html>` و به پایان می رسد. `</html>`

قسمت قابل مشاهده سند HTML بین `<body>` و `</body>` است.

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

## Heading یا سرفصل ها

سرفصل های HTML با تگ های **<h1>** تعریف می شوند **<h6>**.

**<h1>** مهمترین عنوان را تعریف می کند **<h6>**. کم اهمیت ترین عنوان را تعریف می کند

مثال :

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

## پاراگراف ها

پاراگراف های HTML با **<p>** تگ تعریف می شوند:

مثال :

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

## لینک ها

پیوندهای HTML با **<a>** تگ تعریف می شوند:

مثال:

```
<a href="www.google.com">This is a link</a>
```

مقصد پیوند در **href** ویژگی مشخص شده است .

از ویژگی ها برای ارائه اطلاعات اضافی در مورد عناصر HTML استفاده می شود.

در فصل بعد در مورد ویژگی ها یا Attributes بیشتر خواهید آموخت.

## تصاویر

تصاویر HTML با **<img>** تگ تعریف می شوند.

فایل منبع (**src**) ، متن جایگزین (**alt**) ، **width** و **height** به عنوان ویژگی ارائه می شود:

مثال:

```

```

# نحوه مشاهده سورس HTML

آیا تا به حال یک صفحه وب را دیده اید و از خود پرسیده اید "هی! آنها چگونه این کار را کردند؟"

## مشاهده کد سورس HTML:

در یک صفحه HTML کلیک راست کرده و "View Page Source" (در کروم) یا "View Source" (در Edge یا موارد مشابه را در سایر مرورگرها انتخاب کنید. با این کار پنجره ای باز می شود که حاوی کد منبع HTML صفحه است.

یک عنصر HTML را بررسی کنید: روی یک عنصر (یا یک ناحیه خالی) کلیک راست کرده و "Inspect" یا "Inspect Element" را انتخاب کنید تا ببینید از چه عناصری تشکیل شده است) هم HTML و هم CSS را خواهید دید. (همچنین می توانید HTML یا CSS را بلافاصله در پانل عناصر یا سبک هایی که باز می شود ویرایش کنید.

## فصل ۲ عناصر

عنصر HTML همه چیز از تگ شروع تا تگ پایان است:

`< tagname > ...محتوا به اینجا می رود < /tagname >`

نمونه هایی از برخی عناصر: HTML

`< h1 >` اولین عنوان من

`< p >` اولین پاراگراف من.

## عناصر تو در تو

عناصر HTML را می توان تو در تو قرار داد (به این معنی که عناصر می توانند عناصر دیگری را نیز در بر گیرند).

تمام اسناد HTML از عناصر HTML تو در تو تشکیل شده اند.

مثال زیر شامل چهار عنصر

HTML ( `<html>` ، `<body>` ) `<h1>` : `<p>`

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

## هرگز تگ پایان را نادیده نگیرید

برخی از عناصر HTML به درستی نمایش داده می شوند، حتی اگر تگ پایان را فراموش کنید:

```
<html>
```

```
<body>
```

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

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

```
</body>
```

```
</html>
```

با این حال، هرگز به این اعتماد نکنید! اگر تگ پایان را فراموش کنید ممکن است نتایج و خطاهای غیرمنتظره ای رخ دهد!

# عناصر HTML خالی

عناصر HTML بدون محتوا عناصر خالی نامیده می شوند.

تگ `<br>` یک شکست خط را تعریف می کند و یک عنصر خالی بدون برچسب بسته است.

`<p>`This is a `<br>` paragraph with a line break.`</p>`

## HTML به حروف بزرگ و کوچک حساس نیست

تگ های HTML به حروف بزرگ و کوچک حساس نیستند `<P>`: یعنی همان `<p>`.

استاندارد HTML نیازی به برچسب های حروف کوچک ندارد.

توضیحات	تگ
ریشه یک سند HTML را تعریف می کند	<code>&lt;HTML&gt;</code>
بدنه سایت را طراحی میکند	<code>&lt;BODY&gt;</code>
سر فصل را طراحی میکند	<code>&lt;H1&gt; TO &lt;H6&gt;</code>



## فصل ۳ Attributes یا ویژگی ها

- همه عناصر HTML می توانند دارای ویژگی باشند
- ویژگی ها اطلاعات بیشتری در مورد عناصر ارائه می دهند
- ویژگی ها همیشه در تگ شروع مشخص می شوند
- مشخصه ها معمولاً به صورت جفت نام/مقدار مانند `name = "value"` می آیند

### ویژگی href

تگ `<a>` یک هایپرلینک را تعریف می کند. این `href` ویژگی URL صفحه ای را که پیوند به آن می رود مشخص می کند:

```
<a href="https://www.google.com">Visit Schools</a>
```

### ویژگی src

تگ `<img>` برای جاسازی یک تصویر در یک صفحه HTML استفاده می شود. ویژگی `src` مسیر تصویری که باید نمایش داده شود را مشخص می کند:

```

```

### تعیین URL در src

**URL مطلق** - پیوندهایی به یک تصویر خارجی که در وب سایت دیگری میزبانی شده است. مثال. `src="https://yasdl.com/images/img_girl.jpg"`

**نکات:** تصاویر خارجی ممکن است تحت حق چاپ باشند. اگر اجازه استفاده از آن را دریافت نکنید، ممکن است قوانین کپی رایت را نقض کنید. علاوه بر این، شما نمی توانید تصاویر خارجی را کنترل کنید. می توان آن را به طور ناگهانی حذف یا تغییر داد.

## صفات عرض و ارتفاع

تگ `<img>` همچنین باید دارای ویژگی های `width` و `height` باشد که عرض و ارتفاع تصویر (بر حسب پیکسل) را مشخص می کند:

```

```

## ویژگی alt

`alt` مشخصه مورد نیاز برای `<img>` تگ یک متن جایگزین برای یک تصویر مشخص می کند، اگر تصویر به دلایلی نمایش داده نشود. این می تواند به دلیل اتصال کند، یا خطا در `src` ویژگی، یا اگر کاربر از صفحه خوان استفاده می کند، باشد.

```

```

## ویژگی سبک یا style

این `style` ویژگی برای افزودن سبک به یک عنصر مانند رنگ، فونت، اندازه و موارد دیگر استفاده می شود.

```
<p style="color:red;">This is a red paragraph.</p>
```

# ویژگی زبان

برای اعلام زبان صفحه وب، همیشه باید **lang** ویژگی را در داخل تگ قرار دهید **<html>**. این برای کمک به موتورهای جستجو و مرورگرها است.

مثال زیر انگلیسی را به عنوان زبان مشخص می کند:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

کدهای کشور همچنین می توانند به کد زبان در **lang** ویژگی اضافه شوند. بنابراین، دو کاراکتر اول زبان صفحه HTML و دو کاراکتر آخر کشور را مشخص می کنند.

مثال زیر انگلیسی را به عنوان زبان و ایالات متحده را به عنوان کشور مشخص می کند:

```
<!DOCTYPE html>
```

```
<html lang="en-US">
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

## عنوان صفت

این **title** ویژگی اطلاعات اضافی در مورد یک عنصر را تعریف می کند.

هنگامی که ماوس را روی عنصر قرار می دهید، مقدار ویژگی **title** به عنوان یک راهنمای ابزار نمایش داده می شود:

`<p title="I'm a tooltip">This is a paragraph.</p>`

## خلاصه ی فصل

- همه عناصر HTML می توانند دارای ویژگی باشند
- مشخصه **href** ی **a**>URL صفحه ای را که پیوند به آن می رود مشخص می کند
- ویژگی **src** ی **img** مسیر تصویری که قرار است نمایش داده شود را مشخص می کند
- **width** و ویژگی **height** های **img** اطلاعات اندازه تصاویر را فراهم می کند
- ویژگی **alt** یک **img** متن جایگزین برای یک تصویر فراهم می کند
- این **style** ویژگی برای افزودن سبک به یک عنصر مانند رنگ، فونت، اندازه و موارد دیگر استفاده می شود
- ویژگی تگ زبان صفحه وب را اعلام می کند **lang** **<html>**
- این **title** ویژگی اطلاعات اضافی در مورد یک عنصر را تعریف می کند

## فصل ۴ Paragraph و heading

عنوان های HTML عناوین یا زیرنویس هایی هستند که می خواهید در یک صفحه وب نمایش دهید.

### سرفصل ها مهم هستند

موتورهای جستجو از سرفصل ها برای نمایه سازی ساختار و محتوای صفحات وب شما استفاده می کنند.

کاربران اغلب یک صفحه را بر اساس عناوین آن مرور می کنند. استفاده از سرفصل ها برای نشان دادن ساختار سند مهم است.

**<h1>** سرفصل ها باید برای سرفصل های اصلی و به دنبال آن **<h2>** سرفصل ها و سپس کم اهمیت تر **<h3>** و غیره استفاده شوند.

### عناوین بزرگتر

هر عنوان HTML یک اندازه پیش فرض دارد. با این حال، می توانید با **style** استفاده از ویژگی CSS، اندازه هر عنوان را با ویژگی مشخص کنید: **font-size**

### مثال

```
<h1 style="font-size:60px;">Heading 1</h1>
```

## پاراگراف اصولی

شما نمی توانید مطمئن باشید که HTML چگونه نمایش داده می شود.

صفحه نمایش های بزرگ یا کوچک و پنجره های تغییر اندازه نتایج متفاوتی ایجاد می کند.

با HTML ، نمی توانید نمایشگر را با افزودن فاصله یا خطوط اضافی در کد HTML خود تغییر دهید.

هنگامی که صفحه نمایش داده می شود، مرورگر به طور خودکار هر فاصله و خط اضافی را حذف می کند

## جدا سازی پاراگراف ها از یک دیگر

تگ **<hr>** یک شکست موضوعی را در یک صفحه HTML تعریف می کند و اغلب به عنوان یک قانون افقی نمایش داده می شود.

این **<hr>** عنصر برای جداسازی محتوا (یا تعریف تغییر) در یک صفحه HTML استفاده می شود:

### مثال

```
<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>
```

تگ **<hr>** یک تگ خالی است، به این معنی که تگ پایانی ندارد

## Break یا شکست خط

عنصر `<br>` HTML یک شکست خط را تعریف می کند.

`<br>` اگر می خواهید بدون شروع یک پاراگراف جدید، یک خط شکسته (یک خط جدید) ایجاد کنید، استفاده کنید

### مثال

`<p>This is<br>a paragraph<br>with line breaks.</p>`

تگ `<br>` یک تگ خالی است، به این معنی که تگ پایانی ندارد.

## حفظ شکست خط و فاصله ها

عنصر `<pre>` HTML متن از پیش فرمت شده را تعریف می کند.

متن داخل یک `<pre>` عنصر با فونت با پهنای ثابت (معمولاً Courier نمایش داده می شود و هم فاصله ها و هم شکستگی های خط را حفظ می کند:

### مثال

`<pre>`

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

`</pre>`

## استایل یا سبک

ویژگی **style** HTML برای افزودن سبک به یک عنصر مانند رنگ، فونت، اندازه و موارد دیگر استفاده می شود.

### مثال

من قرمز هستم

من آبی هستم

من بزرگ هستم

تنظیم سبک یک عنصر HTML را می توان با **style** ویژگی انجام داد.

ویژگی **style** HTML دارای نحو زیر است:

```
<tagname style='property.value; ">
```

این ویژگی یک ویژگی CSS است .

### تنظیم رنگ پس زمینه

ویژگی **background-color** CSS رنگ پس زمینه یک عنصر HTML را مشخص می کند.

رنگ پس زمینه یک صفحه را روی آبی تنظیم کنید:

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



## مثال

رنگ پس زمینه را برای دو عنصر مختلف تنظیم کنید:

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

## رنگ متن

ویژگی **color** CSS رنگ متن را برای یک عنصر HTML تعریف می کند:

### مثال

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

## فونت ها

ویژگی **font-family** CSS فونت مورد استفاده برای یک عنصر HTML را تعریف می کند:

### مثال

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

## اندازه متن

ویژگی **font-size** CSS اندازه متن را برای یک عنصر HTML تعریف می کند:

### مثال

```
<h1 style="font-size:300%;">This is a heading</h1>
```

```
<p style="font-size:160%;">This is a paragraph.</p>
```

## تراز متن

ویژگی **text-align** CSS تراز افقی متن را برای یک عنصر HTML تعریف می کند:

### مثال

```
<h1 style="text-align:center;">Centered Heading</h1>
```

```
<p style="text-align:center;">Centered paragraph.</p>
```

# قالب بندی متن HTML

HTML شامل چندین عنصر برای تعریف متن با معنای خاص است.

## مثال

**This text is bold**

*This text is italic*

This is <sub>subscript</sub> and <sup>superscript</sup>

## عناصر قالب بندی HTML

عناصر قالب بندی برای نمایش انواع خاصی از متن طراحی شده اند:

- **<b>** متن پررنگ
- **<strong>** متن مهم
- **<i>** متن کج
- **<em>** متن تاکید شده
- **<mark>** متن علامت گذاری شده
- **<small>** متن کوچکتر
- **<del>** متن حذف شده
- **<ins>** متن درج شده
- **<sub>** متن زیرنویس
- **<sup>** متن بالانویس

## عناصر <b> و <strong> HTML

عنصر <b> HTML متن پررنگ را بدون هیچ اهمیت اضافی تعریف می کند.

### مثال

<b>This text is bold</b>

<strong> عنصر HTML متن را با اهمیت زیاد تعریف می کند. محتوای داخل معمولاً به صورت پررنگ نمایش داده می شود.

### مثال

<strong>This text is important!</strong>

## عناصر <i> و <em> HTML

عنصر <i> HTML بخشی از متن را با صدا یا حالت متناوب تعریف می کند. محتوای داخل معمولاً به صورت مورب نمایش داده می شود.

**نکته:** این <i> برچسب اغلب برای نشان دادن یک اصطلاح فنی، عبارتی از زبان دیگر، یک فکر، نام کشتی و غیره استفاده می شود.

### مثال

<i>This text is italic</i>

عنصر <em> HTML متن تاکید شده را تعریف می کند. محتوای داخل معمولاً به صورت مورب نمایش داده می شود.

**نکته:** یک صفحه خوان <em> با استفاده از استرس کلامی، کلمات را با تاکید تلفظ می کند.

### مثال

<em>This text is emphasized</em>

## عنصر <small> HTML

عنصر <small> HTML متن کوچکتر را تعریف می کند:

### مثال

<small>This is some smaller text.</small>

## عنصر HTML <mark>

عنصر <mark> HTML متنی را تعریف می کند که باید علامت گذاری یا برجسته شود:

### مثال

<p>Do not forget to buy <mark>milk</mark> today.</p>

## عنصر HTML <del>

عنصر <del> HTML متنی را تعریف می کند که از یک سند حذف شده است. مرورگرها معمولاً با یک خط از متن حذف شده برخورد می کنند:

### مثال

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

## عنصر HTML <ins>

عنصر HTML متنی <ins> را تعریف می کند که در یک سند درج شده است. مرورگرها معمولاً زیر متن درج شده خط می کشند:

### مثال

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

## عنصر HTML <sub>

عنصر **<sub>** HTML متن زیرنویس را تعریف می کند. متن زیرنویس نیم کاراکتر زیر خط معمولی ظاهر می شود و گاهی اوقات با فونت کوچکتر ارائه می شود. متن زیرنویس را می توان برای فرمول های شیمیایی مانند  $H_2O$  استفاده کرد:

### مثال

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

## عنصر HTML <sup>

عنصر **<sup>** HTML متن بالا را تعریف می کند. متن فوق نویس نیم کاراکتر بالاتر از خط معمولی ظاهر می شود و گاهی اوقات با فونت کوچکتر ارائه می شود. متن فوق اسکرپت را می توان برای پاورقی استفاده کرد، مانند: <sup>[1]</sup> WWW

### مثال

`<p>This is <sup>superscripted</sup> text.</p>`

## ایجاد استایل داخلی در heder

۱	تعریف و باز کردن استایل	<code>&lt;style type="text/css"&gt;</code>
۲	اختصاص دادن رنگ زرد به پس زمینه سایت	<code>body{ background-color: yellow;}</code>
۳	اختصاص دادن رنگ خاکستری به تگ h1	<code>h1{color: gray;}</code>
۴	راستچین کردن تگ های h1 و p	<code>h1, p{text-align: right;}</code>
۵	بستن استایل	<code>&lt;/style&gt;</code>

ایجاد استایل خارجی (ایجاد فایل CSS)  
برای این کار ابتداء یک فایل با پسوند CSS ایجاد کنید



سپس مقادیر را به صورت دلخواه وارد کنید

مثال:

۱	باز کردن تنظیمات body	body{
۲	وارد کردن مقدار رنگ به پس زمینه	background-color: yellow;}
۳	باز کردن تنظیمات h1	h1{
۴	تخصیص رنگ	color: gray;}
۵	بازکردن تنظیمات h1 و p	h1, p{
۶	تخصیص جهت نوشتن	text-align: right;}

## شناختن فایل CSS به HTML

بعد از نوشتن کد های فوق در بخش Head کد زیر را وارد میکنیم

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

**Href** : آدرس محل فایل css.

**Type** : نوع سندی که در بخش href مشخص شده، تعریف میشود.

برای پرونده css مقدار text/css وارد میشود.

**Rel** : نوع ارتباط صفحه html با پرونده استایل شیت مورد نظر مشخص میشود.

## تگ <div>

تعریف و کاربرد

تگ <div> یک بخش یا بخش را در یک سند HTML تعریف می کند.

تگ <div> به عنوان یک محفظه برای عناصر HTML استفاده می شود - که سپس با CSS استایل داده می شود یا با جاوا اسکریپت دستکاری می شود.

تگ <div> به راحتی با استفاده از ویژگی class یا id استایل بندی می شود.

هر نوع محتوایی را می توان در تگ <div> قرار داد!

مثال

```
<html>
<head>
<style>
.myDiv {
border: 5px outset red;
background-color: lightblue;
text-align: center;
}
</style>
</head>
<body>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>

</body>
</html>
```

## اضافه کردن audio به سایت.

برای پخش یک فایل صوتی در HTML، از عنصر <audio> استفاده میکنیم:

مثال:

```
<audio controls>
  <source src="location audio" type="audio/ogg">
  <source src=" location audio " type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

audio -HTML چگونه کار می کند

ویژگی **controls** کنترل های صوتی مانند پخش، مکث و صدا را اضافه می کند.

عنصر **< source >** به شما امکان می دهد فایل های صوتی دیگری را که مرورگر ممکن است از بین آنها انتخاب کند، مشخص کنید. مرورگر از اولین فرمت شناسایی شده استفاده خواهد کرد.

متن بین تگ های **<audio>** و **<audio/>** فقط در مرورگرهایی نمایش داده می شود که از عنصر **<audio>** پشتیبانی نمی کنند.

## اضافه کردن video به سایت.

عنصر **<video >** برای نمایش یک ویدیو در یک صفحه وب استفاده می شود.

مثال:

```
<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>
```

## ایجاد form برای وارد کردن اطلاعات

این فصل ویژگی های مختلف عنصر form را توضیح می دهد.

## Action (۱)

ویژگی action عملی را که باید هنگام ارسال فرم انجام شود را مشخص می کند.

معمولا زمانی که کاربر روی دکمه ارسال کلیک می کند، داده های فرم به فایلی در سرور ارسال می شود.

در مثال زیر، داده های فرم به فایلی به نام "getaction.php" ارسال می شود. این فایل حاوی یک اسکریپت سمت سرور است که داده های فرم را مدیریت می کند:

مثال:

```
<form action="/ getaction.php ">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

داده های فرم را به " getaction.php" ارسال میکند

## ویژگی Method

ویژگی متد روش HTTP را مشخص می کند که در هنگام ارسال داده های فرم استفاده می شود.

داده های فرم را می توان به عنوان متغیرهای URL (با متد "GET") یا به عنوان پست HTTP (متد "POST") ارسال کرد.

روش پیش فرض HTTP هنگام ارسال داده های فرم GET است.

```
<form action="/action_page.php" method="get">
```

```
<form action="/action_page.php" method="POST">
```

روش ارسال	توضیح
GET	در این روش، داده ها و اطلاعات نمون برگ از طریق نوار نشانی مرورگر منتقل می شوند. امنیت این روش پایین است زیرا همراه با نمایش داده ها است. امکان بارگذاری پرونده وجود ندارد و فقط می تواند نویسه های اسکی را ارسال کند.
POST	در این روش امنیت تبادل داده بالا است در این روش برای ارسال داده ها در PHP استفاده از روش POST است. در این حالت تمام داده ها از طریق هدرهای HTTP برای مقصد ارسال می شود.

پایان مبانی HTML

## فصل دوم JavaScript And PHP

```

});
this.secondHeader[j] = secondHeaderByWeek;
});

~.map(this.consultants, consultant => {
  ~.forEach(this.firstHeader, data => {
    ~.forEach(this.timeSlotsData, timeSlot => {
      let dateTime = moment(data.date + ' ' + timeSlot);
      consultant[~.snakeCase(data.label + ' ' + timeSlot)] = {
        'isAvailable': this.consultantIsAvailable(consultant, dateTime),
        'dateTime': dateTime
      };
    });
  });
});
return consultant;
}

```

# What is JavaScript?

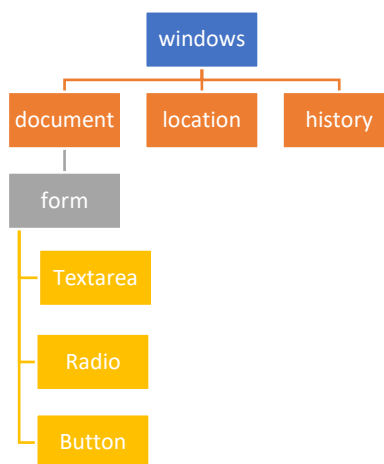
## JavaScript چیست؟

جاوا اسکریپت یک زبان اسکریپت نویسی مبتنی بر اشیا و سطح بالا است که کاربردهای گسترده ای در طراحی تارنما ها و صفحات اینترنتی جهت بهبود کارایی و شکل دهی به ظاهر صفحات وب دارد. با کمک ویژگی های تعاملی و ایجاد واکنش به رویدادها را اضافه کنید HTML جاوا اسکریپت می توانید به صفحات و اطلاعات نمون برگ ها را ثبت و ارسال کنید.

## نحوه دسترسی JavaScript به HTML :

زمانی که یک صفحه وب بارگذاری می شود، مرورگر از آن صفحه، یک مدل شیء گرا (Document Object Model) (DOM) ایجاد میکند و جاوا اسکریپت از طریق این مدل می تواند به تمام عناصر HTML صفحه وب دسترسی داشته باشد. در این مدل هر یک از عناصر HTML یک شیء محسوب میشود.

مثال :



```
<html>
<body>
<form action="site.php">
<input type="txt">
<input type="radio">
<input type="button">
</form>
</body>
</html>
```

روش های استفاده از Java Script



الف) قرار دادن کدهای جاوا در پرونده ای از نوع HTML و استفاده از اسکریپت در بخش `<body>` و `<head>`.

مثال :

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

.....

```
</script>
```

ب) ایجاد پرونده اسکریپت خارجی و ذخیره آن با پسوند js

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

```
</script>
```

**تعریف متغیر :**

در java Script برای تعریف متغیر از کلمه کلیدی **var** استفاده میشود و بعد از آن نام متغیر قرار میگیرد.

مثال:

```
var x = 10;
```

```
var n = 100 , name;
```

```
name = "abass";
```

## قوانین نام گذاری متغیر ها :

۱. یک متغیر نمی تواند ابتدای آن عدد باشد.
۲. درون یک متغیر نمی تواند از علامت های +، -، \*، / و ... استفاده شود
۳. تنها علامتی که در نامگذاری متغیر می تواند استفاده شود علامت \_ (زیر خط)
۴. نام متغیر ها نمی تواند فاصله دار باشد.
۵. ابتدای نام یک متغیر میتواند از \_ (استفاده شود).
۶. نام متغیر نمی تواند از کلمات کلیدی باشد. مثال از نام گذاری (صحیح و غلط متغیر ها )

مثال :

غلط عدم رعایت قانون ۱: 3i

صحیح : i\_max

غلط عدم رعایت قانون ۴: y in

غلط عدم رعایت قانون ۲: X-y

غلط عدم رعایت قانون ۶: DIM

### نمایش پیام با متد alert

پنجره دریافت تأیید، جهت اعلان یک پیام به کاربر و دریافت نظر وی مبنی بر پذیرش یا عدم پذیرش پیام مورد نظر استفاده می شود.

```
<script type="text/javascript">  
Window.alert("welcome to site");  
</script>
```

### ساخت کادر تایید یا لغو

```
<script type="text/javascript">  
    var a=window.confirm("your next?");  
    if (a==true)  
        window.alert("you click yes");  
    else  
        window.alert("you click cancel");  
</script>
```

Confirm : یک کادر با دو گزینه yes و Cancel ایجاد میکند

## تابع در java script

یک تابع مجموعه ای واحد از یک سری دستورالعمل است که در هر بار فراخوانی کل دستورات درون آن یکبار اجرا می شود. از توابع برای شکستن و تقسیم کردن کل برنامه به واحدهای کوچک تر و مستقل استفاده میشود.

**تابع با function تعریف میشود.**

**مثال:**

```
<script type="text/javascript">
Function plus()
{
Var x = 1;
Var y = 2;
Var z = x + y;
}
</script>
```

در این تابع مقدار x و y باهم جمع میشوند و در متغیر z ریخته میشوند.

هر موقع نام تابع فراخوانی شد این جمع صورت میگیرد.

برای نمایش متن از شیء document و متد write استفاده میشود.

```
Document.write("hello");
```

## Event یا رویداد:

ایونت ها اتفاقاتی هستند که روی عناصر اعمال میشود.

یکی از قابلیت های آن، واکنش های تعاملی آن متناسب با رفتار کاربر مانند: کلیک کردن روی دکمه است.

برخی ایونت ها در جدول آمده است.

کارایی	Event
تغییر مقادیر یکی از فیلدهای نمون برگ در صفحات وب	onchange
کلیک ماوس روی یک عنصر	onclick

تمرین : میخواهیم توسط یک کلید نام سه نفر را چاپ کند . ( با استفاده از تابع )

۱	<script type="text/javascript">
۲	function showname(){
۳	document.write("ali"+" ");
۴	document.write("mohammad");
۵	document.write("abdollah");}
۶	</script>
7	<input type="button" name="button" value="show name" onclick="showname()" />

## توضیحات کد.

2. تعریف تابع با نام name

3. چاپ نام ali

4. چاپ نام mohammad

5. چاپ نام abdolah

7. تعریف کلید و استفاده از ایونت onclick و فراخوانی تابع name

## دسترسی java script به داده ها :

زمانی پیش می آید که شما نیاز دارید یک سری محاسبات یا تصمیمات را با توجه به **ورودی** انجام دهید. برای این کار شما باید بتوانید داده های ورودی را به java script بشناسانید.

برای دسترسی به ورودی ها از شی document استفاده میکنیم.

از دو روش میتوان استفاده کرد.

۱. دسترسی با استفاده از id

۲. دسترسی با استفاده از name

## دسترسی با استفاده از id

متد `getElementById` برای دسترسی به عنصر توسط `id` عنصر است.

مثال:

```
document.getElementById("id");
```

## دسترسی با استفاده از name

متد `getElementByName` برای دسترسی به عنصر توسط `name` عنصر است.

```
document.getElementsByName("name");
```

تمرین می‌خواهیم از کابر دو عدد بگیریم و حاصل جمع را نمایش دهیم.

```
<script type="text/javascript">
function plus(){
    var a =
    Number(document.getElementById("1").value);
    var b =
    Number(document.getElementById("2").value);
    document.getElementById("3").value=a+b;
}
</script>
<label>number 1: </label><input type="text"
name="number1" id="1"><br><br>

<label>number 2: </label><input type="text"
name="number2" id="2"><br><br>

<label>show box </label><input type="text"
name="show box" id="3"><br><br>

<input type="button" name="button" value="show"
onclick="plus()"/><br>
```

## عبارات

عبارات در کل به دو دسته تقسیم میشوند

۱. **عبارات ریاضی** : خروجی یک عبارت ریاضی همیشه یک مقدار عددی است .  
مثال:

$$2+3$$

$$4-3$$

$$2*3/(4-7)$$

**عبارات ریاضی از دو قسمت تشکیل شده اند یکی اعداد(عملوندها) یکی علامت های ریاضی(عملگرها)**  
**عملگر های ریاضی شامل:**

**+** (جمع)

**-** (تفریق)

**\*** (ضرب)

**/** (تقسیم)

**%** (باقیمانده تقسیم)

**++** (افزایش)

**--** (کاهش)

**\*\*** (توان)



## تفاوت تقسیم و باقیمانده تقسیم:

تقسیم تا جایی عبارت را ادامه میدهد تا باقیمانده تقسیم صفر شود  
سپس خارج قسمت را نمایش میدهد

۵	۴	
۴	۱/۲۵	خارج قسمت
۱۰		
۸		
۲۰		
۲۰		
۰		باقیمانده

باقیمانده تقسیم : مقدار باقیمانده تقسیم را نشان میدهد.

۵	۴
۴	۱
۱	

## اولویت عملگرها

( )

\*\*

% / \*

- +

۲. **عبارات منطقی** : خروجی یک عبارت منطقی همیشه یک مقدار منطقی است.

یک عبارت منطقی از یک سری عدد که بهشون می‌گیم عملوند و عملگر منطقی تشکیل شده اند .

منطق همیشه می‌گوید یا چیزی درسته است یا غلط.

در زبان برنامه نویسی به :

درست = true

غلط = false

می‌گوییم.

عملگر های منطقی:

> (بزرگتر)

< (کوچکتر)

>= (بزرگتر مساوی)

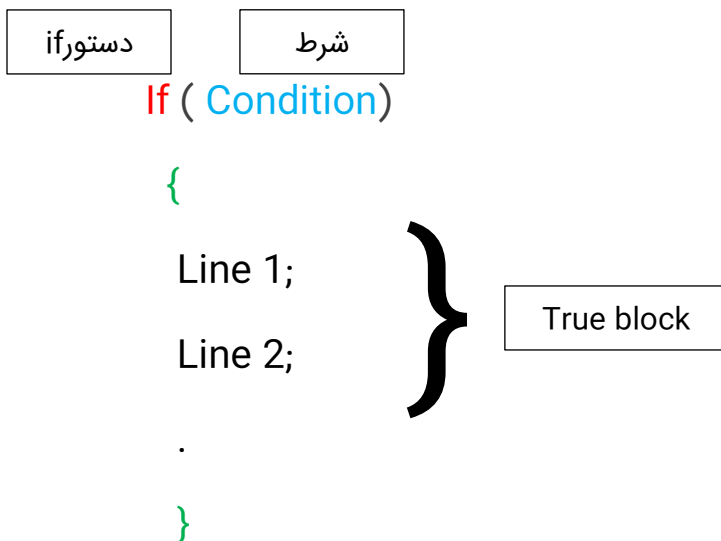
<= (کوچکتر مساوی)

!= (نامساوی)

== (مساوی)

### جملات شرطی:

برای استفاده از جملات شرطی از دستور if استفاده میشود :



در قسمت Condition شما باید یک عبارت وارد کنید (منطقی یا ریاضی) چون در نهایت زبان برنامه نویسی بعد از این که عبارت شما را انجام داد ان را تبدیل به عبارت منطقی میکند.

سه نوع عملگر منطقی دیگر: این نوع عملگر ها اولویت پایین تری نسبت به سایر عملگر ها دارد

AND = بین دو عبارت منطقی استفاده میشود (&&)

مثال:

a AND b =

false AND false = false

false AND true = false

true AND false = false

true AND true = true

OR = بین دو عبارت منطقی استفاده میشود (||)

NOT = روی یک مقدار منطقی اعمال میشود (!)

مثال

A NOT =

True NOT = false

False NOT = true

## حلقه ها:

گاهی اوقات پیش می آید که یک کار را چند بار تکرار کنیم.

حلقه **for** یکی از معروف ترین و پرستفاده ترین حلقه های تکرار در اکثر زبان های برنامه نویسی می باشد، که به راحتی با استفاده از این حلقه میتونیم یک تکه کد رو بارها و بارها اجرا کنیم و هربار یه مقدار متفاوت بدست آوریم.

یعنی بطور کلی این حلقه زمانی بکار ما میاد، که بخوایم تکه کدی رو چندین بار بصورت همزمان اجرا کنیم و هربار یه مقدار متفاوت بدست بیاوریم.

فرض میکنیم میخوایم ، عدد ۰ تا ۱۰۰ رو در خروجی چاپ کنیم، خب بنظره شما چکار باید کرد؟! خب یه روش اینکه بیایم و دونه به دونه از عدد ۰ تا ۱۰۰ رو در خروجی چاپ کنیم ، همانند دستورات زیر:

```
<script>
```

```
document.write("0" + "<br>");  
document.write("1" + "<br>");  
document.write("2" + "<br>");  
document.write("3" + "<br>");  
.  
.  
document.write("100" + "<br>");
```

```
</script>
```

ولی شما فکرشو کنید، ما باید ۱۰۰ خط کد بنویسیم که هر خط یه عدد چاپ کنه ولی خب این اصلا بهینه و استاندارد نیست، به این حالت از برنامه نویسی میگن برنامه نویسی **کثیف**.

ولی الان همین برنامه رو با حلقه For میخوایم بنویسیم :

```
1 <script>
2
3 for( i = 0 ; i < 101 ; i++){
4 document.write( i + "<br>" );
5 }
6
7 </script>
```

### شکل کلی حلقه For در جاوا اسکریپت

حلقه For حلقه ای ایست که با استفاده از اون میتونیم یک تکه کد رو چندین بار بصورت همزمان اجرا کنیم و هر بار یک مقدار متفاوت بدست آوریم.

شکل کلی حلقه for رو میتونیم بصورت زیر نمایش بدیم:

```
for ( Bakhsh_1 ; Bakhsh_2 ; Bakhsh_3 ) {
    // قطعه کدی که قرار است تکرار شود
}
```

اول کلمه کلیدی for رو مینویسیم، بعدش یه پرانتز باز میکنیم و داخل پرانتز باید ۳ بخش رو مشخص کنیم :

```
for( i = 0 ; i < 101 ; i++ ){  
document.write( i + "<br>" );  
}
```

**بخش اول** «در این بخش میتونیم متغیرهامون رو تعریف کنیم و مقدار هم بهشون بدیم، در واقع متغیرهایی که قراره تحت تاثیر اجرای حلقه تکرار قرار بگیرند.

**بخش دوم** «در این بخش باید شرط مون رو قرار بدیم، یعنی شرطی که اگه برقرار بود و درست بود میره و حلقه رو اجرا میکنه و اگه شرط درست نبود خب هیچی و حلقه اجرا نمیشه.

**بخش سوم** «اگه شرط درست بود، هرچیزی که در این بخش قرار داشت ، با هربار اجرای حلقه اینم اجرا میشه. ( معمولا در بخش ۳ افزایش مقدار متغیر موجود در بخش ۱ رو اعمال میکنن )

## حلقه while :

حلقه while یکی دیگه از **حلقه های تکرار** یا Loop در زبان جاوا اسکریپت می باشد، که با استفاده از این حلقه میتونیم مشخص کنیم تا زمانی که یه شرط درست باشد حلقه اجرا بشه.

بخوایم کلی حساب کنیم حلقه while دقیقا همون حلقه for می باشد، یعنی عملکردشون شبیه همه ولی خب تنها تفاوتی که دارن در نحوه استفاده شونه مثلا:

- بفرض مثال در حلقه FOR ما ۳ بخش داریم ولی در حلقه While فقط یه بخش اصلی داره اونم شرطمون هست

```
// حلقه for
for ( i=1 ; i<10 ; i++ ) {
...
}
```

```
// حلقه while
while ( i < 10 ) {
...
}
```

این حلقه فقط تا زمانی که یه شرط درست باشه اجرا میشود

حواستون باشه که اگه مثلا میخواستید اعداد ۱ تا هر عددی رو در خروجی چاپ کنه، باید حتما متغیر رو در درون حلقه افزایش بدید. مثه  $i++$  وگرنه حلقه بی نهایت میشه و مرورگرتون هنگ میکنه و در نهایت مجبور میشید سیستم رو ریستارت کنید.

```
<script>
var i = 1;
while( i < 11 ){
document.write( i + "<br>" );
i++;
}
</script>
```



## متد replace

متد replace از متد های شی location است. که ادرس یک صفحه را به صورت یک پارامتر از کاربر دریافت سپس صفحه جاری را به آن صفحه انتقال میدهد.

مثال:

```
function loc(){  
    location.replace("https://google.com")  
}
```

پایان مبحث javascript مقدماتی



## php مقدماتی:

قبل از شروع برنامه نویسی نیاز دارید با دو مفهوم **سرویس گیرنده** و **سرویس دهنده** آشنا شوید.

وقتی شما قصد خرید از فروشگاه را دارید، در این تعامل شما که تقاضای خرید دارید **سرویس گیرنده** هستید و فروشنده فروشگاه، **سرویس دهنده** شما است.

در دنیای شبکه های رایانه ای به سیستمی که سرویس ارائه میکند سرویس دهنده (**Server**) و سیستم هایی که سرویس دریافت میکنند سرویس گیرنده (**Client**) میگویند.

زبان های برنامه نویسی مبتنی بر وب به دو دسته کلی تقسیم می شوند و بر اساس اینکه زبان برنامه نویسی در طرف سرویس دهنده یا سرویس گیرنده کارایی دارد، به آنها زبان برنامه نویسی سمت سرویس دهنده یا سمت سرویس گیرنده می گوییم.

برای ساخت صفحات وب پویا نیاز به برنامه نویسی سمت سرور داریم تا در خواست کاربر را پردازش و پاسخ مناسب را ارسال کند.

برای کار هایی مانند دریافت، ذخیره، نمایش داده ها از پایگاه داده و تعامل با کاربر در سمت سرویس انجام میشود.

زبان های سمت سرور مانند Perl, PHP و ... هستند.

سرویس دهنده وب (web server) مجموعه نرم افزار هایی که توانایی دریافت در خواست از مرورگر و ارسال پاسخ به آن دارد.

رایج ترین وب سرور ها را IIS , Apache رامیتوان معرفی کرد.

## IIS (Internet Information Services server)

(سرور خدمات اطلاعات اینترنتی)

وب سروری است که برای نشر و توزیع سریع محتویات مبتنی بر وب در سیستم عامل های ویندوز، ب هوسیله شرکت مایکروسافت طراحی و ارائه شده است. این وب سرور تجاری است و باید برای اجرای همه محصولات مبتنی بر وب که ب هوسیله ویژوال استودیو تولید شده اند، فعال شود.

وب سرور Apache با توجه به Open Source بودن. رایگان است و در اختیار همه برنامه نویسان قرار دارد و روی همه سیستم عامل ها قابل اجرا هستند.

برای استفاده از وب سرور مجازی Apache . از نرم افزار WAMP که مجموعه ای از نرم افزار های php , mysql , apache , windows در خود دارد استفاده میکنیم.

## php چیست؟

PHP یک زبان **تفسیرشونده** است که اساس و بنیان آن طراحی و ایجاد صفحات پویای وب بوده است. به عبارتی دیگر، کسانی که وظیفه کدنویسی سایت را بر عهده دارند با استفاده از زبان PHP قادر خواهند بود وب اپلیکیشن‌هایی طراحی کنند که به صورت تعاملی می‌باشند. منظور از تعاملی بودن وجود قابلیت‌هایی مانند سطوح دسترسی مختلف برای کاربران، امکان ثبت‌نام در سایت، پنل کاربری، ذخیره اطلاعات کاربران و بسیاری از امکانات دیگر است که امروزه در تمامی سایت‌های امروزی مشاهده می‌کنیم.

## برنامه‌های تفسیر شونده یا کامپایلری چی هستند؟ و تفاوت‌های آنها در چیست؟

وقتی صحبت از زبان‌های برنامه‌نویسی مختلف می‌شود، اساساً می‌توان آن‌ها را به دو دسته کلی تقسیم نمود: آن‌هایی که **کامپایل** می‌شوند و آن‌هایی که **اینترپرت** می‌گردند به طوری که از جمله زبان‌هایی که به دسته اول تعلق دارند می‌توان جاوا یا سی‌شارپ را مثال زد و از جمله زبان‌هایی که تفسیر می‌شوند هم می‌توان به **پای‌پی** و جاوااسکریپت اشاره کرد.

## Compiler چیست؟

Compiler یک برنامه کامپیوتری است که کدهای یک زبان برنامه‌نویسی سطح بالا را به کدی خوانا برای ماشین تبدیل می‌کند. به عبارتی، برنامه‌ای است که کدهای قابل خواندن توسط انسان را به زبانی که پردازنده‌های کامپیوتر قادر به درک آن باشند (یعنی کدهای باینری یا همان صفر و یک) تبدیل می‌کند.

## Interpreter چیست؟

Interpreter (مفسر) یک برنامه کامپیوتری است که هر خط از دستورات یک زبان سطح بالا را به کد ماشین تبدیل می‌کند. Compiler و Interpreter هر دو کاری یکسان، یعنی تبدیل کدهای زبان سطح بالا به کد ماشین، انجام می‌دهند اما **کامپایلر کد را پیش از اجرای برنامه به کد ماشین تبدیل می‌کند** (یعنی یک فایل اجرایی همچون exe می‌سازد) **در حالی که مفسر کد را حین اجرا به کد ماشین تبدیل می‌کند**.

## دید کلی در مورد کد نویسی php

ابتدایی ترین کد نویسی هر زبان برنامه نویسی چاپ کردن "hello world" است.

کد زیر با استفاده از زبان PHP که درون کدهای HTML قرار گرفته، نوشته شده است

```
<!DOCTYPE html>

<meta charset="utf-8">

<title>print hello world</title>

<?php

echo 'Hello World';

?>
```

صفحات وب که در آنها از کد php استفاده شده است باید با پسوند php و در پوشه ریشه نرم افزار WAMP

(پوشه WWW) ذخیره شوند تا به وسیله Apache قابل شناسایی، دسترسی و پردازش باشند.

مسیر پوشه: C:\WAMP\WWW

با ذخیره فایل PHP در این پوشه در واقع صفحه را روی **localhost** قرار میدهیم.

**Localhost** یا **میزبان (host)**: همه ی وب سایت ها نیاز به یک فضای میزبانی در اینترنت دارند و سایت در آن ذخیره شود. در واقع یک هارد دیسک است که با اینترنت قابل دسترسی هستند. این حافظه ها رایگان نیستند و باید در مدتی معین حق اجاره ای پرداخت شود.

**Hosting Server** یا **سرویس دهنده میزبانی**: سرویس دهنده ای که خدمات میزبانی ارائه میدهد.

پوشه **www** همان فضای میزبان است. برای دسترسی به این صفحه در مرورگر عبارت **localhost** یا معادل IP آن یعنی **127.0.0.1** را وارد میکنیم.

## دستور پرینت یا خروجی echo

وضایف:

۱. نمایش محتوای متغیرها

۲. رشته‌ها

شکل کلی دستور echo

(نام متغیر یا رشته ثابت) echo;

دستور echo بدون پرانتز یا با پرانتز تعریف می‌شود. در نتیجه این دستور را به صورت `echo` و `echo()` در اسکریپت‌های مختلف می‌بینیم.

حالا کار را کمی پیچیده‌تر می‌کنیم.

در رشته متنی می‌توانیم هر چیزی را قرار دهیم؛ چه چیز بهتر و کاربردی‌تر از تگ‌های HTML! در قطعه کد زیر یک متن را به صورت بولد شده با تگ `strong` در خروجی نمایش داده‌ایم.

```
echo "<strong>test tag html.</strong>";
```

دستور echo در PHP می‌تواند به تعداد نامحدودی ورودی بگیرد. بنابراین اگر لازم بود چندین متن یا متغیر را پشت سر هم در خروجی چاپ کنیم، می‌توانیم با علامت کاما (,) این چند ورودی را از هم جدا کنیم.

```
echo "ali", "Nazanin", "abdollah", "hashemi";
```

## دستور print در PHP

دستور پرینت در PHP نیز مشابه دستور echo استفاده می‌شود. از دستور print می‌توان برای چاپ متن یا متغیر استفاده کرد.

برای استفاده از دستور پرینت می‌توان از نحو (syntax) با پرانتز یا بدون پرانتز استفاده کرد.

```
print "<h5>Hello!</h5>";
```

## تفاوت echo و print در PHP

این دو دستور تفاوت زیادی با هم ندارند. اصلی‌ترین تفاوت print و echo در پی‌اچ‌پی این است که تابع print پس از اجرا مقدار ۱ برمی‌گرداند.

وقتی از دستور echo برای چاپ متن در PHP استفاده می‌کنیم، مقدار مورد نظر ما در خروجی برنامه پرینت شده و برنامه به خط بعدی منتقل می‌شود. به اصطلاح این دستور هیچ مقدار بازگشتی (returned value) ندارد.

اما وقتی به دستور print می‌رسیم، اگر عملیات پرینت محتوا به درستی انجام شود، به عنوان خروجی این دستور، عدد ۱ را خواهیم داشت.

از این ویژگی زمانی استفاده می‌شود که می‌خواهیم از دستور پرینت در ساختارهای شرط در PHP استفاده کنیم.

## متغیرها در php

متغیرها در زبان php با \$ شروع میشوند و قوانین نام گذاری آنها مشابه زبان C# است.

زبان php در تعریف متغیر نسبت به حروف بزرگ و کوچک حساس است پس \$x با \$X تفاوت میکند.

نوع داده	مثال
اعداد صحیح	\$n=25;
اعداد اعشاری	\$pi=3.14;
رشته‌ها	\$city="kerman";
منطقی	\$pass=false;
آرایه‌ها	\$ar=array("pos","zero","neg");

در زبان php نوع متغیر با توجه به نوع محتوای آن مشخص میشود و نیاز به تعیین نوع داده برای متغیر مانند سایر زبان برنامه نویسی نیست.



## مقدار دهی و نمایش متغیر ها :

```
<?php
```

```
$name = "abdola";  
$site = "site.com";
```

```
print "<h5>Hello " . $name . "!</h5>";  
print welcome to site " . $site . "<br>";
```

```
echo "<h5>Hello " . $name . "!</h5>";  
echo "welcome to site " . $site . "<br>";
```

تحقیق : رشته ها بین تک کوتیشن و دابل کوتیشن قرار میگیرند ، تفاوت انها را به کلاس بیاورید.

کاربرد نقطه (.) در بین رشته ها را بیابید

مثال :

```
echo("hi" . "class" . "st11");
```

## آرایه ها:

در زبان برنامه نویسی PHP منظور از آرایه، ساختمان داده ای است که امکان ذخیره سازی عناصر چندگانه در یک متغیر منفرد را ارائه می کند. این عناصر به صورت جفت های **کلید-مقدار** ذخیره می شوند.

## ساختار کلی آرایه:

آرایه ها به دو دسته **اندیس دار (Indexed)** و **انجمنی دار (Association)** تقسیم میشوند.

به صورت پیشفرض اندیس های هر آرایه از شماره ۰ شروع میشوند:

n	۵	۴	۳	۲	۱	۰	اندیس	\$AR=array
...	Yosuf	reza	karim	hashemi	abass	ali	مقدار	

برای نمونه فرض کنید می خواهید نام چند میوه را ذخیره کنید. بدون وجود آرایه می بایست چند متغیر مستقل برای ذخیره نام میوه های مختلف ایجاد کنید. در سوی دیگر اگر از آرایه برای ذخیره نام های میوه ها استفاده کنید، چیزی شبیه زیر به دست می آورید:

```
$array_fruits = array('sib', 'moz', 'gilas', 'anbe');
```

## شیوه مقداردهی اولیه آرایه

زمانی که می خواهیم آرایه ای را مقداردهی اولیه کنیم، چند روش مختلف برای این منظور وجود دارد. در اغلب موارد این سازه `array()` در زبان برنامه نویسی است که برای مقداردهی اولیه یک آرایه استفاده می شود.

```
$array = array();
```

در قطعه کد فوق متغیر `$array` با یک آرایه خالی مقداردهی اولیه می شود. از نسخه ۵/۴ PHP به بعد شما می توانید از ساختار زیر برای مقداردهی اولیه PHP استفاده کنید:

```
$array = [];
```

## اضافه کردن عناصر به آرایه:

```
$array = [ ];  
$array[ ] = 'One';  
$array[ ] = 'Two';  
$array[ ] = 'Three';  
echo '<pre>';  
print_r($array);
```

قطعه کد فوق خروجی زیر را تولید می‌کند:

```
Array  
(  
    [0] => One  
    [1] => Two  
    [2] => Three  
)
```

نکته مهمی که باید اشاره کرد این است که اندیس آرایه از ۰ شروع می‌شود. هر زمان که عنصر جدیدی بدون مشخص ساختن اندیس، به یک آرایه اضافه شود، آرایه آن را به طور خودکار به اولین اندیس خالی انتساب می‌دهد.

البته شما می‌توانید آرایه‌ای از مقادیر قبلاً مقداردهی شده ایجاد کنید. این وضعیت خلاصه‌ترین روش برای ایجاد آرایه‌هایی است که مقادیر آن‌ها از قبل مشخص است.

```
$array = ['One', 'Two', 'Three'];
```

## دسترسی به عناصر آرایه:

ولین روش بدیهی برای دسترسی به عناصر آرایه، واکشی آنها از طریق کلید یا اندیس آرایه است:

```
$array = ['One', 'Two', 'Three'];
```

```
echo $array[0];  
echo "<br>";
```

```
echo $array[1];  
echo "<br>";
```

```
echo $array[2];  
echo "<br>";
```

## آرایه‌های انجمنی (Associative)

آرایه انجمنی مشابه آرایه اندیس‌دار است؛ تنها تفاوت این است که از مقادیر رشته‌ای برای

اندیس‌های آرایه استفاده می‌شود. در کد زیر روش تعریف یک آرایه انجمنی را می‌بینید:

```
$employee = ['name' => 'abass', 'email' => 'abass@example.com',  
            'phone' => '1234567890'];
```

به طور جایگزین می‌توانید از ساختار زیر نیز استفاده کنید.

```
$employee = [];  
$employee['name'] = 'hashemi';  
$employee['email'] = 'hashemi@example.com';  
$employee['phone'] = '1234567890';
```

```
$score=array("ali"=>18, "omid"=>17, "neda"=>19);
```

```
$score['reza']= 18;
```

```
$score['masoud']= 17;
```

```
$score['neda']= 20;
```

## شمارش تعداد عناصر آرایه:

### تابع count

تابع count برای شمارش تعداد عناصر موجود در یک آرایه استفاده می‌شود. این وضعیت در اغلب مواردی که می‌خواهید حلقه‌ای به صورت for روی یک آرایه تعریف کنید مورد نیاز است.

```
$array = ['One', 'Two', 'Three'];  
echo count($array);
```

## ثابت ها :

ثابت در واقع یک شناسه یا یک نام برای یک مقدار ساده است. این مقدار در طول اجرای برنامه یا همان اسکریپت ثابت بوده و تغییر نمی‌کند.

## تعریف ثابت ها در PHP

برای تعریف ثابت در php از تابع define به صورت زیر استفاده می کنید .

```
define('pi',3.14);
```

```
echo pi;
```

```
define('pi',3.14);
```

برنامه ای بنویسید که مساحت دایره را حساب کند: شعاع \* شعاع \* عدد پی

```
define("PI",3.14);
```

```
$r=8.5;
```

```
$area= PI*$r*$r;
```

```
echo("Circle Area : " . $area );
```

## دستورات شرطی:

در زبان PHP از کلمه کلیدی **if** برای شروع شرط استفاده می شود. این کلمه در انگلیسی به معنای **<اگر>** است.

فرض کنید متغیری به نام `$nomre` داریم. می خواهیم اگر نمره کاربر بیشتر از **10** بود، به او پیام تبریک قبولی بدهیم.

```
$score = 14;
```

```
if ($score > 10) {  
    echo "You Pass the Exam!";  
}
```

## دستور switch:

برای بررسی حالت های مختلف یک عبارت از switch استفاده میکنیم:

دستور switch که به نوعی می توان آن را جایگزین چندین دستور if elseif در نظر گرفت. به طور کلی، ساختار دستورات سوئیچ در زبان برنامه نویسی PHP به صورت زیر است:

```
switch (Value) {  
    case First value:  
        break;  
    case Second value:  
        break;  
    case Third value:  
        break;  
    default:
```

همان طور که از ساختار دستور سوئیچ مشخص است، متغیر یا شرطی را داخل پرانتزهای مرتبط با switch قرار داده سپس در چندین case مختلف می سنجیم ببینیم که آیا متغیر/شرط ما با کدام یک از کیس ها برابری می کند و در نهایت اگر با هیچ کدام از کیس ها هم خوانی نداشت، دستورات داخل بلوک default اجرا خواهند شد. برای روشن تر شدن نحوه استفاده از دستور سوئیچ، ابتدا مثالی با دستور if می آوریم:



```
<?php
$i = 2;
if ($i == 0) {
    echo "i equals 0";
} elseif ($i == 1) {
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
} else {
    echo "none of them";
}
```

متغیری تعریف کرده‌ایم تحت عنوان  $i$  که مقدار اولیه ۲ را هم برایش در نظر گرفته‌ایم. حال با یکسری دستورات if و elseif سنجیده‌ایم که مقدار این متغیر با کدام یک از شروط برابری می‌کند.

اکنون می‌خواهیم دستورات شرطی فوق را با switch ریفتور (بازنویسی) کنیم:

```
<?php
$i = 2;
switch ($i) {
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
    default:
        echo "none of them";
}
```

ابتدا متغیری تعریف کرده‌ایم تحت عنوان  $i$  و مقدار پیش‌فرض ۲ را هم برایش در نظر گرفته‌ایم؛

سپس این متغیر را به عنوان پارامتر ورودی دستور switch در نظر گرفته‌ایم و از سه کیس مختلف

برای چک کردن مقدار این متغیر استفاده کرده‌ایم.

در کیس اول، کلیدواژه case را نوشته، یک فاصله قرار داده و شرط مد نظر را نوشته‌ایم و پس از شرط

هم یک علامت : قرار داده‌ایم (لازم به ذکر است که در کیس اول، شرط برابر بودن مقدار متغیر  $i$  با

عدد ۰ است).

نکته علامت : اصطلاحاً Colon (کالِن) و علامت ; اصطلاحاً Semicolon (سیمِ کالِن) گفته می‌شود.  
به خاطر داشته باشیم که پس از کلیدواژه case، به جای علامت : می‌توان از علامت ; نیز استفاده کرد:

پس از هر دستور case، بایستی دستوراتی که مد نظرمان هست را بنویسیم. مثلاً چنانچه مقدار متغیر  $i$  برابر با ۰ باشد، دستور داده‌ایم که با استفاده از echo استرینگ 0 equals i چاپ شود.

نکته مهمی که در اینجا وجود دارد، استفاده درست از دستور break است. این دستور -همان‌طور که از نامش مشخص است- این وظیفه را دارا است تا روند اجرای برنامه را متوقف سازد. به عبارت دیگر، چنانچه شرط در نظر گرفته شده در اولین case برآورده شود، دستورات قرار گرفته داخل این بلوک اجرا شده سپس دستور break اجرا می‌شود که این وظیفه را دارا است تا ما را از داخل switch خارج ساخته و ادامه برنامه اجرا گردد.

به همین منوال، مقادیر ۱ و ۲ را هم چک می‌کنیم تا در نهایت به دستور default می‌رسیم که کاربردی همچون دستور else در دستورات شرطی if دارد. به عبارت دیگر، چنانچه هیچ کدام از کیس‌ها اصطلاحاً true نباشند، مفسر وارد این بلوک می‌شود (توجه داشته باشید که اگر هر تعداد case پس از default قرار گیرد، هرگز چک نخواهند شد)

پیش از این گفتیم که کاربرد دستور break در سوئیچ‌ها بسیار حائز اهمیت است. برای روشن‌تر شدن این مسئله، کدهای فوق را به صورت زیر تغییر می‌دهیم:

```
<?php
$i = 0;
switch ($i) {
    case 0:
        echo "i equals 0<br>";
        // break;

    case 1:
        echo "i equals 1<br>";
        // break;

    case 2:
        echo "i equals 2<br>";
        // break;

    default:
        echo "none of them";
}
```

همان‌طور که می‌بینیم، ابتدا مقدار پیش‌فرض متغیر  $i$  را برابر با ۰ قرار داده، سپس کلیه دستورات break را کامنت کرده‌ایم (همچنین از تگ `<br>` برای خواناتر شدن خروجی در داخل استرینگ‌ها استفاده کرده‌ایم). حال اقدام به اجرای این اسکریپت می‌کنیم:

می‌بینیم علیرغم اینکه انتظار می‌رود صرفاً case اول اجرا گردد، اما تمامی کیس‌ها و حتی بلوک default نیز اجرا شده است و اینجا است که بایستی به اهمیت کاربرد دستور break پی ببریم. در واقع، کاری که break انجام می‌دهد این است که به محض برآورده شدن شرط، جلوی اجرای ادامه دستورات قرار گرفته داخل سوئیچ را می‌گیرد.

همچنین به خاطر داشته باشیم که به جای دستور break از معادلی همچون continue (که بیشتر در حلقه‌ها استفاده می‌شود) نیز می‌توان استفاده کرد؛

### آشنایی با حلقه foreach در زبان PHP

در زبان PHP ساختاری تحت عنوان foreach وجود دارد که به منظور دستیابی به تمامی اِلمان‌های یک آرایه مورد استفاده قرار می‌گیرد. ساختار کلی foreach به صورت زیر است:

به تعداد عناصر یک آرایه، یک قطعه بلوک کد را اجرا و تکرار می‌کند.

```
foreach ($array as $value)
```

```
{
```

```
}
```

مثال:

تعریف آرایه با مقدار ثابت \$a=['one','two','three'];

```
foreach($a as $b) {  
    echo($b."<br>");  
}
```

تعریف حلقه ابتداع آرایه را به حلقه معرفی سپس با استفاده از کلمه کلیدی as مقادیر آرایه را به یک متغیر انتساب می‌دهیم و در داخل حلقه متغیر ساخته شده را نمایش می‌دهیم.

همچنین توجه داشته باشید که چنانچه تصمیم گرفتید تا هم اندیس‌ها و هم مقادیر را از آرایه مد نظر استخراج کنید، مابین متغیری که قرار است اندیس را در خود ذخیره سازد با متغیری که قرار است مقدار آن اندیس را در خود ذخیره سازد از علامت => استفاده نمایید.

مثال : می‌خواهیم نام موجودیت های یک ارایه همراه با شماره اندیس های آنها را چاپ کنیم.

```
$a=['a','b','c','d'];
```

```
foreach($a as $b=>$c){
    echo "key is =>" . $b . " " . "name is =>" . $c . "<br>";
}
```

### تگ select ساخت منو کشویی:

در html برای ساخت منو کشویی از **select** و برای مقدار دهی به هر یک از اعضای منو از تگ **option** استفاده میشود.

مثال html :

```
<label>:مقطع تحصیلی خود را انتخاب کنید</label>
```

```
<select name="school">
```

```
<option value="10">دهم</option>
```

```
<option value="11">یازدهم</option>
```

```
<option value="12">دوازدهم</option>
```

```
</select>
```

اکنون می‌خواهیم با استفاده از آرایه به هر عنصر در منو مقدار دهی کنیم .

مثال:

1	echo('<select>');	ساخت منو
2	\$a=['10','11','12','13'];	ایجاد ارایه
3	foreach(\$a as \$list){	دستور حلقه foreach
4	echo '<option value='.\$list.'>'.\$list.'</option>';}	تخصیص دادن مقدار داخل ارایه به option و نمایش
5	echo('</select>');	بستن تگ منو

## ارسال فرم از صفحه html و دریافت توسط php

برای این کار باید یک آشنایی قبلی از کاربرد action و method در ایجاد فرم داشته باشیم.

**Action:** اکشن آدرس مقصد ما را مشخص می کند و از نوع URL است. اطلاعات فرم پس از ثبت توسط کاربر به این صفحه هدایت می شوند. مقدار اکشن نمی تواند خالی باشد و باید حتما مشخص شود چون در صورت خالی بودن ایجاد مشکل می کند. در صورتی که مقداری مشخص نکنیم اطلاعات فرم به آدرس پیشفرض form.html هدایت می شود.

**Method:** متد نحوه ارسال اطلاعات را مشخص می کند و دو حالت post و get برای آن وجود دارد. به لحاظ عملکرد این دو حالت مشابه یکدیگرند اما نحوه ارسال در آنها متفاوت است.

## مقایسه GET و POST

در متد GET اطلاعات ارسالی در این روش بصورت کد نشده منتقل می شود. در این روش اطلاعات به آدرس URL اضافه می شود و ارسال می شود. بدیهی است که این اطلاعات در قسمت address bar نیز قابل مشاهده است به همین خاطر برای ارسال اطلاعات حساس مانند پسوردها، نباید از این روش استفاده کرد. در صفحه php توسط آرایه \$\_GET دریافت میشود

نام عنصر = \$\_GET["نام متغیر"];

در متد POST اطلاعات کد شده و بصورت مخفی ارسال می شوند. این روش برای ارسال اطلاعات حساس مناسب است.

در صفحه php توسط آرایه \$\_POST دریافت میشود

نام عنصر = \$\_POST["نام متغیر"];

در این قسمت قصد داریم یک فرم ساده برای آشنایی شما با کار با فرم ها در php ارائه دهیم. در این قطعه کد از متد POST استفاده می کند و دارای سه فیلد برای دریافت اطلاعات می باشد. مقصد این فرم، صفحه index2.php است که پس از ثبت اطلاعات کاربر توسط دکمه submit ، این داده ها در این صفحه برای پردازش های بعدی ارسال می شوند.

### کد های صفحه html

```
<form action="index2.php" method="post">

    <label>enter user name: </label><input type="text"
name="user"><hr>

    <label>enter email: </label><input type="email"
name="email"><hr>

    <label>enter password: </label><input type="password"
name="pass"><hr>

    <input type="submit" value="send">

</form>
```



```
$user=$_POST["user"];

$email=$_POST["email"];

$pass=$_POST["pass"];

echo $user.'<hr>';

echo $email.'<hr>';

echo $pass.'<hr>';
```

### اعتبار سنجی : توابع اعتبار سنجی

نام	شکل کلی	کاربرد
isset( )	isset(\$m )	این تابع خروجی را به شکل یک متغیر بولین باز می گرداند. اگر هر متغیری را درون تابع isset قرار بدهیم؛ نتایج را به شکل true یا false باز می گرداند. اگر متغیری که ما انتقال داده ایم؛ بیان شده باشد و نیز شامل مقدار مشخصی باشد؛ و یا مقداری در این متغیر ذخیره نشده باشد و به اصطلاح یک متغیر نول باشد؛ عبارت True بازگردانده می شود. اما اگر متغیر درون کد نویسی تعریف یا نوشته نشده باشد؛ عبارت false باز گردانده می شود.
empty( )	empty(\$m)	این تابع بررسی می کند که آیا یک متغیر؛ مقداری اتخاذ کرده است یا خیر. در صورت خالی بودن خروجی این تابع true میباشد
exit( )	exit(پیام دلخواه)	برای خاتمه دادن به دستورات ورودی این تابع میتواند یک پیام باشد اما همیشه الزامی نیست
filter_var( )	filter_var ( عبارت یا الگو، نام متغیر )	برای بررسی صحت رایانامه میتوان بجای عبارت الگو از FILTER_VALIDATE_EMAIL استفاده کرد

بعضی از عملگرهای زبان php با سایر زبان ها تفاوت دارد

عملگر	توضیحات	برای مقایسه $m=1, n=2$	خروجی
===	بررسی برابری مقدار دو متغیر	$m === n$	false
!==	بررسی نابرابری متغیرها	$m !== n$	?

## مثال isset

```
$a = 5;
if (isset($a)) {
    echo "True";
}
else{
    echo "False";
}
```

خروجی این برنامه چیست؟

## مثال empty

```
متغیر دکمه ارسال if(isset($_GET['sub']))
{
    متغیر کادر تکست if(empty($_GET['n']))
    {
        echo "fill your name first";
    }
    else
    {
        echo "welcome ".$_GET['n'];
    }
}
```

## مثال exit

```
If ($pass != $pass2)
    exit("رمز و تکرار رمز مطابقت ندارد");
```

## مثال filter

```
If(filter_var($mail, FILTER_VALIDATE_EMAIL)===false)
    exit("ایمیل وارد شده صحیح نیست");
```

## پایان مبحث php بخش اول

## فصل دوم مبانی php (پایگاه داده ها)

### پایگاه داده چیست ؟

بانک اطلاعاتی یا همان پایگاه داده دیتابیس Database | مجموعه‌ای سازمان‌یافته از داده‌ها است. پایگاه‌های داده از ذخیره‌سازی الکترونیکی و ایجاد تغییر در داده‌ها پشتیبانی می‌کنند. مدیریت داده‌ها به وسیله پایگاه داده بسیار آسان می‌شود. برای مدیریت داده‌ها در یک بانک اطلاعاتی از سیستم مدیریت پایگاه داده (Database Management System) یا همان DBMS استفاده می‌شود.





### داده در پایگاه داده چیست ؟

در علوم کامپیوتر، داده (Data) مجموعه‌ای از اطلاعات است که به قالبی بهینه و مناسب برای انتقال یا پردازش تبدیل شده‌اند. در حقیقت، امکان استفاده از داده‌ها به صورت‌های مختلفی از جمله متن، عدد، رسانه، بایت و سایر موارد وجود دارد. واژه Data از کلمه Datum برگرفته شده که به معنی «یک تکه اطلاعات» است.

### پایگاه داده به چه معنا است؟

پایگاه داده یا همان دیتابیس (بانک اطلاعاتی) به صورت لغوی به معنی مجموعه داده‌ها شناخته می‌شود. به بیان ساده، پایگاه‌های داده یک مجموعه از داده‌ها هستند که در کامپیوتر ذخیره می‌شوند و به نوعی ساختاردهی شده‌اند که امکان دسترسی به داده‌ها از طریق پایگاه داده برای کاربران تسهیل پیدا کند. پایگاه داده امکان تغییر (Manipulation) داده‌ها و ذخیره‌سازی الکترونیکی آن‌ها را پشتیبانی می‌کند؛ به طوری می‌توان با استفاده از بانک اطلاعاتی، داده‌ها را در فایل‌ها یا جدول‌ها ساماندهی کرد. سطرها، ستون‌ها و اندیس‌های جدول، منجر به ساده‌سازی دو عمل جستجو و تعریف پایگاه داده می‌شوند. با توجه به اینکه پایگاه داده به منظور مدیریت، بازیابی و رسیدگی به داده‌ها به صورت بلادرنگ مورد استفاده قرار می‌گیرد، بنابراین، به طور وسیعی از پایگاه داده‌ها در وب سایت‌های متعدد و سایر بسترها و کاربردهای نرم‌افزاری استفاده می‌شود.

در دنیای فناوری اطلاعات شرکت های مختلفی برای مدیریت داده ها، به تولید و توسعه سرویس دهنده های منبع داده اقدام کرده اند که میتوان به محصولاتی مانند MySQL، ORACLE، Microsoft SQL Server، Microsoft Access اشاره کرد.

	<b>Microsoft Access</b>
	<b>Microsoft SQL Server</b>
	<b>ORACLE</b>
	<b>MySQL</b>

## SQL چیست؟

SQL مخفف عبارت **Structured Query Language** است که آنرا “زبان پرس و جوی ساختار یافته” معنی می کنند.

اس کیو ال یک زبان استاندارد برای دسترسی و کار با پایگاه داده (database) است. این زبان از سال ۱۹۸۷ یک زبان استاندارد بین المللی (ISO) بوده است.

مهم ترین دلایلی که میتوان نام برد برای انتخاب MySQL عبارت است از:

۱. رایگان بودن
۲. هزینه حفظ و نگهداری بسیار پایین
۳. سهولت پیکربندی اولیه و داشتن محیط بسیار ساده و کاربرپسند
۴. سازگار با سیستم عامل های مختلف
۵. در دسترس بودن کد اصلی (متن باز) برای توسعه ملی یا حتی خصوصی در سازمان ها
۶. توانایی کار کردن هم زمان با سایر سرویس دهنده های پایگاه داده تجاری

### ساخت پایگاه داده در WAMP :

نرم افزار PHP my admin را در WAMP اجرا کنید.

در سربرگ SQL کد کوئری زیر را وارد کنید:

```
CREATE DATABASE school;
```

CREATE	DATABASE	school
دستور ساخت	دستور نوع (دیتابیس)	نام دیتابیس

نوع داده	ماهیت	اندازه	توضیحات
<b>VARCHAR(n)</b>	رشته‌ای	وابسته به تعداد نویسه‌های تعریف شده	n تعداد نویسه‌ها را مشخص می‌کند که حداکثر ۲۵۵ نویسه است. توجه: اگر رشته‌ای با طول بیش از ۲۵۵ نویسه را در آن قرار دهید، به نوع TEXT تبدیل خواهد شد.
<b>TEXT</b>	رشته‌ای	وابسته به تعداد نویسه‌های تعریف شده	رشته‌ای با طول حداکثر ۶۵۵۳۵ نویسه
<b>INT</b>	صحیح	۴ بایت	محدوده اعداد مجاز به صورت علامت‌دار: ۲۱۴۷۴۸۳۶۴۸ تا ۲۱۴۷۴۸۳۶۴۷ محدوده اعداد مجاز به صورت بدون علامت: ۰ تا ۴۲۹۴۹۶۷۲۹۵
<b>FLOAT(size,d)</b>	اعشاری	۴ یا ۸ بایت	از این نوع داده برای ذخیره اعداد اعشاری با ممیز شناور استفاده می‌شود. size حداکثر تعداد ارقام و d حداکثر تعداد ارقام سمت راست ممیز اعشار را مشخص می‌کند.
<b>DATE</b>	تاریخ	۳ بایت	این نوع داده برای نگهداری تاریخ میلادی، بدون ساعت است. محدوده تاریخ از '1000-01-01' تا '9999-12-31' است.
<b>TIME</b>	ساعت	۳ بایت	این نوع داده برای نگهداری زمان استفاده می‌شود.

## مهمترین دستورات: SQL

- **SELECT** استخراج داده از یک دیتابیس
- **UPDATE** بروزرسانی داده در یک دیتابیس
- **DELETE** حذف داده از یک دیتابیس
- **INSERT INTO** درج داده جدید در یک دیتابیس
- **CREATE DATABASE** درج یک پایگاه داده جدید
- **ALTER DATABASE** ویرایش یک پایگاه داده
- **CREATE TABLE** ایجاد یک جدول جدید
- **ALTER TABLE** ویرایش یک جدول
- **DROP TABLE** حذف یک جدول
- **CREATE INDEX** ایجاد یک شاخص (کلید جستجو)
- **DROP INDEX** حذف یک شاخص

## ساخت جدول :

```
CREATE TABLE   databasename. tablename(  
  
Namefilde1   datatype,  
  
Namefilde2   datatype,  
  
Namefilde3   datatype,  
  
Namefilde4   datatype,  
  
PRIMARY KEY (Namefilde)  
  
);
```

## مثال :

```
CREATE TABLE lib_school.books (  
keyid INT(10) NOT NULL ,  
name VARCHAR(120) ,  
family VARCHAR(50) ,  
city VARCHAR(50),  
year INT(10) ,  
PRIMARY KEY (keyid)  
);
```



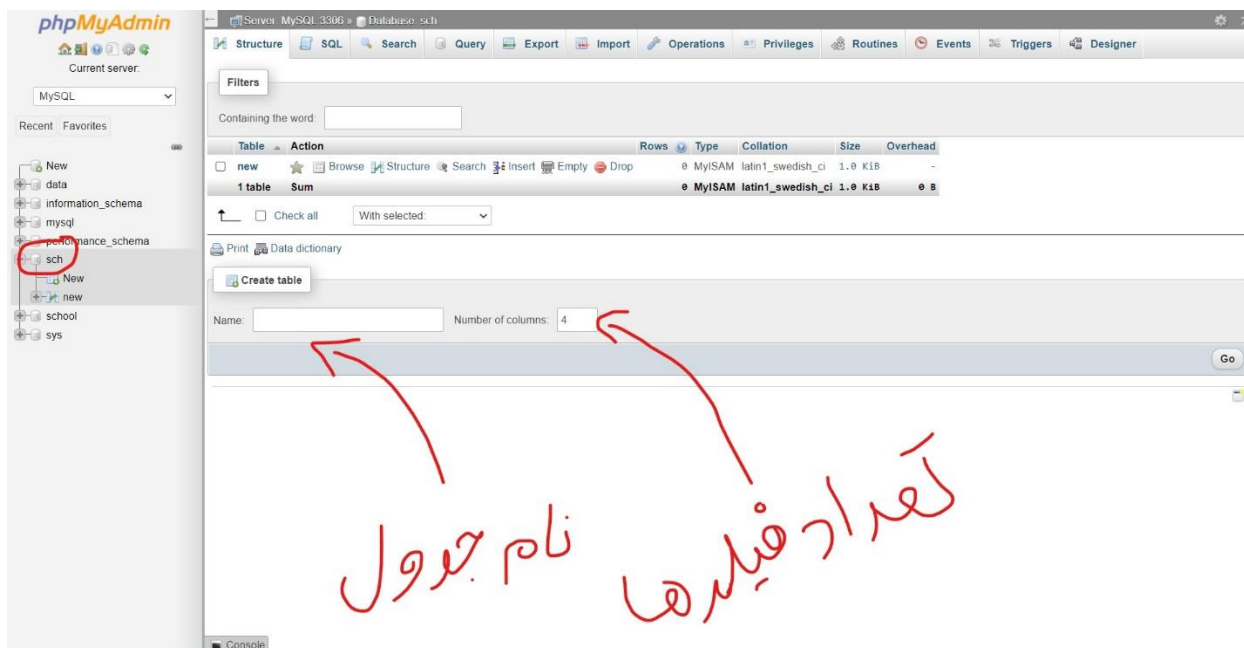
تمرین :

جدول زیر را با کوئری ایجاد کنید.

Description	size	data type	Field name
کدملی (کلید)	۱۰	INT	Id_code
نام	۵۰	VARCHAR	Name
نام خانوادگی	۶۰	VARCHAR	Family
سن	۱۰	INT	Age
تاریخ تولد		DATE	birthdate

ساخت جدول توسط wizard .

برای ایجاد جدول، پایگاه داده مورد نظر را انتخاب کرده، سپس نام جدول و تعداد فیلدها را وارد کنید.



## تعیین زبان جدول:

Table name:  Add  column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---
	INT		None			<input type="checkbox"/>	---

Structure

Table comments:

Collation:

Storage Engine:

PARTITION definition:

Partition by:  ( Expression or c

Partitions:

Console

در قسمت Collation نوع ساختار زبان مورد نظر را انتخاب کنید.

برای نمایش صحیح زبان فارسی از **utf8\_persian\_ci** استفاده میشود.

**جدول تمرین قبل را توسط wizard طراحی کنید.**

## ویرایش ساختار جدول:

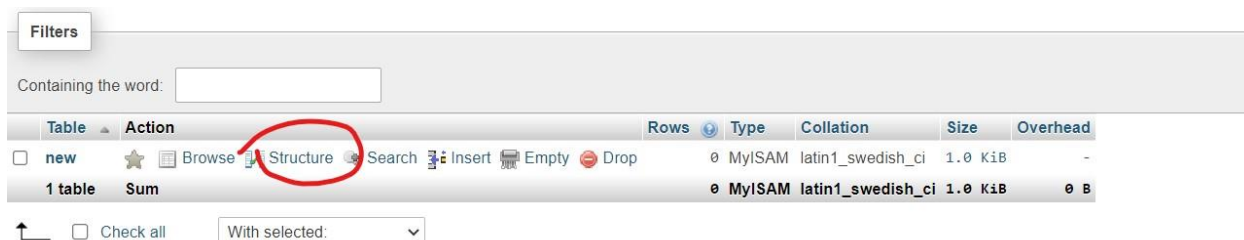
اگر کاربر نوع داده فیلد یا تعداد نویسه یک فیلد را درست وارد نکرده باشد، باید بتوانید ساختار جدول را ویرایش کنید.

برای ویرایش ساختار جدول از دستور **Alter** استفاده میشود.

همچنین میتوانید به صورت **wizard** نیز ویرایش کنید.

## ابتداء توسط wizard تغییرات را اعمال میکنیم.

۱. از فهرست پایگاه داده دیتابیس مورد نظر را انتخاب میکنیم.
۲. جدول را انتخاب کنید.
۳. سپس Structure انتخاب کنید.



## اضافه کردن فیلد:

تعداد فیلد هایی که میخواهید به جدول اضافه کنید را وارد کنید سپس روی دکمه go کلیک کنید.



به جدول تمرین قبل فیلد آدرس را با مشخصات زیر اضافه کنید :

Name	Type	Length/Values	Collation
address	VARCHAR	۴۰۰	utf8-persian-ci

## دستور SQL برای اضافه کردن فیلد:

نام جدول ALTER TABLE

ADD (نوع فیلد نام فیلد)

## حذف فیلد :

با کلیک روی دکمه **DROP** در فیلد مورد نظر میتوان یک فیلد را حذف کرد.

دستور SQL برای حذف فیلد:

نام جدول ALTER TABLE

نام فیلد DROP COLUMN

## ویرایش یک فیلد:

با کلیک روی دکمه change میتوان مقادیر یک فیلد را تغییر داد.

دستور SQL ویرایش فیلد:

ALTER TABLE `نام جدول` CHANGE `نام فیلد`

VARCHAR(200) `نام فیلد`

## تغییر کلید اصلی:

با کلیک روی دکمه Primary میتوان کلید اصلی را تغییر داد.

کد SQL تغییر کلید:

ALTER TABLE `نام جدول` DROP PRIMARY KEY,

ADD PRIMARY KEY (`نام فیلد`)

## درج رکورد در جدول:

از دستور **INSERT** برای درج رکورد در جدول استفاده میشود.

دو روش استفاده از این دستور وجود دارد:

نام جدول . نام پایگاه داده INSERT INTO  
VALUES (مقدار ۱, مقدار ۲, مقدار ۳,...)

(نام فیلد ۱, نام فیلد ۲, نام فیلد ۳,...) نام جدول . نام پایگاه داده INSERT INTO  
VALUES (مقدار ۱, مقدار ۲, مقدار ۳,...)

### تمرین :

جدولی که ساخته اید مقدار دهی کنید.

### حذف رکورد از جدول:

دستور **DELETE** برای حذف record ها در یک جدول استفاده می شود.

دستور **DELETE** برای حذف سطرها در یک جدول استفاده می شود.

نام جدول. نام پایگاه داده **DELETE FROM**

شرط -> **WHERE** some\_column=some\_value;

همچنین میتوان در برگه Browse رکورد را حذف کرد.

### ویرایش رکورد ها:

دستور SQL :

نام جدول. نام دیتابیس **UPDATE**

**SET** column1=value1,  
column2=value2,...

**WHERE** some\_column=some\_val

نام جدول. نام پایگاه داده UPDATE  
SET  
مقدار ۱ = نام فیلد ۱ ,  
مقدار ۲ = نام فیلد ۲ ,  
...  
شرط **WHERE** ;

توجه داشته باشید شرط جلوی **WHERE**  
تعیین میکند کدام رکوردها به روزرسانی  
شوند. در صورتی که از شرط استفاده نشود  
تمام رکوردها به روز می شوند.

## ویژگی افزایش خودکار (AUTO INCREMENT):

تعریف ویژگی **AUTO INCREMENT** روی یک فیلد در SQL، آن فیلد را به یک **شمارنده** تبدیل می کند. به این ترتیب، با افزودن هر رکوردی به جدولی که فیلد شمارنده دارد. یک واحد به شمارنده اضافه می گردد. فیلد شمارنده یکی از گزینه های مناسب برای تعریف به عنوان کلید اصلی است، چراکه مقادیر **یکتا** دارد.

برای این که یک فیلد دارای این ویژگی باشد باید حداقل یکی از ویژگی های : **کلید اصلی (Primary Key)** یا **نامه (Index)** باشد در غیر این صورت پایگاه داده اجازه ایجاد این ویژگی را نمیدهد.

## مراحل ایجاد افزایش خودکار:

- پایگاه داده و جدول موردنظر را انتخاب کنید.
- برای اصلاح ساختار جدول، برگه Structure را انتخاب کنید.
- به فیلد مورد نظر ویژگی AUTO\_INCREMENT را اضافه کنید. (A\_T)



## مثال SQL

```
CREATE TABLE Persons
(
  P_Id int NOT NULL AUTO_INCREMENT,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255),
  PRIMARY KEY (P_Id)
)
```

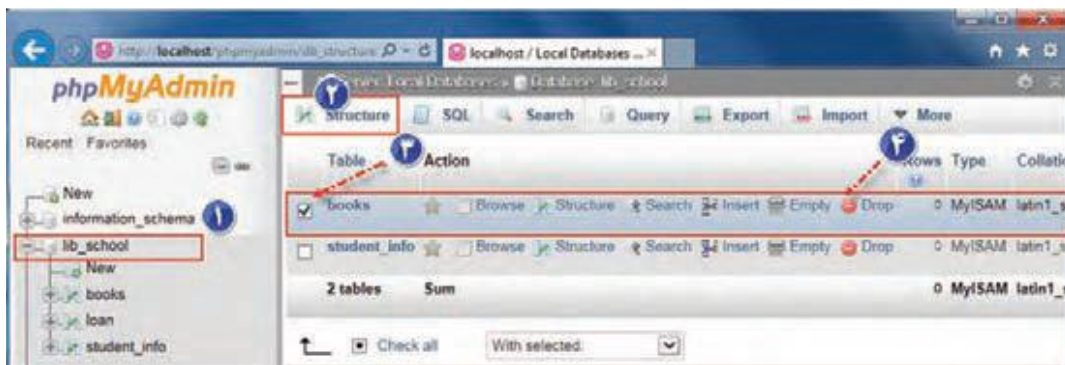
## حذف جدول :

از دستور **DROP TABLE** برای حذف یک جدول از پایگاه داده استفاده می شود.

نام جدول ، نام پایگاه داده **DROP TABLE**

## حذف جدول به صورت Wizard:

پایگاه داده موردنظر را انتخاب و برگه Structure را کلیک کنید. جدول موردنظر را از بین جدول های موجود انتخاب کرده، سپس روی Drop کلیک کنید .



## حذف پایگاه داده :

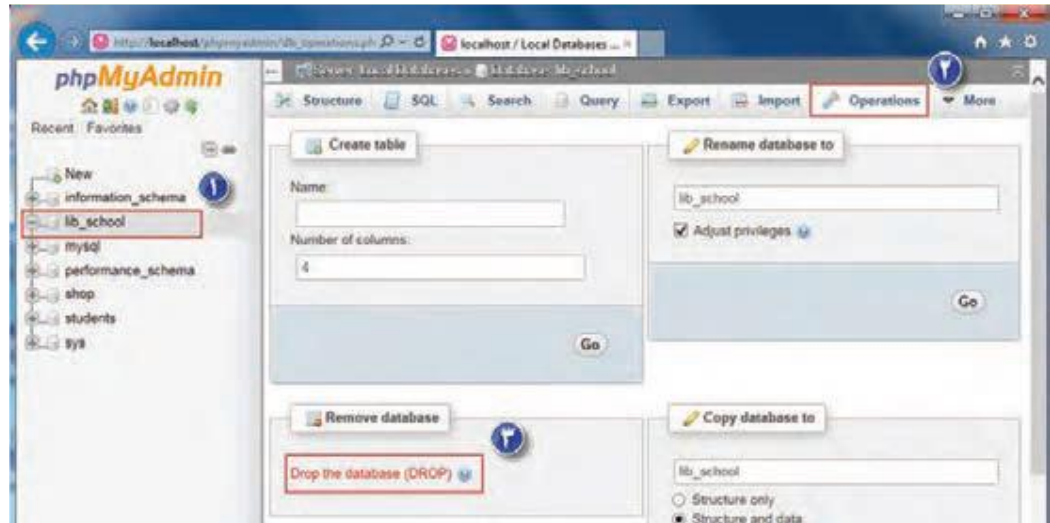
از دستور **DROP DATABASE** برای حذف یک پایگاه داده استفاده می شود.

## دستور SQL:

نام پایگاه داده **DROP DATABASE**

## حذف پایگاه داده به صورت Wizard

پایگاه داده مورد نظر را انتخاب و پس از برگه Structure برگه Operations را انتخاب کنید. روی پیوند Drop the database (DROP) کلیک کنید.



## فصل سوم (اتصال پایگاه داده به وب سایت)



در این فصل قصد داریم درمورد ایجاد کوئری و تهیه گزارش ، اتصال پایگاه داده به صفحه وب و پشتیبان گیری از پایگاه داده بحث کنیم.

### کوئری چیست؟

در زبان انگلیسی استاندارد، query به معنای درخواست اطلاعات است. در برنامه نویسی کامپیوتری نیز به همین معنی است، با این تفاوت که اطلاعات از پایگاه داده بازیابی می شود. به عبارت دیگر، **کوئری به درخواست داده از یک پایگاه داده اشاره دارد.** برای کوئری نوشتن نیاز است از مجموعه دستورات از پیش تعریف شده ای استفاده شود که پایگاه داده بتواند دستورالعمل را درک کند، به این مفهوم زبان کوئری نیز گفته می شود.

### اولین دستور کوئری:

**SELECT**: برای استخراج داده ها از پایگاه داده مورد استفاده قرار می گیرد و یکی از محبوب ترین دستورات محسوب می شود زیرا تقریباً هر درخواست با یک کوئری SELECT شروع می شود.

## شکل کلی دستور select :

نام فیلد۱، نام فیلد۲، ... SELECT

نام جدول FROM

شرط WHERE

فیلد ۱ و فیلد ۲ و ... نام فیلدهایی از جدول هستند که داده های آن نمایش داده می شود. در صورتی که بخواهید همه فیلدها را نمایش دهید به جای وارد کردن نام آنها از نویسه \* استفاده کنید.

برای اینکه شرط های متعددی قرار دهیم، باید تنوع عملگرها را بررسی کنیم.

عملگر	عملکرد	مثال
=	مساوی بودن مقداری را با مقادیر فیلد انتخاب شده بررسی می کند.	"تولید محتوا" = مقادیری که مساوی عبارت «تولید محتوا» باشد را جست و جو می کند.
!=	نامساوی بودن مقداری را با مقادیر فیلد انتخاب شده بررسی می کند.	50 != مقادیری که با عدد 50 مساوی نباشند را جست و جو می کند.
IN	مساوی بودن با چند مقدار را بررسی می کند.	IN ("الزامات"، "برنامه سازی 2") مساوی بودن مقادیر فیلد را با یکی از مقادیر داخل پرانتز بررسی می کند.
BETWEEN	برای انتخاب اطلاعات در یک محدوده خاص، در بین دو مقدار تعیین شده استفاده می شود.	BETWEEN 10 AND 100 بین دو مقدار 10 و 100 را بررسی می کند.
IS NULL	NULL بودن مقدار فیلد را مشخص می کند.	WHERE b_auther IS NULL رکوردهایی که مقدار نام مؤلف آنها تهی است

مثال (=):

برای مثال می‌خواهیم اطلاعات شخصی که نام آن "ali" است را چاپ کنیم:

```
SELECT name,family,age
FROM stu
WHERE name = "ali" ;
```

مثال (!=):

برای مثال در پایگاه داده تمامی اطلاعات افراد با سن ۲۰ درج شده اند می‌خواهیم ببینیم آیا شخصی هست که سن او غیر از ۲۰ وارد شده است یا نه اگر وجود داد آن را نمایش دهد.

```
SELECT name,family,age
FROM stu
WHERE age != '20' ;
```

مثال (IN):

در این مثال می‌خواهیم افرادی که در شهر های کرمان ، تهران ، اصفهان هستند را نمایش دهد

```
SELECT *
FROM stu
WHERE city IN ('kerman', 'esfahan', 'tehran');
```

مثال (BETWEEN):

در این مثال می‌خواهیم سن افرادی که بین ۱۸ تا ۲۰ هستند را نمایش دهیم:

```
SELECT *
FROM stu
WHERE age BETWEEN 18 AND 20;
```

مثال (IS NULL):

در این مثال می‌خواهیم افرادی که سن خود را وارد نکرده اند را نمایش دهیم:

```
SELECT *
FROM stu
WHERE age IS NULL;
```

## مرتب سازی داده های جدول

با استفاده از دستور ORDER BY میتوان خروجی کوئری را بر اساس صعودی (ASC) یا نزولی (DESC) مرتب سازی کرد.

این دستور از طریق مشخص کردن نام ستون یا ستون هایی که بر اساس آن ها می خواهید مرتب سازی انجام شود، استفاده می شود. شکل کلی دستور ORDER BY به شکل زیر است

```
SELECT column_name(s)
FROM table_name
WHERE condition
ORDER BY column_name [ASC|DESC];
```

در اینجا، column\_name(s) مشخص می کند که کدام ستون ها در نتایج پرس و جو باید نمایش داده شوند، table\_name نام جدول است که می خواهید از آن اطلاعات را بازیابی کنید و condition مشخص می کند که کدام ردیف ها باید در نتیجه پرس و جو نمایش داده شوند. در انتها، ORDER BY column\_name ستونی را مشخص می کند که می خواهید بر اساس آن مرتب سازی انجام شود. همچنین می توانید با اضافه کردن ASC یا DESC، ترتیب مرتب سازی را مشخص کنید.

برای مثال، فرض کنید که می خواهید یک لیست از اسامی دانشجویان را بر اساس نام خانوادگی آنها مرتب کنید. دستور SQL زیر این کار را انجام می دهد:

```
SELECT first_name, last_name
FROM students
ORDER BY last_name;
```

این دستور، نتایج را بر اساس ستون last\_name مرتب می کند، یعنی نتایج بر اساس حروف الفبا به ترتیب صعودی مرتب می شوند. برای مرتب سازی به ترتیب نزولی، می توانید از دستور ORDER BY با اضافه کردن کلمه کلیدی DESC به طور زیر استفاده کنید:

```
SELECT first_name, last_name
FROM students
ORDER BY last_name DESC;
```

## توابع اماری :

توابع اماری در SQL برای انجام عملیات های مربوط به محاسبه و نمایش اطلاعات آماری از داده های موجود در جداول استفاده می شوند. در ادامه، توابع اماری رایج در SQL را توضیح می دهیم و یک مثال ساده در هر کدام از آنها را ارائه می کنیم:

۱- تابع: MAX این تابع بیشترین مقدار در یک ستون را برمی گرداند. شکل کلی این تابع به صورت زیر است:

```
SELECT MAX(column_name)
FROM table_name;
```

برای مثال، فرض کنید که می خواهید بیشترین سن دانشجویان را در یک جدول مشخص کنید. دستور SQL زیر این کار را انجام می دهد:

```
SELECT MAX(age)
FROM students;
```

۲- تابع: MIN این تابع کمترین مقدار در یک ستون را برمی گرداند. شکل کلی این تابع به صورت زیر است:

```
SELECT MIN(column_name)
FROM table_name;
```

برای مثال، فرض کنید که می خواهید کمترین نمره دانشجویان را در یک جدول مشخص کنید. دستور SQL زیر این کار را انجام می دهد:

```
SELECT MIN(grade)
FROM students;
```

۳- تابع: COUNT این تابع تعداد ردیف های موجود در یک جدول را برمی گرداند. شکل کلی این تابع به صورت زیر است:

```
SELECT COUNT(*)
FROM table_name;
```

در اینجا، علامت ستاره (\*) به معنای تمام ستون های جدول است. اگر می خواهید تعداد ردیف های موجود در یک ستون خاص را بشمارید، نام آن ستون را به جای علامت ستاره قرار دهید.

باری مثال، فرض کنید که می خواهید تعداد دانشجویان را در یک جدول مشخص کنید. دستور SQL زیر این کار را انجام می دهد:

```
SELECT COUNT(*)  
FROM students;
```

این دستور، تعداد کل ردیف های موجود در جدول students را برمی گرداند. برای محاسبه تعداد ردیف های موجود در یک ستون خاص، نام آن ستون را به جای علامت ستاره قرار دهید، به عنوان مثال:

```
SELECT COUNT(grade)  
FROM students;
```

این دستور، تعداد کل ردیف های موجود در ستون grade جدول students را برمی گرداند.

**اتصال پایگاه داده به صفحه وب:**  
**در زبان PHP چندین تابع برای اتصال به پایگاه داده تعریف شده است که به آنها میپردازیم .**

## تابع اول (mysqli\_connect)

این تابع برای تست اتصال صفحه وب به پایگاه داده استفاده میشود.  
ساختار تابع این گونه است:

برای اتصال به پایگاه داده در PHP ، می‌توان از تابع `mysqli_connect()` استفاده کرد. این تابع به چهار آرگومان نیاز دارد:

1. (است "localhost" معمولاً) نام سرور
2. نام کاربری
3. رمز عبور
4. نام پایگاه داده

برای مثال، اگر نام کاربری "root" ، رمز عبور "" و نام پایگاه داده "mydatabase" باشد، کد اتصال به پایگاه داده در PHP به صورت زیر است:

```
<?php
// اتصال به پایگاه داده
$connection = mysqli_connect("localhost", "root", "", "mydatabase");

// بررسی اتصال
if (!$connection) {
    die("mysql_connect_error() . : اتصال با مشکل مواجه شد");
}
echo "اتصال برقرار شد";
?>
```

در این کد، تابع `mysqli_connect()` برای اتصال به پایگاه داده استفاده شده است و نتیجه آن در متغیر `$connection` ذخیره می‌شود. سپس با استفاده از دستور `if`، بررسی می‌شود که آیا اتصال به درستی برقرار شده است یا خیر. در صورتی که اتصال برقرار نشده باشد، با



استفاده از تابع `mysqli_connect_error()` خطا نمایش داده می‌شود. در غیر این صورت، پیام "اتصال برقرار شد" نمایش داده می‌شود.

## تابع دوم (`mysqli_connect_errno()`)

تابع `mysqli_connect_errno()` یک تابع بسیار مفید در PHP است که برای دریافت کد خطا مربوط به تلاش برای برقراری اتصال به پایگاه داده با استفاده از تابع `mysqli_connect()` به کار می‌رود.

خروجی این تابع در صورت داشتن خطا شماره خطا را برمیگرداند در صورت عدم خطا عدد ۰ را برمیگرداند

برای مثال:

```
<?php
$connection = mysqli_connect("localhost", "root", "", "mydatabase");

if (mysqli_connect_errno()) {
    echo "خطا در برقراری اتصال به پایگاه داده" . mysqli_connect_error();
}
?>
```

## تابع سوم (`mysqli_connect_error()`):

تابع `mysqli_connect_error()` نیز یک تابع مهم در PHP است که برای نمایش **پیام خطای** مربوط به تلاش برای برقراری اتصال به پایگاه داده با استفاده از تابع `mysqli_connect()` به کار می‌رود.

مثال:

```
<?php
$connection = mysqli_connect("localhost", "root", "", "mydatabase");

if (mysqli_connect_error()) {
    echo "خطا در برقراری اتصال به پایگاه داده" . mysqli_connect_error();
}
?>
```

## تابع چهارم (mysqli\_query):

تابع `mysqli_query()` یکی دیگر از توابع مهم در PHP است که برای اجرای یک پرس و جو در پایگاه داده با استفاده از اتصال برقرار شده به کار می‌رود. این تابع برای اجرای هر نوع پرس و جویی در پایگاه داده، اعم از `SELECT`، `INSERT`، `UPDATE` و `DELETE` استفاده می‌شود. ساختار کلی تابع `mysqli_query()` به صورت زیر است:

`mysqli_query(connection, query)`

در اینجا:

- `connection`: اتصالی است که قبلاً با استفاده از تابع `mysqli_connect()` برقرار شده است.
- `query`: پرس و جویی است که می‌خواهید در پایگاه داده اجرا کنید.

به عنوان مثال، اگر بخواهیم یک پرس و جوی `SELECT` برای بازیابی اطلاعات از جدول `users` در پایگاه داده اجرا کنیم، می‌توانیم از تابع `mysqli_query()` به صورت زیر استفاده کنیم:

```

<?php
// برقراری اتصال به پایگاه داده
$connection = mysqli_connect("localhost", "username", "password",
"database_name");

// اجرای پرس و جوی SELECT
$result = mysqli_query($connection, "SELECT * FROM users");

// بازایی شده‌اند SELECT بازایی ردیف‌هایی که با پرس و جوی
while ($row = mysqli_fetch_assoc($result)) {
    echo "نام: " . $row["name"] . "، ایمیل: " . $row["email"] . "<br>";
}

// بستن اتصال به پایگاه داده
mysqli_close($connection);
?>

```

در این مثال، پرس و جوی SELECT با استفاده از تابع `mysqli_query()` اجرا شده و ردیف‌های بازایی شده با استفاده از تابع `mysqli_fetch_assoc()` بازایی شده‌اند. سپس اتصال به پایگاه داده با استفاده از تابع `mysqli_close()` بسته می‌شود.

## تابع پنجم (mysqli\_close):

تابع `mysqli_close()` یکی دیگر از توابع مهم در PHP است که برای قطع اتصال به پایگاه داده از طریق اتصالی که قبلاً با استفاده از تابع `mysqli_connect()` برقرار شده است، به کار می‌رود. ساختار کلی تابع `mysqli_close()` به صورت زیر است:

**`mysqli_close(connection)`**

در اینجا:

- **connection:** اتصالی است که قبلاً با استفاده از تابع `mysqli_connect()` برقرار شده است.

به عنوان مثال، برای بستن اتصال به پایگاه داده، می‌توانیم از تابع `mysqli_close()` به صورت زیر استفاده کنیم:

```
<?php
```

```
// برقراری اتصال به پایگاه داده
```

```
$connection = mysqli_connect("localhost", "username",  
"password", "database_name");
```

```
// اجرای پرس و جوها و بازیابی اطلاعات
```

```
// بستن اتصال به پایگاه داده
```

```
mysqli_close($connection);
```

```
?>
```

## تابع ششم (mysqli\_fetch\_assoc())

تابع `mysqli_fetch_assoc()` یکی از توابع مهم در PHP است که برای بازیابی یک ردیف از نتایج یک پرس و جوی SQL در PHP با استفاده از اتصال `mysqli` به پایگاه داده، به کار می‌رود. ساختار کلی تابع `mysqli_fetch_assoc()` به صورت زیر است:

### `mysqli_fetch_assoc(result)`

در اینجا:

- `result`: نتیجه پرس و جوی SQL است که برای آن از تابع `mysqli_query()` استفاده شده است.

#### یک مثال جهت یادآوری اضافه کردن مقادیر به دیتابیس:

همانطور که در مباحث قبل در مورد کوئری اضافه کردن رکورد صحبت کردیم می‌خواهیم در php از آن استفاده کنیم و مقادیر دو متغیر را در دیتابیس ذخیره کنیم. برای اضافه کردن رکورد از دستور **INSERT INTO** استفاده می‌کنیم. حالا ببینیم در PHP چگونه پیاده سازی میشود: سناریو: ما یک دیتابیس دو فیلدی داریم (نام و ایمیل) حالا می‌خواهیم توسط دو متغیر مقادیری را در دیتابیس ذخیره کنیم. مرحله اول: ابتدا دو متغیر با نام دلخواه اضافه کنید در این جا ما از دو متغیر به نام (\$user,\$email) استفاده می‌کنیم. مرحله دوم: برقراری اتصال با دیتابیس را ایجاد می‌کنیم. مرحله سوم: در این مرحله باید کوئری اضافه کردن رکورد را وارد کنیم.

تمام کوئری مانند کوئری داخل phpMyAdmin است با تفاوت این که به جای وارد کردن مستقیم اطلاعات از متغیر استفاده میشود مثال:

**PhpMyAdmin:** INSERT INTO نام پایگاه داده نام جدول . نام پایگاه داده  
VALUES (...مقدار ۳, مقدار ۲, مقدار ۱)

---

**PHP Code:** INSERT INTO `table` (`name`, `email`)  
VALUES ('متغیر۱', 'متغیر۲');

کد نهایی مثال به این صورت است:

```
<?PHP
$user="abdolah";
$email="abdolah.com";
// برقراری اتصال به پایگاه داده
$connection = mysqli_connect("localhost", "root", "", "mz");
// اجرای پرس و جوی INSERT
$result = mysqli_query($connection,
"INSERT INTO `table` (`username`, `password`) VALUES ('$user', '$user');");
// بستن اتصال به پایگاه داده
mysqli_close($connection);
?>
```

**پروژه (۵ نمره):** یک صفحه لاگین ایجاد کنید که مقادیر نام و رمز عبور را از کاربر دریافت کند سپس اطلاعات اعتبار سنجی شوند بعد در صورت مشکل در ورودی به کاربر اخطار دهد اگر مشکل وجود نداشت اطلاعات را در دیتابیس ذخیره کند. بعد از ذخیره سازی پیغامی مبنی بر ذخیره اطلاعات به کاربر بدهد سپس مقادیر داخل دیتابیس را نمایش دهد

## پشتیبان گیری:

### • چرا ما نیاز داریم از دیتابیس پشتیبان بگیریم؟

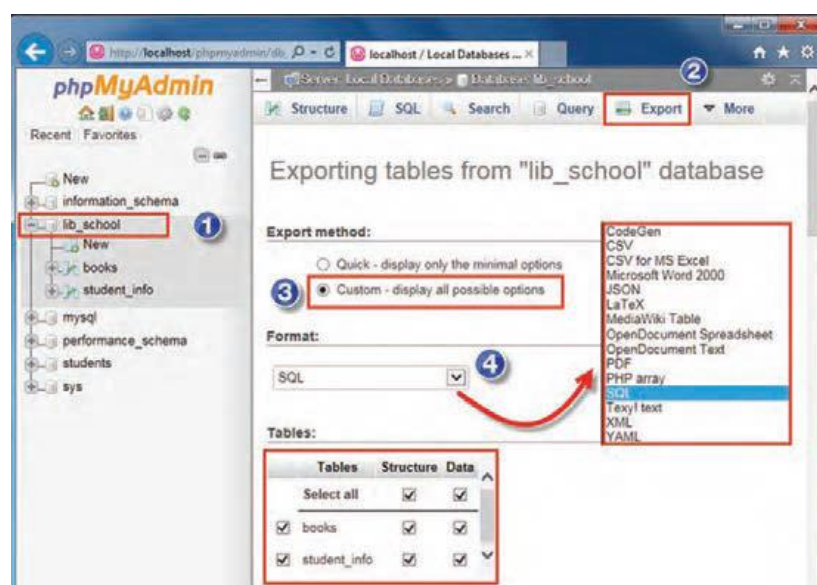
در دنیای فناوری اطلاعات، اطلاعات به عنوان یکی از دارایی‌های بسیار ارزشمند شناخته می‌شوند. همانطور که میدانید، پایگاه داده‌ها حاوی اطلاعات مهمی از جمله اطلاعات کاربران، اطلاعات محصولات، اطلاعات مالی و غیره هستند که از ارزش بسیار بالایی برخوردار هستند.

اگر به هر دلیلی اطلاعات موجود در پایگاه داده از بین بروند، ممکن است برای شما تلفات اقتصادی بسیار بالایی به وجود آید. به همین دلیل، برای پیشگیری از این مشکلات و حفظ اطلاعات، پشتیبان‌گیری از پایگاه داده بسیار مهم است.

به علاوه، پشتیبان‌گیری از پایگاه داده در صورتی که بخواهید از یک سیستم به سیستم دیگر منتقل کنید، بسیار مفید است. در این صورت، با استفاده از فایل پشتیبان، می‌توانید اطلاعات خود را به راحتی و با دقت به سیستم دیگر منتقل کنید.

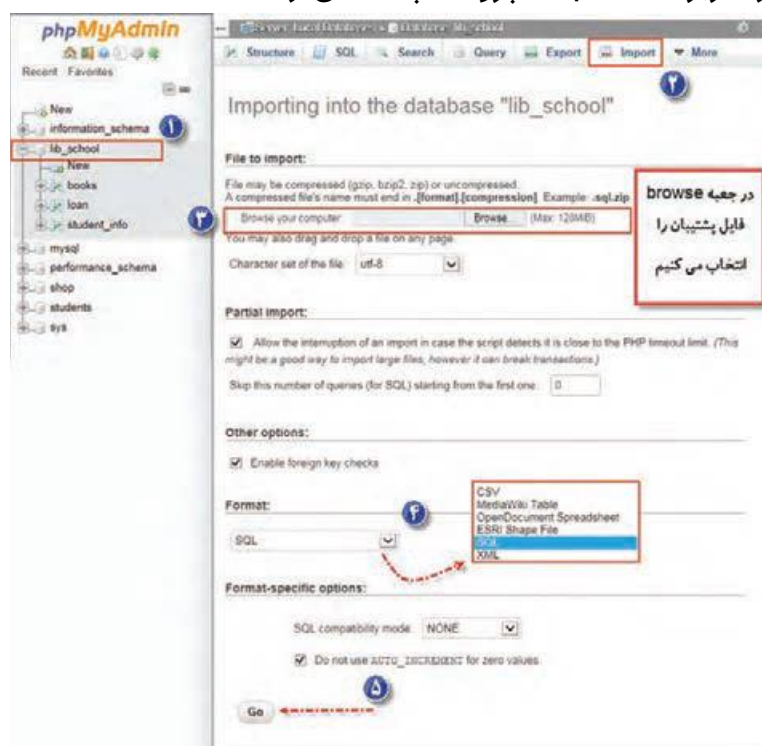
### مراحل تهیه فایل پشتیبان (از پایگاه داده و جدول):

۱. PhpMyAdmin را اجرا کنید
۲. (دیتابیس، جدول) مورد نظر را انتخاب کنید
۳. در سر برگ، برگه export را انتخاب کنید
۴. خروجی مورد نظر را انتخاب کنید (فرمت‌های مختلفی از جمله sql, pdf و ...)



### بازیابی (دیتابیس ، جدول) :

۱. پایگاه داده مقصد برای بازیابی اطلاعات را باز کنید.
۲. در صورتی که پایگاه داده وجود ندارد ابتدا آن را ایجاد کنید.
۳. از سربرگ import پرونده پشتیبان را انتخاب کنید.





هنگام پشتیبان گیری از پایگاه داده چه تنظیمی انجام شود تا نیاز به ایجاد پایگاه داده هنگام بازیابی نباشد؟

باید قبل از گرفتن فایل پشتیبان حالت **Custom - display all possible options** را انتخاب کنیم تا وارد تنظیمات بیشتر شویم سپس در بخش **Object creation options** گزینه **Add CREATE DATABASE / USE statement** و گزینه **IF NOT EXISTS** را انتخاب کنیم .

---

این مراحل در گرفتن فایل پشتیبان جدول یکسان است.

## پایان پودمان ۴

## پودمان ۵ (مدیریت صفحات وب پویا)

در طراحی وبسایت‌های پویا، اطلاعات کاربران بسیار مهم هستند و این اطلاعات باید به درستی دریافت، ذخیره و مدیریت شوند. در این پودمان، به بررسی چگونگی مدیریت صفحات وب پویا با استفاده از PHP می‌پردازیم.

### در این پودمان چندین تابع و آرایه جهت مدیریت صفحات پویا استفاده میشود را بررسی میکنیم.

#### آرایه `$_SESSION`:

کاربر در صفحه وب دارای دو حالت است یا کاربر لاگین کرده است یا نکرده است. با توجه به این که سرویس‌هایی که سایت ارائه میکند به وضعیت کاربر بستگی دارد باید بتوانیم از وضعیت کاربر اطلاع داشته باشیم. برای این کار باید از session استفاده کنیم تا وضعیت کاربر را ذخیره کنیم. Session چه کار میکند؟ یک پرونده در پوشه موقت روی سرویس دهنده ایجاد میکند که متغیرهای ثبت شده در session و مقادیر آنها ذخیره میشود. تعریف و مقدار دهی کلی :

مقدار = `$_SESSION["نام متغیر"]`

## تابع session\_start

برای این که بتوان از ارایه \$\_SESSION استفاده کرد باید تابع session\_start را فراخوانی کرد.

نکته قبل از فراخوانی تابع نباید دستوری از جنس HTML یا مشابه آن وجود داشته باشد. در صورتی که قبل از فراخوانی این تابع مثلاً با echo عبارتی را نمایش داده باشید با خطا یا عدم کارکرد صحیح session ها مواجه میشوید.  
مثال :

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
    <?php
        $_SESSION["user"]="ali";
        $_SESSION["pass"]="12345";
        echo("set");
    ?>
</body>
</html>
```

حالا بیاییم تعداد بازدید های سایت را مشاهده کنیم.

```
<?php
session_start();
if (isset($_SESSION['namesess'])){ // بررسی موجود بودن متغیر
    $_SESSION['namesess']+=1;
}else{
    $_SESSION['namesess']=1; اگر متغیر وجود نداشت مقدار
    // دهی اولیه میشود
}
$mas="visit".$_SESSION['namesess'];
?>

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
    <?php
    echo($mas);
    ?>
</body>
</html>
```

## تابع unset:

برای حذف محتوای یک متغیر از این تابع استفاده میشود.

مثال : زمانی که کد قبلی را اجرا کردیم بعد از هر بار رفرش صفحه یک بازدید اضافه میشود حالا میخواهیم این بازدید ها را ریست کنیم .

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
    <?php
        unset($_SESSION['namesess']); حذف تعداد بازدید ها
    ?>
</body>
</html>
```

## تابع `session_unset` :

این تابع برای حذف متغیرهای session استفاده میشود این تابع ارگومان ورودی ندارد.

## تابع `session_destroy` :

این تابع جهت حذف کامل پرونده ایجاد شده برای session در وب سرویس استفاده میشود .

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>

</body>
</html>
```

حذف همه متغیرهای تعریف شده در session

از بین بردن و تخریب کامل فایل ایجاد شده برای session

## ارسال پرونده به سرویس دهنده میزبان :

در این قسمت با توابع کار با فایل ها آشنا میشویم.

### تابع `basename`:

این تابع مسیر کامل یک پرونده را از ورودی دریافت کرده و در خروجی نام و پسوند پرونده را به صورت یک رشته برمیگرداند.

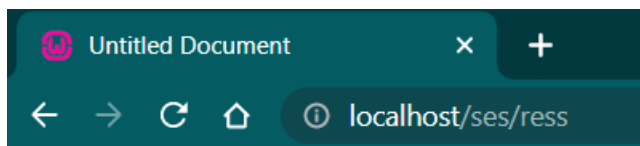
شکل کلی :

```
$vari=basename(location);
```

مثال :

```
<?php
    echo basename("C:\wamp64\www\sas\index.php")
?>
```

خروجی :



index.php

## تابع file\_exists:

این تابع مسیر کامل یک پرونده یا پوشه را دریافت کرده سپس در خروجی وجود یا عدم وجود آن را با true یا false برمیگرداند.

مثال :

```
<?php
$f="C:\wamp64\www\ses\dd";
if (file_exists($f)){
    echo("پرونده وجود داد");
}else{
    echo("پرونده وجود ندارد");
}
?>
```

## تابع pathinfo :

این تابع مسیر کامل را از ورودی دریافت سپس در یک خروجی آرایه انجمنی از اطلاعات پرونده و مسیر دسترسی آن نمایش میدهد .  
شکل کلی :

```
$vari=pathinfo(location);
```

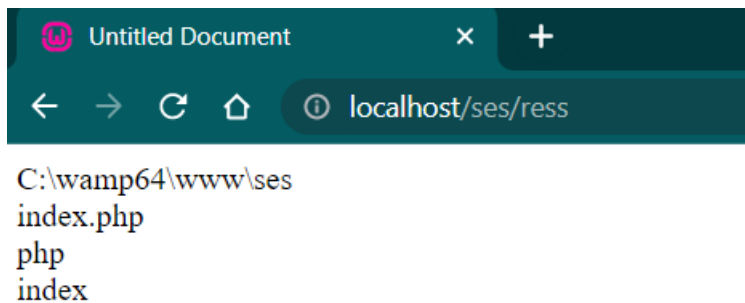
نام شناسه	توضیحات شناسه
PATHINFO_DIRNAME	مسیر و نام پوشه‌های پرونده را برمی‌گرداند.
PATHINFO_BASENAME	نام و پسوند پرونده را برمی‌گرداند.
PATHINFO_EXTENSION	پسوند پرونده را برمی‌گرداند.
PATHINFO_FILENAME	نام را بدون پسوند پرونده برمی‌گرداند.



مثال :

```
<?php
    $f="C:\wamp64\www\sas\index.php";
    echo pathinfo($f,PATHINFO_DIRNAME)."<br>";
    echo pathinfo($f,PATHINFO_BASENAME)."<br>";
    echo pathinfo($f,PATHINFO_EXTENSION)."<br>";
    echo pathinfo($f,PATHINFO_FILENAME)."<br>";
?>
```

خروجی :



## تابع `getimagesize`:

این تابع در ورودی مسیر کامل و نام یک تصویر را دریافت کرده، در خروجی به صورت یک آرایه انجمنی اطلاعات کاملی از تصویر را برمی گرداند. در صورتی که پرونده ورودی از نوع تصویری نباشد، خروجی این تابع `false` خواهد بود.

## شکل کلی تابع :

```
$a= getimagesize(location);
```

## حالا چگونه از این آرایه انجمنی استفاده کنیم؟

بعد از این که اطلاعات تصویر در آرایه ذخیره شد هر کدام از اندیس ها یک نام دارند که در جدول زیر کاربرد و نام آنها آمده است:

اندیس انجمنی آرایه در خروجی	توضیحات	مثال در خروجی
۰	عرض تصویر بر اساس پیکسل	275
۱	ارتفاع تصویر بر اساس پیکسل	183
۳	عرض و ارتفاع تصویر به صورت رشته‌ای	width="275" height="183"
bits	تعداد بیت‌های رنگ تصویر	8
channels	تعداد کانال‌های رنگ تصویر	3
mime	نوع تصویر	image/jpeg

مثال : در این مثال ما یک ادرس تصویر به برنامه دادیم حالا می‌خواهیم به اطلاعات آرایه دسترسی داشته باشیم کد آماده شده به شکل زیر است :

```
<?php
$a=getimagesize("C:\wamp64\www\sas\js.png");
echo $a[0]."<br>";
echo $a[1]."<br>";
echo $a[3]."<br>";
echo $a[bits]."<br>";
echo $a[channels]."<br>";// احتمال خطا وجود دارد
echo $a[mime]."<br>";

?>
```

**تمرین :** درمورد آرایه `$_FILES` و تابع `move_uploaded_file` تحقیق و برای هر کدام یک مثال ساده بیان کنید (۳ نمره).

### تابع `substr`:

تابع `substr()` در PHP بخشی از یک رشته ی کامل را برش داده و آن را در خروجی استخراج می کند.

برای مثال : برای خرید در یک سایت لازم است که فهرست کالاهای فروشگاه را مشاهده کنید و مشخصات کالای موردنظر خود را با جزییات بیشتری مانند امکانات و قیمت آن بررسی کنید. مشخصات کالا در فروشگاه برخط به صورت کامل در یک رشته ذخیره می شود؛ ولی لازم نیست همیشه تمام این مشخصات دیده شود ، بلکه نمایش بخش کوچکی از آن کفایت می کند. **تابع `substr`** بخشی از یک رشته را برمیگرداند.

### شکل کلی این تابع:

**`$a=substr(` (تعداد نویسه ,شروع اولین نویسه ,رشته )`);`**

**عدد • نشان دهنده اولین کاراکتر است.**

نوشتن اعداد منفی برای اندیس شروع اولین نویسه باعث می شود بررسی رشته به صورت معکوس انجام شود.

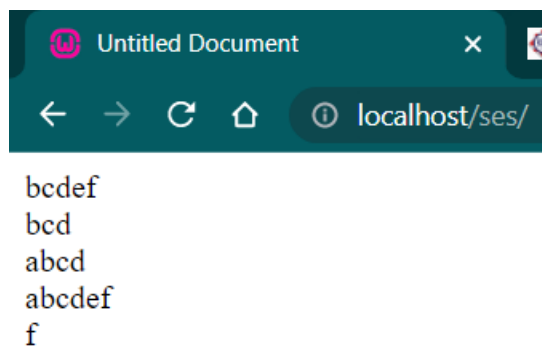
## مثال :

<?php

```
echo substr('abcdef',1)."<br>";    => bcdef
echo substr('abcdef',1,3)."<br>";  => bcd
echo substr('abcdef',0,4)."<br>";  => abcd
echo substr('abcdef',0,8)."<br>";  => abcdef
echo substr('abcdef',-1,1)."<br>"; => f
```

?>

خروجی به شکل زیر است:



## تابع date:

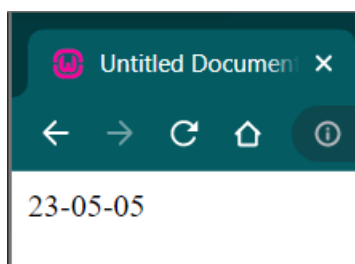
این تابع تاریخ و زمان جاری سیستم را برمیگرداند .  
شکل کلی تابع به این شکل است .

```
$a=date("فرمت نمایش تاریخ");
```

یک نمونه از قالب نمایش تاریخ **y-m-d** است به مثال زیر توجه کنید:

```
$a=date("y-m-d ");  
echo $a;
```

خروجی به شکل زیر است :



برای مشاهده دیگر فرمت ها به صفحه ۲۷۵ کتاب مراجعه کنید.

## تابع unlink:

بری حذف یک پرونده از این تابع استفاده میشود .  
شکل کلی این تابع :

```
unlink("مسیر پرونده");
```

اگر عملیات به درستی انجام شد true و در غیر این صورت false برمیگرداند

**مثال:** در این مثال می‌خواهیم یک پرونده به نام js.png را حذف کنیم اگر حذف به درستی انجام شد پیام ok و در غیر این صورت پیام err بدهد .

```
<?php
$a="C:\wamp64\www\ses\js.png";
if (!unlink($a)){
    echo "err";
}else{
    echo("ok");
}
?>
```

**پایان کتاب طراحی وبسایت و مدیریت پایگاه داده**  
**پیروز و موفق باشید. ☺**