



<HTML>

Hypertext Markup Language

HTML কি?

HTML কোন প্রোগ্রামিং ল্যাঙ্গুয়েজ নয়, একে **Hyper Text Mark Up Language** বলা হয়। **Mark Up Language** এক সেট **Mark Up** ট্যাগের সমন্বয়ে গঠিত হয়। একটা ওয়েব পেজের বিভিন্ন অংশ ব্রাউজারের মাধ্যমে কিভাবে প্রদর্শিত হবে, তা **HTML** এ **Mark Up** ট্যাগ সমূহ ব্যবহার করে প্রকাশ করা হয়।

HTML VERSIONS

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2012
XHTML5	2013



HTML EDITORS

- HTML can be edited by using a professional HTML editor like:
- Adobe Dreamweaver
- Microsoft Expression Web
- CoffeeCup HTML Editor
- Notepad ++ / Notepad

HTML DOCUMENT EXAMPLE

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This is my first paragraph.</p>
```

```
</body>
```

```
</html>
```


HTML ট্যাগ কি?

HTML এ প্রোগ্রাম লেখার জন্য <> এবং দুইটা চিহ্ন এবং এর মধ্যে কিছু Pre-defined Word যেমন html, head, title, body ইত্যাদি Keyword ব্যবহার করা হয়। <> বা চিহ্ন এবং এর মাঝে লেখা এক একটি Keyword কে একত্রে ট্যাগ বলা হয়।

HTML Element কি?

HTML এ যেকোন শুরু ও শেষ tag এবং মাঝের অংশকে সংশ্লিষ্ট tag এর Element বলা হয়। যেমন `<h1> This is an example of element.</h1>` । এখানে `<h1>` হেডার 1 শুরু এবং `</h1>` হেডার1 শেষ ট্যাগের মাঝে `This is an example of element.` লেখা হয়েছে, তাই `This is an example of element.` হচ্ছে `h1` tag এর element । কিছু কিছু ট্যাগের কোন Element থাকে না যেমন `
` `` ইত্যাদি।

সাধারণতঃ যে সকল ট্যাগের শেষ ট্যাগ থাকে না তাদের ইলিমেন্টও থাকে না। এ ধরনের শুরু ট্যাগের মধ্যেই / চিহ্নটি অন্তর্ভুক্ত থাকে, এবং এর আগে একটা স্পেস দিতে হয়। আর এগুলোকে বলা হয় HTML empty Element

HTML Attribute কি?

HTML এ ট্যাগের কার্যক্ষমতাকে বর্ধিত করার জন্য শুরু tag এর মধ্যে কিছু pre-defined কথাকে HTML Attribute বলে। মূলত HTML এ ট্যাগের কার্যক্ষমতাকে বর্ধিত করার জন্য Attributes ব্যবহার করা হয়। যেমন ` Hello SAtech360.` এখানে `size="5"` অংশটি হল font ট্যাগের একটি এট্রিবিউট, যা প্রকাশ করছে ইলিমেন্ট অর্থাৎ This is a paragraph. লেখাটির সাইজ কেমন হবে। এবং `color="red"` দ্বারা প্রকাশ করছে লেখাটির রং হবে লাল।

HTML Entities কি?

HTML এ কিছু বিশেষ Character যেমন <, > চিহ্ন এটি সাধারণত HTML tag গুলো লেখার কাজে ব্যবহৃত হয়, তাই এগুলোর মধ্যে কিছু লিখলে browser সেটিকে ট্যাগ ভেবে ভুল করে বা আমাদের ইচ্ছা অনুযায়ী সঠিক ভাবে তথ্য প্রদর্শন করতে পারেনা। আবার এমন কিছু চিহ্ন যেমন © ® ™ ইত্যাদি চিহ্ন গুলি এডিটরে লেখার জন্য আপনার কীবোর্ড এ লেখার ব্যবস্থা নেই। এখন যদি এই ধরনের কোন চিহ্ন আপনি ওয়েব পেজে দেখাতে চান তাহলে এইচটিএমএল এনটাইটি ব্যবহার করে দেখাতে পারবেন।

কয়েকটি বহুল ব্যবহৃত এনটাইটি:

©

®

™

$

<

>

HTML DOCTYPE Tag কি?

HTML এ `<!DOCTYPE>` Tag হচ্ছে আপনার লেখা HTML ডকুমেন্টটি কি ধরনের তা ব্রাউজারকে বোঝানোর জন্য ব্যবহৃত হয়। এইচটিএমএল এর কোন ভার্সন ব্যবহার করবেন সেটার উপর ভিত্তি করেই এই ডিক্লেয়ারেশন টি দেয়া হয়। যেমন উপরের ডকুমেন্ট টি এইচটিএমএল ৫ এর একটি ডকুমেন্ট, তাই এটাকে আমরা এভাবে `<!DOCTYPE html>` লিখি। এরূপ যদি এটা XHTML এর ১.০ ভার্সন হতো তাহলে ডিক্লেয়ারেশনটি দিতে হতো এভাবে:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML Comments কি ?

HTML এ Comments হচ্ছে, একজন HTML কোডার বা প্রোগ্রামারের সোর্স কোডের ব্যাখ্যা বা পাদটীকা। এটাকে আমরা Coding Documentation ও বলতে পারি। সাধারণত: যেকোনো Programming language এর Compiler গুলো comments কে কোড হিসাবে execute না করে এটাকে (Ignore) এড়িয়ে যায়। Comments System দিয়ে আমরা HTML এর Coding Documentation ছাড়াও এই মুহূর্তে ব্যবহৃত হবেনা কিন্তু ভবিষ্যতে কাজে লাগতে পারে এমন HTML Code গুলোও আমরা সাময়িক সময়ের জন্য hide করে রাখতে পারি।



HTML Comments Example

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Comment</title>
</head>
<body>
  <!--This is a paragraph-->
  <p>Hello HTML!</p>
</body>
</html>
```

HTML body tag Elements কি?

browser এর body তে কিভাবে data প্রদর্শিত হবে তা HTML body tag Element এর মাধ্যমে নির্ধারণ করা হয়। এবং এটি `</head>` ট্যাগ এর পরেই বসে। আর html এর `<body>` `</body>` এর মধ্যে অবস্থিত সব tag ই body tag elements ।

HTML এ body tag element কি কি?

- ❖ Heading Tag Elements
- ❖ Paragraph Tag Elements
- ❖ List Tag Elements
- ❖ Link Tag Element
- ❖ Image Tag Element
- ❖ Text Formatting Tag Elements
- ❖ Layout Tag Elements
- ❖ Iframe Tag element
- ❖ Table Tag Elements
- ❖ Form Tag Elements
- ❖ Block Tag Elements

Heading Tag Element কি?

HTML এর মাধ্যমে কোন ডকুমেন্ট বা প্যারাগ্রাফের শিরোনাম লেখার জন্য Heading Tag ব্যবহার করা হয়। HTML এ মোট ছয় ধরনের হেডিং ট্যাগ রয়েছে এগুলো হল `<h1> </h1>` , `<h2> </h2>` , `<h3> </h3>` , `<h4> </h4>` , `<h5> </h5>` এবং `<h6> </h6>` ।

Example of Heading Tag Element

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Heading Example</title>

</head>
<body>
  <h1>This is an example of heading 1</h1>
  <h2>This is an example of heading 2</h2>
  <h3>This is an example of heading 3</h3>
  <h4>This is an example of heading 4</h4>
  <h5>This is an example of heading 5</h5>
  <h6>This is an example of heading 6</h6>
</body>
</html>
```

Paragraph Tag Element কি?

যে কোন ডকুমেন্ট এক বা একাধিক প্যারাগ্রাফের মাধ্যমে লেখা হয়। HTML এ প্যারাগ্রাফ তৈরির জন্য `<p>` বা প্যারাগ্রাফ ট্যাগ ব্যবহার করা হয়। যেমন `<p>This is a paragraph.</p>`। ব্রাউজারের মাধ্যমে প্রতিটা প্যারাগ্রাফ প্রদর্শন করা হলে প্রতিটা প্যারাগ্রাফের পর একটা করে লাইন ব্রেক তৈরি হয়।

Example Of Paragraph Tag Element

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Paragraph Example</title>

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


List Tag Element কি?

আমাদের দৈনন্দিন দিনের বিভিন্ন বিষয়কে যেমন লিস্ট আকারে সাজায় রাখি। ঠিক একই ভাবে একটা ওয়েব পেজের কন্টেন্ট কে সুন্দর করে সাজানো এবং এর তথ্য উপস্থাপনার অন্যতম পদ্ধতি হচ্ছে লিষ্ট। HTML এর মাধ্যমে তিন ধরনের লিষ্ট তৈরি করা যায়,

একটি হচ্ছে Order List

এবং দ্বিতীয়টি হচ্ছে Un-Order List

এবং তৃতীয়টি হচ্ছে definition List। Order List

এ বিভিন্ন তথ্যকে পর্যায়ক্রমিকভাবে সাজিয়ে উপস্থাপন করা হয় এবং প্রতিটা লাইনের শুরুতে ক্রমিক সংখ্যা থাকে। অন্যদিকে Un-Order List এ প্রতিটা লাইনের সামনে ছোট বৃত্তাকার বা বর্গাকার চিহ্ন সহ অনেক ধরনের লিস্ট থাকে। আর definition list দিয়ে বিভিন্ন বিষয়ের ডেফিনেশন কে লিস্ট আকারে সাজানো হয়। HTML এর মাধ্যমে

Un-Order List তৈরি করার জন্য ``

এবং Order List তৈরি করার জন্য `` ট্যাগ ব্যবহার করা হয়।

আর definition list তৈরী করার জন্য `<dl></dl>`, `<dt>` এবং `<dd></dd>` ই তিনটি ব্যবহৃত হয়।

Example of un-order list

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Un Order List Example</title>

</head>
<body>
  <h4>Disc Type list</h4>
  <ul type="disc">
    <li>Home</li>
    <li>About Us</li>
    <li>Contact Us</li>
  </ul>
  <h4>Circle Type list</h4>
  <ul type="circle">
    <li>HTML</li>
    <li>CSS</li>
    <li>PHP</li>
  </ul>
  <h4>Square Type list</h4>
  <ul type="square">
    <li>Pragaph</li>
    <li>Table</li>
    <li>List</li>
  </ul>
</body>
</html>
```

Example of Order list

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Order List Example</title>

</head>
<body>
  <h3>Alphabet Type list</h3>
  <ol type="A">
    <li>Home</li>
    <li>About Us</li>
    <li>Contact Us</li>
  </ol>
  <h3>Number Type list</h3>
  <ol type="1">
    <li>HTML</li>
    <li>CSS</li>
    <li>PHP</li>
  </ol>
  <h3>Roman Number Type list</h3>
  <ol type="I">
    <li>Pragraph</li>
    <li>Table</li>
    <li>List</li>
  </ol>
</body>
</html>
```

Example of Definition list

```
<!DOCTYPE html>
<html>
<head>
  <title>Definition List Example</title>

</head>
<body>
  <dl>
    <dt>UK</dt>
    <dd>United Kingdom</dd>
    <dt>USA</dt>
    <dd>United States of America</dd>
  </dl>
</body>
</html>
```

Link Tag Element কি?

Link এর আভিধানিক অর্থ সংযুক্ত করা। অর্থাৎ একটা ওয়েব পেজের সাথে অন্য একটা পেজকে যুক্ত করাই হল Linking। HTML এ Linking টা করা হয় `<a> ` tag দিয়ে। আর যার সাথে লিংকআপ টা করা হবে সেটা `<a>` tag এর href attribute এর মধ্যে দিতে হয়। যেমন: ` Link Text or Image or Video` আর এই লিঙ্কিং তা তিন ধরনের হতে পারে।

1. Inpage Linking
2. Internal Linking
3. External Linking

চলুন উপরের তিন প্রকার লিংক সম্পর্কে জানা যাক:

Inpage linking হচ্ছে সাধারণত পেজ এর মধ্যে বিভিন্ন পজিশন বা ট্যাগ এর সাথে লিংকআপ করা। যেমন : `Go Middle` অর্থাৎ , একই পেজ এর middle নামক id এর সাথে লিংকআপ করা। Internal Link হচ্ছে একই সার্ভার এর এক পেজ এর সাথে অন্য পেজ এর লিংকআপ করা। যেমন : `Go Test Page` অর্থাৎ , একই Server এর test.php নামক পেজ এর সাথে লিংকআপ করা।

External Link হচ্ছে ভিন্ন কোনো সার্ভার এর যেকোনো পেজ এর সাথে লিংকআপ করা। যেমন : `Go Test Page` অর্থাৎ , example.com Server এর test.php নামক পেজ এর সাথে লিংকআপ করা।

Image Tag Element কি?

ওয়েব পেজে ইমেজ বা ছবি যুক্ত করার জন্য প্রয়োজনীয় ট্যাগটি হচ্ছে ``। এর কোন শেষ ট্যাগ নেই। শুধুমাত্র `` দিয়ে কোন কাজ হয় না, এর সাথে সবসময়ই `src` এট্রিবিউটটি ব্যবহার করতে হয়। যেমন ``, তাহলে `pic.png` ইমেজটি প্রদর্শিত হবে। `` tag এ `src` attribute ছাড়াও আরো আছে `alt` attribute যা দিয়ে আপনি image কোনো কারণে না পেলে অলটারনেটিভ একটা লেখা দেখতে পারেন। এ ছাড়া `width` এবং `height` attribute দিয়ে ইমেজ এর `width` এবং `height` সেট করতে পারবেন।

Example Of image Tag Element

```
<!DOCTYPE html>
<html>
<head>
  <title>Image Tag Example</title>
</head>
<body>
  <h3> This is an example of image.</h3>
  
</body>
</html>
```

Text Formatting Element কি?

Microsoft Word এ Bold, Italic, Underline, strikethrough, Subscript, Superscript ইত্যাদি টেক্সট ফরমেটিং এর কথা নিশ্চয় আপনাদের মনে আছে ? HTML এর ক্ষেত্রেও Microsoft Word এর মত টেক্সট ফরমেটিং এর কাজ গুলো করা যায়। HTML এ টেক্সট ফরমেটিং এর জন্য , <i>, <u>, <strike>, <sub>, <sup>, <big>, <small>, , <samp>, <tt>, <abbr>, <var>, <code>, <address> ইত্যাদি ট্যাগ সমূহ ব্যবহার করা হয়।

Example of Text Formatting Element

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Formatting Tag Example</title>

</head>
<body>
  <p>
    <b> (Bold)</b> This is an example of bold. <br />
    <i>(Italic)</i> This is an example of Italic. <br />
    <u> (Underline) </u> This is an example of Underline. <br />
    <strike>(Strike) </strike> This is an example of Strike. <br />
    (CO<sub>2</sub>) This is an example of Subscript. <br />
    (E=MC<sup>2</sup>) This is an example of Superscript. <br />
    <big>(Big text)</big>This is an example of Big. <br />
    <small>(Small text)</small>This is an example of Small. <br />
    <strong>(Strong text)</strong>This is an example of Strong. <br />
    <samp>(Sample text) </samp>This is an example of Sample. <br />
    <tt>(Teletype)</tt>This is an example of Teletype. <br />
    (<abbr> U.N.O</abbr> United Nations Organization.) This is an example of Abbreviation. <br />
    (<var>x</var> is a variable.)This is an example of Variable <br />
    (<code>Computer code text.</code>) This is an example of Code.<br />

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

HTML div কি?

div মানে Division, একটা অংশ বা ভাগ। এইচটিএমএল পেইজের একটা সেকশন। এইচটিএমএল ইলিমেন্টকে বিভিন্ন গ্রুপে ভাগ করার জন্য div ট্যাগ ব্যবহার করা হয়।

HTML span কি?

div ডিফল্ট রূপে একটা আরেকটার নিচে বসে , অর্থাৎ আপনি যদি কোনো টেক্সট লাইনের মধ্যে নির্দিষ্ট অংশকে CSS Style করার জন্য একাধিক div ব্যবহার করেন , দেখবেন একটা div আরেকটা div এর নিচে বসে সংশ্লিষ্ট লাইনটিকে ভেঙে দিবে। এবং এক লাইনে আন্তে হলে আপনাকে আলাদা করে আবার CSS লিখতে হবে। আর এই কাজটি আপনি span দিয়ে করলে এই ধরনের সমস্যা হবে না। মূলত span ব্যবহার করা হয় একটা লাইনের মধ্যের একটা নির্দিষ্ট অংশে CSS Style করার জন্য।

Example of span tag Element

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Span</title>
</head>
<body>
  <p>This is a <span style="color:red;font-weight:bold">Example of span
tag</span>. The span tag is used to inline-elements CSS in a document. </p>

  </body>
</html>

</body>
</html>
```

HTML এ Table কি?

ওয়েব পেজে বিভিন্ন ধরনের তথ্য উপাত্ত পরিসংখ্যান রিপোর্ট আকারে উপস্থাপন করার একটি অন্যতম পন্থা হচ্ছে টেবিল ব্যবহার করা। টেবিল তৈরির জন্য টেবিল ট্যাগ `<table>` ব্যবহার করা হয়। একটি table কত গুলো row বা সারি তে বিভক্ত, সেটি উপস্থাপন করার জন্য `<tr>` tag ব্যবহার করা হয়। tr বলতে “table row” বুঝায়। আবার প্রত্যেকটা সারিতে কত গুলো data cells এ বিভক্ত, তা বুঝাতে `<td>` tag ব্যবহার করা হয়। td বলতে “table data,” বুঝায়। data cell এর কাজ হচ্ছে content কে সংরক্ষণ করা। একটি `<td>` tag text, links, images, lists, forms, other tables ইত্যাদি সংরক্ষণ করে।

HTML এ Table তৈরী করতে হলে কি জানা দরকার ?

ট্যাগ	ট্যাগের বিবরণ
<table>	টেবিল তৈরি করার জন্য ব্যবহার করা হয়।
<th>	টেবিলের হেডার তৈরি করার জন্য ব্যবহার করা হয়।
<tr>	টেবিলের সারি(row) তৈরি করার জন্য ব্যবহার করা হয়।
<td>	টেবিলের সেল বা ডেটা তৈরি করার জন্য ব্যবহার করা হয়।
<caption>	টেবিলের ক্যাপশন সেট করার জন্য ব্যবহার করা হয়।
<colgroup>	একের অধিক কলামকে গ্রুপ করার জন্য ব্যবহার করা হয়।
<col>	<colgroup> এলিমেন্টের মধ্যে কলাম প্রোপার্টি সেট করার জন্য ব্যবহার করা হয়।
<thead>	টেবিলের Header সেট করার জন্য ব্যবহার করা হয়।
<tbody>	টেবিলের body সেট করার জন্য ব্যবহার করা হয়।
<tfoot>	টেবিলের footer সেট করার জন্য ব্যবহার করা হয়।

Example of simple table

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table</title>
</head>
<body>
  <table border="1" width="300" height="100">
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Address</th>
    </tr>
    <tr>
      <td>Masuma Akter</td>
      <td>50</td>
      <td>Dhaka</td>
    </tr>
    <tr>
      <td>Rashed Khan</td>
      <td>55</td>
      <td>Khulna</td>
    </tr>
  </table>

</body>
</html>
```


HTML Table এ cellpadding এবং cellspacing কি?

HTML Table এ সেল এর বর্ডার থেকে তার content এর দূরত্ব কতটুকু হবে তা নির্ধারণের জন্য cellpadding ব্যবহৃত হয়। আর এক সেল থেকে অন্য সেল কতটুকু দূরত্বে বসবে তা নির্ধারণের জন্য cellspacing ব্যবহৃত হয়।

cellpadding সহ HTML Table

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table with Cellpadding</title>
</head>
<body>
  <table border="1" width="300" height="100" cellpadding="10">
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Address</th>
    </tr>
    <tr>
      <td>Masuma Akter</td>
      <td>50</td>
      <td>Dhaka</td>
    </tr>
    <tr>
      <td>Rashed Khan</td>
      <td>55</td>
      <td>Khulna</td>
    </tr>
  </table>

</body>
</html>
```

cellspacing সহ HTML Table

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table with cellspacing</title>
</head>
<body>
  <table border="1" width="300" height="100" cellspacing="10">
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Address</th>
    </tr>
    <tr>
      <td>Masuma Akter</td>
      <td>50</td>
      <td>Dhaka</td>
    </tr>
    <tr>
      <td>Rashed Khan</td>
      <td>55</td>
      <td>Khulna</td>
    </tr>
  </table>

</body>
</html>
```



HTML Table এ rowspan এবং colspan এর কাজ কি?

HTML Table এ একাধিক row কে একটা single cell এ রূপান্তর বা merge করার জন্য rowspan এবং একাধিক cell কে একটা single cell এ রূপান্তর বা merge করার জন্য colspan attribute ব্যবহৃত হয়।

Example of colspan

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table with colspan</title>
</head>
<body>
  <table border="1" width="300" height="100" border="1">
    <tr>
      <th colspan="3">Student Information</th>

    </tr>
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Address</th>
    </tr>
    <tr>
      <td>Masuma Akter</td>
      <td>50</td>
      <td>Dhaka</td>
    </tr>
    <tr>
      <td>Rashed Khan</td>
      <td>55</td>
      <td>Khulna</td>
    </tr>
  </table>

  </body>
</html>
```


Example of rowspan

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table with rowspan</title>
</head>
<body>
  <table border="1" width="300" height="100" border="1">
    <tr>
      <th colspan="3">Student Information</th>

    </tr>
    <tr>
      <th rowspan="4">Name</th>

    </tr>
    <tr>
      <td>Masuma Akter</td>
    </tr>
    <tr>
      <td>Rashed Khan</td>
    </tr>
    <tr>
      <td>Sojib Khan</td>
    </tr>
  </table>

</body>
</html>
```

html table caption

html table এ caption যুক্ত করতে হলে আপনাকে `<caption></caption>` টি ব্যবহার করতে হবে।

Example:

```
<table border="1" width="300" height="100" border="1">
  <caption>HTML Table Example with Caption</caption>
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>Address</th>
  </tr>
  <tr>
    <td>Masuma Akter</td>
    <td>50</td>
    <td>Dhaka</td>
  </tr>
  <tr>
    <td>Rashed Khan</td>
    <td>55</td>
    <td>Khulna</td>
  </tr>
</table>
```

html table এ কিভাবে header, body এবং footer এরিয়া নির্ধারণ করবেন?

html table এ header এরিয়া নির্ধারণ করার জন্য `<thead></thead>` tag , body এরিয়া নির্ধারণ করার জন্য `<tbody></tbody>` tag এবং footer এরিয়া এর জন্য `<tfoot></tfoot>` ব্যবহৃত হয়।

Example of table

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table with header,body and footer</title>
</head>
<body>
  <table border="1" width="300" height="100" border="1">

    <thead>
      <tr>
        <th>Name</th>
        <th>Age</th>
        <th>Address</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Masuma Akter</td>
        <td>50</td>
        <td>Dhaka</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td colspan="3"><center>This is Footer</center></td>
```

html এ Form কি?

সাইটের ব্যবহারকারীদের থেকে সরাসরি তথ্য নেয়ার যে মাধ্যম সেটাকে বলা হয় form। বিশেষ করে ডাইনামিক এবং ডাটাবেজ নির্ভর ওয়েব সাইটের লগ ইন সিস্টেম, ভোটিং সিস্টেম, কন্টাক্ট ফর্ম সহ ব্যবহারকারীর তথ্য সংগ্রহ করার জন্য এইচটিএমএল ফর্ম ব্যবহৃত হয়ে থাকে।

Form ট্যাগ দিয়ে user/visitor দের information collect করার পর আপনাকে Database এ backend script অর্থাৎ ASP, PHP or python ইত্যাদি দিয়ে process করে সংরক্ষন করতে পারবেন।

HTML এ `<form>—</form>` এই ট্যাগ এর মাধ্যমে code করতে হয়। Information process করতে বিভিন্ন Tags & Attributes ব্যবহার করতে হয়।

Form Attributes

form তৈরী করার জন্য ব্যবহৃত <form> tag এ বহুল বেবহত Attributes গুলো হল-

1. Action এই attribute এর মাধ্যমে backend script এ data pass করে।
2. Target এই attribute এর কাজ হল database এ pass হওয়া result টি new tab (window), frame অথবা current window তে show সেটাকে specify করা। এর value গুলো হল _self, _blank, _top এবং _parent
3. Enctype এই attribute টি data কে কিভাবে server এ submit/পাঠানোর পূর্বে encode করে সেটাই নির্ধারণ করে।

Form Controls

Visitor দেৱ Data collect কৰাৰ জন্য কিছু form control আছে, যেখানে input type পৰিবৰ্তন এৰ মাধ্যমে কি কি ধৰনেৰ data কিভাবে collect কৰবেন সেটা নিৰ্ধাৰণ কৰতে পাৰবেন। চলুন দেখা যাক-

1. Text Input Controls
2. Checkboxes Controls
3. Radio Box Controls
4. Select Box Controls
5. File Select boxes
6. Hidden Controls
7. Clickable Buttons
8. Submit and Reset Button

Text input control

Text Input Controls সাধারণত তিন ধরনের হয়ে থাকে যা আপনি HTML FORM এ ব্যবহার করতে পারবেন।

1. Single-line text input controls
2. Password input controls
3. Multiple-Line Text Input Controls

Single-line text input controls

এই Control টি user থেকে শুধু মাত্র এক লাইনে input নেওয়ার জন্য ব্যবহৃত হয়। যেমন, সার্চ বাক্স, নাম ইত্যাদি। আর এটি তৈরী করতে হয় HTML এর <input> tag ব্যবহার করার মাধ্যমে।

Example:

```
<form >
```

```
    First name: <input type = "text" name = "first_name" />
```

```
    Last name: <input type = "text" name = "last_name" />
```

```
</form>
```

Single-line text input controls এর attributes কি কি?

Sr.No	attribute এর নাম এবং তার ব্যবহার
১	type কি টাইপের input control সেটি নির্ধারণ করার জন্য ব্যবহৃত হয়। সাধারণত এটার value হিসেবে text সেট করা হয়।
২	name Input Control এ দেওয়া value টি সার্ভার এ কি নামে যাবে, বা সার্ভার থেকে কি নামে গ্রহণ হবে সেটা নির্ধারণের জন্য ব্যবহৃত হয়।
৩	value সার্ভার এ কোনো initial বা default value পাঠানোর জন্য ব্যবহৃত হয়।
৪	size characters এর ভিত্তিতে input control এর size নির্ধারণের জন্য ব্যবহৃত হয়।
৫	maxlength Text Control এ সর্বোচ্চ কতগুলো character ইনপুট দিতে পারবে, তা নির্ধারণের জন্য ব্যবহৃত হয়।

HTML Single line input tag

`<form >`

Input Type color: `<input type="color" name="favcolor">
`

Input Type Date: `<input type="date" name="bday">
`

Input Type (date and time): `<input type="datetime-local" name="bdaytime">
`

Input Type E-mail: `<input type="email" name="email">
`

Input Type Month: `<input type="month" name="bdaymonth">
`

Input Type Number: `<input type="number" name="quantity" min="1" max="5">
`

Input Type Range: `<input type="range" name="points" min="0" max="10">
`

Input Type Search: `<input type="search" name="googlesearch">
`

Input Type Telephone: `<input type="tel" name="usrtel">
`

Input Type Time: `<input type="time" name="usr_time">
`

Input Type URL: `<input type="url" name="homepage">
`

Input Type Week: `<input type="week" name="week_year">`

`</form>`

Password input controls

এইটাও একটা single-line text input , এটা User দেব Password collect করার জন্য ব্যবহৃত হয়, এই input control টি ব্যবহারের ফলে user দেব input করা character গুলো mask হয়ে যায় অর্থাৎ character/অক্ষর গুলি dot এর মত আসে যা আমরা সাধারণত password বক্স এ দেখতে পাই। এটি `<input type="password">` দিয়ে ডিফাইন করা হয়।

Example:

```
<form >
```

```
    Password: <input type = "password" name = "password" />
```

```
</form>
```


Multiple-Line Text Input Controls

একাধিক লাইনের input প্রধানের জন্য এই input method টি ব্যবহৃত হয়। এটি `<textarea></textarea>` দিয়ে define করা হয়। চলুন একটা উদাহরণ দিয়ে দেখা যাক :

```
<form>
```

Description :

```
<textarea rows = "5" cols = "50" name = "description">
```

Enter description here...

```
</textarea>
```

```
</form>
```

Multiple-Line Text Input Controls এর attributes কি কি?

Sr.No	attribute এর নাম এবং তার ব্যবহার
-------	----------------------------------

১	name Input Control এ দেওয়া value টি সার্ভার এ কি নামে যাবে, বা সার্ভার থেকে কি নামে গ্রহণ হবে সেটা নির্ধারণের জন্য ব্যবহৃত হয়।
২	rows input field টি কত row এর সমান height হবে তা নির্ধারণ করে।
৩	cols input field টি কত column এর সমান width হবে তা নির্ধারণ করে।

HTML Meta

ক্যারেক্টার সেট, পেজের বর্ণনা, কী-ওয়ার্ড, পেজের রচয়িতা এবং অন্যান্য মেটাডেটা নির্দেশ করার জন্য এলিমেন্ট ব্যবহার করা হয়।

ব্রাউজার, সার্চ ইঞ্জিন এবং অন্য ওয়েব সার্ভিসের জন্য মেটাডেটা ব্যবহার করা হয়।

Character Set নির্দেশ করতে:

```
<meta charset="UTF-8">
```

Web Page এর Description দিতে:

```
<meta name="description" content="Free HTML tutorials">
```

Search Engine এর জন্য Key words নির্দেশ করতে:

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

পেজের রচয়িতার নাম নির্দেশ করতে:

```
<meta name="author" content="Sahab Uddin">
```

ডকুমেন্ট কে প্রত্যেক ৩০ সেকেন্ডে রিফ্রেশ করতে:

```
<meta http-equiv="refresh" content="30">
```

HTML Script

জাভাস্ক্রিপ্ট, জে কোয়েরী ইত্যাদি স্ক্রিপ্ট যুক্ত করার জন্য `<script>` `</script>` ট্যাগ ব্যবহার করা হয়। যেমন:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML script Example</title>
  <script type="text/javascript" src="jquery.js"></script>
  <script type="text/javascript" src="sliding_effect.js"></script>
</head>
<body>

  <h2>HTML script Example</h2>
  <p>This is a Paragraph</p>

</body>
</html>
```

HTML Audio

এইচটিএমএল <audio> tag টি একটি ওয়েব পেইজে অডিও ফাইল প্লে করতে ব্যবহৃত হয়।

Example:

```
<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>
```



HTML Audio - How It Works

1. The **controls** attribute adds audio controls, like play, pause, and volume.
2. The **<source>** element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
3. The text between the **<audio>** and **</audio>** tags will only be displayed in browsers that do not support the **<audio>** element.

HTML Video

এইচটিএমএল <video> tag টি একটি ওয়েব পেইজে video ফাইল প্রদর্শন করতে ব্যবহৃত হয়।

Example

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
</video>
```



How it Works

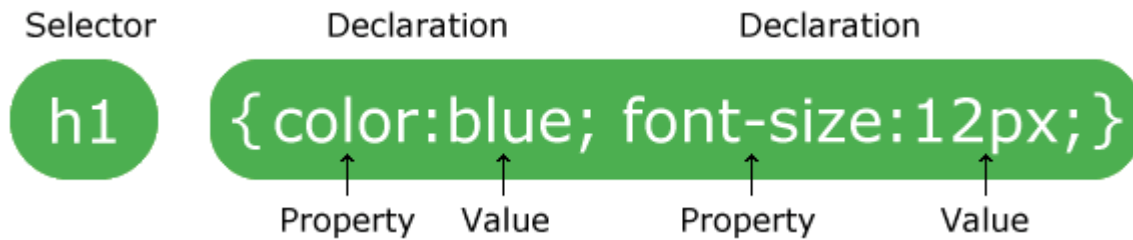
1. The **controls** attribute adds video controls, like play, pause, and volume.
2. It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.
3. The **<source>** element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
4. The text between the **<video>** and **</video>** tags will only be displayed in browsers that do not support the **<video>** element.



What is CSS

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS Syntax

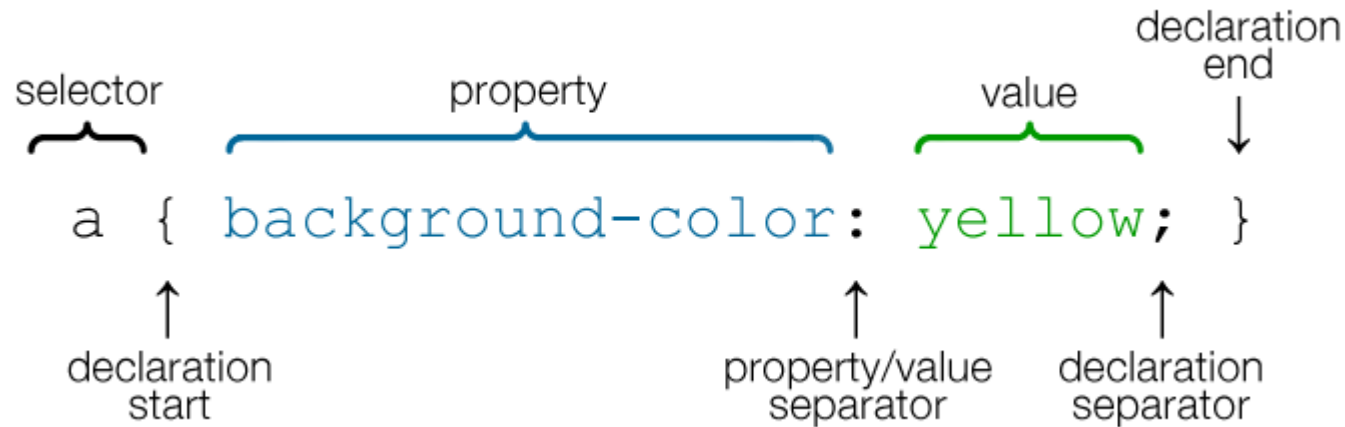


CSS Selector কি?

আপনি যে HTML Element কে Style করতে চান, সেই Element এর নাম বা tag, Id এবং class কে CSS Selector বলা হয়। সহজ কথায় যার উপর CSS style প্রয়োগ হবে, তাকে CSS Selector বলা হয়। আর সে গুলো হচ্ছে যথাক্রমে Element এর নাম বা tag, Id এবং class.

CSS এর Syntax কি?

Web Page এ CSS প্রয়োগ করার পদ্ধতিকে বলা হয় CSS Syntax. CSS Syntax এর দুটি অংশ আছে। একটি হচ্ছে Selector অংশ আর অন্যটি হচ্ছে Declaration অংশ। আর declaration অংশ আবার দুইভাবে বিভক্ত , প্রথম অংশ হচ্ছে property আর সেমিকোলন (:) এর পরের অংশকে বলা হয় property value.



CSS Syntax এর আরো কিছু বিষয় জানা যাক

- CSS এ এক বা একাধিক Declaration ব্লক থাকতে পারে। একের অধিক Declaration ব্লককে সেমিকোলনের (:) মাধ্যমে পৃথক করা হয়।
- প্রতিটি Declaration ব্লকে একটি CSS Property এর নাম ও একটি ভ্যালু থাকে এবং কোলনের মাধ্যমে এদেরকে পৃথক করা হয়।
- প্রতিটি সিএসএস Declaration ব্লক সর্বদাই সেমিকোলনের মাধ্যমে শেষ হয় এবং ডিক্লেয়ারেশন ব্লক/ব্লকসমূহ Curly Braces(দ্বিতীয় বন্ধনী) দ্বারা বেষ্টিত থাকে।



Web Page এ CSS ব্যবহার করার পদ্ধতি

- Inline CSS
- Inpage CSS
- Internal CSS
- External CSS

CSS এ @import কি?

CSS এ internal অথবা external যেকোনো। .css file কে অন্য আরেকটি .css file এ import করার জন্য এটি ব্যবহৃত হয়। তবে আপনি চাইলে এটিকে html ফাইলেও ব্যবহার করতে পারেন। যদিও html file এ performance এর বিবেচনায় link ট্যাগ ই উত্তম। এটির syntax দুইভাবে হয়। একটি হচ্ছে string হিসেবে আরেকটি হচ্ছে url হিসেবে import করা।

Example__

```
@import "navigation.css";
```

or

```
@import url("navigation.css");
```

CSS এ Unit কি?

CSS Style দিয়ে HTML Elements, Id এবং class গুলোর উপর width এবং height প্রকাশ করার মাধ্যম কে CSS Unit বলা হয়। অনেক CSS Property length ভ্যালু নেয়, যেমন- width, height, margin, padding, font-size, border-width ইত্যাদি। Unit একটি নম্বর যা বিভিন্ন length এর একক অনুসরণ করে, যেমন 10px, 2em ইত্যাদি।

CSS এ দু'ধরনের লেন্থ ইউনিট রয়েছে:

- **Relative**
- **Absolute.**

CSS Relative Lengths

Relative Length বলতে এক Length এর সাথে অন্য Length এর Property মিল বুঝায়।

ইউনিট	বর্ণনা
em	এলিমেন্টের font-size এর সাথে সম্পর্কযুক্ত (2em অর্থ 2 বার বর্তমান ফন্টের আকার)
ex	বর্তমান ফন্ট x-height এর সাথে সম্পর্কযুক্ত (কম ব্যবহৃত)
ch	“0” (শূন্য) প্রস্থের সাথে সম্পর্কযুক্ত
rem	রুট(Root) এলিমেন্টের font-size এর সাথে সম্পর্কযুক্ত
vw	ভিউপোর্ট*(viewport*) এর 1% প্রস্থের সাথে সম্পর্কযুক্ত।
vh	ভিউপোর্ট*(viewport*) এর 1% উচ্চতার সাথে সম্পর্কযুক্ত।
vmin	ভিউপোর্ট*(viewport*) এর 1% ছোট ডাইমেনশনের(dimension) সাথে সম্পর্কযুক্ত।
vmax	ভিউপোর্ট*(viewport*) এর 1% বড় ডাইমেনশনের(dimension) সাথে সম্পর্কযুক্ত।
%	

CSS Absolute Length

Absolute Length এর ইউনিট fixed এবং এর দৈর্ঘ্য তার ঠিক প্রকৃত সাইজেই হবে।

ইউনিট	বর্ণনা
cm	সেন্টিমিটার(centimeters)
mm	মিলিমিটার(millimeters)
in	ইঞ্চি(inches) (1in = 96px = 2.54cm)
px *	পিক্সেল(pixels) (1px = 1/96th of 1in)
pt	পয়েন্ট(points) (1pt = 1/72 of 1in)
pc	পিকাস(picas) (1pc = 12 pt)

CSS Text

1. Text Alignment ➡ text-align
2. Text Color ➡ color
3. Text Decoration ➡ text-decoration
4. Text Transformation ➡ text-transform
5. Text Indentation ➡ text-indent, letter-spacing, line-height, word-spacing
6. Text Shadow ➡ text-shadow

CSS Fonts

1. Font Families → font-family, family-name, generic-name;
2. Font Style → font-style, font-variant
3. Font Size → font-size
4. Google Fonts
5. font-family: family-name, generic-name;

CSS Links

- 1.`a:link` - a normal, unvisited link
- 2.`a:visited` - a link the user has visited
- 3.`a:hover` - a link when the user mouses over it
- 4.`a:active` - a link the moment it is clicked

- 1. color,
- 2. font-family,
- 3. background

CSS Lists

- **list-style-type** (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
- **list-style-position** (specifies whether the list-item markers should appear inside or outside the content flow)
- **list-style-image** (specifies an image as the list item marker)



CSS Tables

1. Table Borders
2. border-collapse
3. Table Width and Height
4. Horizontal Alignment -> text-align
5. vertical-align
6. Table Padding
7. Horizontal Dividers -> border-bottom
8. Hoverable Table
9. Table Color

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element

```
selector:pseudo-class {  
  property: value;  
}
```


Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(<i>language</i>)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every element that is the only child of its parent

CSS Pseudo-elements

A CSS pseudo-element is used to style specified parts of an element.

Selector	Example	Example description
<u>::after</u>	p::after	Insert content after every <p> element
<u>::before</u>	p::before	Insert content before every <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>	p::first-line	Selects the first line of every <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

CSS Borders

HTML Element গুলোর বর্ডারের স্টাইল, কালার এবং প্রস্থ পরিবর্তন করার জন্য CSS border প্রোপার্টি গুলো ব্যবহৃত হয়।

- border-color
- border-gradient
- border-image
- border-radius
- border-style
- border-width
- diffant-border

Border Property

- **border**
 - **border-width**
 - **border-style**
 - **border-color**
- **border-top**
 - **border-top-width**
 - **border-top-style**
 - **border-top-color**
- **border-right**
 - **border-right-width**
 - **border-right-style**
 - **border-right-color**
- **border-left**
 - **border-left-width**
 - **border-left-style**
 - **border-left-color**
- **border-bottom**
 - **border-bottom-width**
 - **border-bottom-style**
 - **border-bottom-color**
- **border-radius**
- **border-image**
- **border-image-source**
- **border-image-slice**
- **border-image-width**
- **border-image-outset**
- **border-image-repeat**

CSS Margins

একটি HTML Element এর বর্ডার এর বাহিরে কি পরিমান ফাকা জায়গা থাকবে তা সেট করার জন্য CSS Margin ব্যবহৃত হয়। আরো সহজভাবে বলা যায় HTML Element এর চারপাশে স্পেস রাখার জন্য CSS Margin Property ব্যবহার করা হয়।

- **margin-top**
- **margin-right**
- **margin-bottom**
- **margin-left**

এই Margin Property গুলো নিচের ভ্যালুসমূহ গ্রহণ করতে পারেঃ

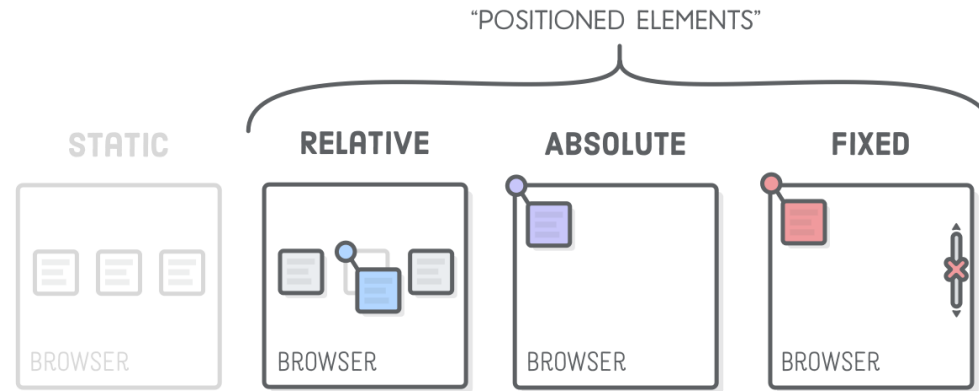
- **auto** – ব্রাউজার নিজ থেকে মার্জিন নিয়ে নেয়।
- **length** – px, pt, cm, ইত্যাদি একক ব্যবহার করে মার্জিন সেট করা হয়
- **%** – সংশ্লিষ্ট এলিমেন্টের প্রস্থ(width) অনুযায়ী মার্জিন সেট করা হয়।
- **inherit** – প্যারেন্ট(parent) ট্যাগ এর সাপেক্ষে উত্তরাধিকার সূত্রে মার্জিন নিয়ে নেয়।

CSS Padding

HTML Element এর বর্ডারের ভিতরে এবং কন্টেন্টের(content) চারপাশে ফাঁকা স্থান তৈরি করার পদ্ধতিকে CSS Padding বলা হয়।

- padding-একটি ডিক্লেয়ারেশনের মাধ্যমে সবগুলো প্যাডিং প্রোপার্টি সেট করার জন্য একটি শর্টহ্যান্ড প্রোপার্টি।
 - padding-top-এলিমেন্টের উপরের প্যাডিং সেট করার জন্য ব্যবহৃত হয়।
 - padding-right- এলিমেন্টের ডানের প্যাডিং সেট করার জন্য ব্যবহৃত হয়।
 - padding-bottom-এলিমেন্টের নিচের প্যাডিং সেট করার জন্য ব্যবহৃত হয়।
 - padding-left-এলিমেন্টের বামের প্যাডিং সেট করার জন্য ব্যবহৃত হয়।
-
- auto- ব্রাউজার নিজ থেকে Padding নিয়ে নেয়।
 - length- px, pt, cm, ইত্যাদি একক ব্যবহার করে Padding সেট করা হয়
 - % - সংশ্লিষ্ট এলিমেন্টের প্রস্থ(width) অনুযায়ী মার্জিন সেট করা হয়।
 - inherit- প্যারেন্ট(parent) ট্যাগ এর সাপেক্ষে উত্তরাধিকার সূত্রে মার্জিন নিয়ে নেয়।

CSS Positions



Web এ একটি HTML Element এর Position কোথায় হবে তা নির্ধারণের জন্যে CSS Position Property ব্যবহৃত হয়। আরো সহজ ভাবে বলা যায় একটি HTML Element এর Position সেট করার জন্য CSS Position Property ব্যবহার করা হয়।

CSS Positions

CSS এ Position Property এর ভ্যালুসমূহঃ

- static
- relative
- fixed
- absolute
- sticky

CSS এ Position Property এর গ্লোবাল(global) ভ্যালুসমূহঃ

- inherit
- initial
- unset



position:static;

ডিফল্টভাবে HTML Element সমূহের পজিশন static থাকে। আর তখন top, right, bottom এবং left প্রোপার্টির ভ্যালু সমূহ কাজ করেনা।

position:relative;

position:relative; ও static এর মতই পার্থক্য হচ্ছে এখানে আপনি চাইলে top, left, bottom, right এবং z-index ব্যবহার করে HTML Element কে তার অবস্থান থেকে সরাতে পারবেন। এমনকি অন্য একটা element এর উপর ও বসাতে পারবেন।

position:absolute;

position: absolute ও static এর মতই, পার্থক্য হচ্ছে এখানে HTML Element তার নিকটবর্তী পূর্বসূরী(ancestor) Element এর অবস্থানের সাপেক্ষে নিজের অবস্থানের পরিবর্তন ঘটে। এক্ষেত্রে , যদি Absolute পজিশনে থাকা এলিমেন্টের কোনো পূর্বসূরী না থাকে তাহলে ইহা ডকুমেন্ট বডি(body) কে পূর্বসূরী ধরে নিজের অবস্থানের পরিবর্তন ঘটাতে পারে এবং পেজ স্ক্রলিং এর সাথে সাথে নড়া চড়া করে।

position:fixed;

`position: fixed;` যুক্ত HTML Element গুলো viewport এর সাপেক্ষে নিজের পজিশনের পরিবর্তন ঘটাতে পার। অর্থাৎ পেজ স্ক্রল করলেও ইহা নিজের অবস্থান থেকে এক বিন্দুও নড়ে না।

position: sticky;

`position: sticky;` যুক্ত HTML Element ইউজারের পেজ স্ক্রলের সাপেক্ষে নিজের অবস্থানের পরিবর্তন হয়। তা ছাড়া স্ক্রল পজিশনের উপর ভিত্তি করে `position: sticky` একইসাথে Relative এবং Fixed পজিশনের মধ্যে toggle করে।

CSS Display

display:inline;

একাধিক html element কে একটা single লাইনে প্রদর্শনের display:inline ব্যবহৃত হয়। Inline Element গুলোতে আপনি left এবং right সাইড এ margin এবং padding ব্যবহার করতে পারেন, কিন্তু top এবং bottom এ padding এবং margin ব্যবহার করতে পারবেন না।

display:block

html element গুলোকে block আকারে প্রদর্শনের জন্য css এর display:block ব্যবহৃত হয়। আর html element গুলোর মধ্যে block element গুলো যেমন <p> এবং <div> tag গুলো বাই ডিফল্ট নিউ লাইনে বসে। আবার যেগুলো block element নয় সেগুলো যেমন এবং ট্যাগ গুলো পাশাপাশি বসে।

CSS Display

display:inline-block

`display:inline-block` অনেকটা `inline` এর মতোই , পার্থক্য হচ্ছে `display:inline-block` এ `width` এবং `height` কাজ করে , কিন্তু `display:inline` এ `width` এবং `height` কাজ করেনা।

display:contents

সরাসরি কোনো `html Element`, `Class` অথবা `Id` এর উপর ব্যবহৃত `style` কে `Omit` অর্থাৎ নিষ্ক্রিয় করার জন্য `display:contents` ব্যবহৃত হয়।

CSS Floating

HTML Element গুলিকে পাশাপাশি আনতে CSS Property float ব্যবহৃত হয়। float দিয়ে এলিমেন্ট গুলিকে পাশাপাশি (আড়াআড়িভাবে) বা side-by-side বসানো যায়। float সবসময় আড়াআড়িভাবে কাজ করে। HTML Layout তৈরীর সময় কোন div এর পাশে কোন div হবে তা ঠিক করার জন্য CSS Float ব্যবহৃত হয়।

চলুন CSS float property এর value গুলো সম্পর্কে সংক্ষেপে জানাযাক:

- float:left – এর মাধ্যমে element টি তার কন্টেইনার এর left এ থাকে।
- float:right– এর মাধ্যমে element টি তার কন্টেইনার এর right এ থাকে।
- float:none – অর্থাৎ float হবে না। এটি হচ্ছে float এর ডিফল্ট অবস্থা।
- float:inherit – এর মাধ্যমে parent এর float এর অবস্থা ইনহেরিট করবে।



CSS Opacity

The **opacity** property sets the opacity level for an element.
The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

CSS Syntax

```
opacity: number|initial|inherit;
```

Pseudo Class

বিভিন্ন উপাদানের অবস্থান, অবস্থা ইত্যাদি সিলেক্টর দ্বারা সিলেক্ট করে, বিশেষ ইফেক্ট দেয়ার জন্য সিউডো ক্লাস ব্যবহার করা হয়।

Syntax:

```
selector:pseudo-class {property:value;}
```

```
selector.class:pseudo-class {property:value;}
```

নেভিগেশন বারের বিভিন্ন মেনু বিভিন্ন অবস্থায় কেমন দেখাবে তা সিউডো ক্লাস দ্বারা নির্ধারণ করা হয়। এছাড়া বিভিন্ন চাইল্ড সিলেক্টরের ফাস্ট চাইল্ড সিলেক্ট করা সহ আরো অনেক ধরনের কাজ সিউডো ক্লাস দ্বারা করা যায়।

বিভিন্ন ধরনের সিউডো ক্লাস

- (Anchor Pseudo class)
- (First child Pseudo-class)
- (lang Pseudo-class)

Psudo Element

সিউডো ক্লাসের মতই বিভিন্ন উপাদানের অবস্থান, অবস্থা ইত্যাদি সিলেক্টর দ্বারা সিলেক্ট করে, বিশেষ ইফেক্ট দেয়ার জন্য সিউডো ইলিমেন্ট ব্যবহার করা হয়।

```
selector::pseudo-element {property:value;}
```

```
selector.class:pseudo-element {property:value;}
```

কোন প্যারাগ্রাফের প্রথম অক্ষর বা প্রথম লাইন কেমন দেখাবে তা সিউডো ইলিমেন্ট দ্বারা নির্ধারণ করা হয়। এছাড়া কোন প্যারাগ্রাফ বা হেডিং এর শুরুতে কোন ইমেজ, বা বিশেষ কোন টেক্সট যুক্ত করা সহ আরো অনেক ধরনের কাজ সিউডো ইলিমেন্ট দ্বারা করা যায়।

বিভিন্ন ধরনের সিউডো ইলিমেন্ট

- First letter Pseudoelement
- first line Pseudoelement
- Before Pseudo element
- After Pseudo element

CSS Media Types & Media query

বিভিন্ন মিডিয়া ধরন/ডিভাইসের জন্য বিভিন্ন স্টাইল ব্যবহার করতে @media Rule ব্যবহার করা হয়।
সিএসএস(২)-এ একে মিডিয়া ধরন বলা হতো, কিন্তু সিএসএস(৩)-তে এটাকে মিডিয়া কুয়েরি বলা হয়।
মিডিয়া কুয়েরি ডিভাইসের ক্ষমতা দেখে, এবং অনেক কিছু চেক করার জন্য ব্যবহার করা যেতে পারে,

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

media types প্রোপারটির ভ্যানু-সমূহ

- **all** - সকল মিডিয়া টাইপ ডিভাইসের জন্য ব্যবহার করা হয় ।
- **aural** - অননুমোদিত। বক্তৃতা এবং সাউন্ড সমন্বয়কারীদের জন্য ব্যবহার করা হয়।
- **braille** - অননুমোদিত। ব্রেইল স্পর্শীয় প্রতিক্রিয়া ডিভাইসের জন্য ব্যবহার করা হয়।
- **embossed** - অননুমোদিত। পেইজ ব্রেইল প্রিন্টারের জন্য ব্যবহার করা হয় ।
- **handheld** - অননুমোদিত। ছোট বা হ্যান্ডহোল্ড ডিভাইসের জন্য ব্যবহার করা হয়।
- **print** - প্রিন্টারের জন্য ব্যবহার করা হয়।
- **projection** - অননুমোদিত। প্রোজেক্ট উপস্থাপনার জন্য ব্যবহার করা হয় যেমনঃ স্লাইড।
- **screen** - কম্পিউটার স্ক্রিন, ট্যাবলেট, স্মার্টফোন ইত্যাদির জন্য ব্যবহার করা হয়।
- **speech** - স্ক্রিনরিডারদের জন্য ব্যবহার করা হয় যেটি উচ্চস্বরে পেজটি "পড়ে" ।
- **tty** - অননুমোদিত। টেলিটাইপ এবং টার্মিনাল মিডিয়ার মত একটি fixed-pitch অক্ষর গ্রিড ব্যবহার করতে ব্যবহার করা হয়।
- **tv** - অননুমোদিত। টেলিভিশন-ধরনের ডিভাইসের জন্য ব্যবহার করা হয়।

XHTML

XHTML কি

XHTML এর পূর্ণরূপ হল EXtensible Hyper Text Markup Language। এটি অবিকল HTML এর মত তবে কিছুটা কঠোর। XHTML সব বড় বড় ব্রাউজারে সাপোর্ট করে।

কেন XHTML ব্যবহার করবো?

HTML কোডিং এ কোন ভুল থাকলেও অনেক সময় বিভিন্ন ব্রাউজারে তা ধরা পড়েনা এবং ব্রাউজারে সুন্দর ভাবেই দেখা যায়। কিন্তু বর্তমানে বাজারে অনেক নতুন ব্রাউজার এসেছে। অনেকেই কম্পিউটার ছাড়াও মোবাইলে সাইট ভিজিট করছে। অনেক মোবাইল ব্রাউজার HTML এর ভুলগুলো সঠিকভাবে দেখাতে পারেনা। XHTML এ যেহেতু কোডিং সঠিকভাবে করতে হয় তাই XHTML ব্যবহার করাই শ্রেয়।

ডকুমেন্ট এর গঠন (Document Structure)

XHTML DOCTYPE আবশ্যিক XHTML কোডিং এর ক্ষেত্রে।

এর মধ্যে xmlns attribute ও আবশ্যিক।

<html>, <head>, <title>, <body> ও আবশ্যিক।

XHTML

```
DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title of document</title>
</head>
<body>
some content
</body>
</html>
```

XHTML

কিভাবে HTML থেকে XHTML রূপান্তর করবো?

- ১। প্রতিটি পেজ এর প্রথম লাইনে XHTML `<!DOCTYPE>` যুক্ত করতে হবে।
- ২। প্রতিটি পেজ এর HTML elements এর মধ্যে `xmlns attribute` যুক্ত করতে হবে।
- ৩। প্রতিটি elements গুলো **lowercase** বা ছোটহাতের করতে হবে।
- ৪। প্রতিটি elements সঠিকভাবে বন্ধ করতে হবে।
- ৫। Attribute names গুলো **lowercase** বা ছোটহাতের করতে হবে।
- ৬। attribute values গুলো সঠিকভাবে Quote করতে হবে।

Forms Controls

1. .form-control
2. .form-control-file
3. .form-control-sm
4. .form-control-range
5. .form-check
6. .form-check-input
7. .form-check-label
8. .form-check
9. .form-group
10. .is-invalid
11. .is-valid
12. .invalid-feedback



Image

1. `.img-fluid`
2. `.img-thumbnail`
3. `.rounded`



collapse

1. `.collapse`
2. `.collapsing`
3. `.collapse.show`
4. `.accordion`

JavaScript

JavaScript কি?

JavaScript যেটাকে আমরা সংক্ষেপে JS নামেই চিনি, যা একটি লাইট ওয়েট, ইন্টারপ্রেটেড (অর্থাৎ কম্পাইল করার প্রয়োজন নেই) এবং prototype-based, মাল্টি প্যারাডাইম, ডাইনামিক স্ক্রিপ্টিং ল্যাঙ্গুয়েজ যা একাধারে object-oriented, imperative, এবং declarative তথা functional programming স্টাইল সাপোর্ট করে।

যদিও জাভাস্ক্রিপ্টকে একটি ক্লাইন্ট সাইড স্ক্রিপ্টিং ল্যাঙ্গুয়েজ বা ব্রাউজার স্ক্রিপ্টিং ল্যাঙ্গুয়েজ বলা হয় কিন্তু বর্তমানে জাভাস্ক্রিপ্ট ক্লায়েন্ট সাইড অর্থাৎ ব্রাউজার এর বাহিরে সার্ভার সাইড ও নেটওয়ার্ক প্রোগ্রামিং, মোবাইল ও ডেস্কটপ অ্যাপ ডেভেলপমেন্ট, গেইম ডেভেলপমেন্ট থেকে শুরু করে ইন্টারনেট অব থিংস, মেশিন লার্নিং সহ আরো অনেক কাজেই ব্যবহৃত হচ্ছে। আর এটাকে ইন্টারমেডিয়েট ল্যাঙ্গুয়েজ ধরে এর উপর তৈরি হয়েছে বেশ কিছু যেমন CoffeeScript, TypeScript এর মতো সাব ল্যাঙ্গুয়েজ, এ ছাড়াও Apache CouchDB, MongoDB ইত্যাদি ডাটাবেইজ গুলো তাদের স্ক্রিপ্টিং ল্যাঙ্গুয়েজ এবং কুয়েরী ল্যাঙ্গুয়েজ হিসেবে জাভাস্ক্রিপ্ট ব্যবহার করে। আর node.js তো জাভাস্ক্রিপ্টকে ভিত্তি করে পরিপূর্ণ Desktop এবং Server Side Scripting Language হিসেবে নিজেকে প্রতিষ্ঠিত করেছে। এছাড়াও জাভাস্ক্রিপ্ট এর উপর ভিত্তি করে তৈরি হাজার হাজার লাইব্রেরী এবং ফ্রেমওয়ার্ক যেমন – AngularJS, VueJS, ReactJS, Express, JQuery বদলে দিয়েছে সফটওয়্যার ডেভেলপমেন্ট এর চেহারা। এবং Cross Mobile Platform গুলো যেমন Vue Native, React Native যেগুলো বদলে দিয়েছে Mobile Apps ডেভেলপমেন্ট এর ধারণা।

JavaScript এর সংক্ষিপ্ত পরিচিতি

১৯৯৫ সালে নেটস্কেপের প্রকৌশলী Brendan Eich জাভাস্ক্রিপ্ট তৈরি করেন, যা মুক্তি পায় ১৯৯৬ সালের শুরুর দিকে নেটস্কেপ ২ (ব্রাউজার) এর সাথে। এর নাম দেয়া হয়েছিল LiveScript, কিন্তু মার্কেটিং কৌশলের অংশ হিসেবে সান মাইক্রোসিস্টেম এর জাভা ল্যাংগুয়েজের জনপ্রিয়তাকে পুঁজি করার জন্য, জাভা আর জাভাস্ক্রিপ্টের মাঝে তেমন কোন মিল না থাকা সত্ত্বেও এর নাম রাখা হয় জাভাস্ক্রিপ্ট। মাইক্রোসফট এই প্রোগ্রামিং ভাষার সাথে প্রায় মিলে যায় এরকম একটি ল্যাংগুয়েজ JScript নাম দিয়ে প্রায় ৩ মাস পর ইন্টারনেট এক্সপ্লোরারের সাথে বাজারে নিয়ে আসে। এদিকে নেটস্কেপ Ecma International (স্ট্যান্ডার্ড নির্ধারণ করে এরকম একটি ইউরোপীয়ান সংস্থা) এর কাছে ল্যাংগুয়েজটি উপস্থাপন করে – যার ফলাফল ১৯৯৭ সালে ECMAScript এর প্রথম সংস্করণ হিসেবে বাজারে আসে। ১৯৯৯ সালে এই স্ট্যান্ডার্ডটি আরো উন্নত হয় ECMAScript সংস্করণ ৩ হিসেবে – আর সেই থেকে ভাষাটির তেমন কোন বড় পরিবর্তন হয়নি। চতুর্থ সংস্করণটি ভেঙে যায়, ভাষাটির জটিলতা নিয়ে মতবিরোধের ফলাফল হিসেবে। তবে এই চতুর্থ সংস্করণের অনেক অংশবিশেষ কে ভিত্তি হিসেবে ধরে ২০০৯ সালে নতুন ECMAScript এর পঞ্চম সংস্করণ প্রকাশ করা হয়। সর্বশেষ ২০১৫ সালের ১৭ জুন ECMA International তার ষষ্ঠ প্রধান সংস্করণ প্রকাশ করে, আনুষ্ঠানিক ভাবে যা ECMAScript 2015 নামে অভিহিত, এবং বর্তমানে এটি ECMAScript 6 বা ES6 নামে পরিচিত।

How to Write JS

JavaScript কে HTML Page এর মধ্যে <script> start tag এবং </script> end tag এর মধ্যে লিখে যেমন রান করা যায় , আবার আপনি চাইলে .js extension যুক্ত যেকোনো file এ লিখে সেটিকে পরবর্তীতে অন্য যেকোনো file যেমন .php, .html, .asp, .py সহ আরো অনেক ল্যাংগুয়েজ ফাইল এ সংযুক্তির মাধ্যমে সেই জাভাস্ক্রিপ্টকে রান করতে পারেন। এছাড়া যেকোনো html tag এর মধ্যে লিখেও আপনি JavaScript কে রান করতে পারেন। তো চলুন তিনটা পদ্ধতিই দেখা যাক:

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write("<h2>Hello JavaScript</h2>");
</script>
</body>
</html>
```


সরাসরি html tag এর মধ্যে

```
<!DOCTYPE html>  
<html>  
<body>  
<button onclick="document.getElementById('p').innerHTML = 'Hello World!'">Just Click Me</button>  
<button onclick="alert('Hey Dear!')">Click Again</button>  
<p id="p">Hello Bangladesh</p>  
</body>  
</html>
```


.js extension যুক্ত যেকোনো file

```
function changeMe(){  
  document.getElementById('p').innerHTML = 'Hello World!';  
}
```

```
function getAlert(){  
  alert('Hey Dear!');  
}
```

```
$(document).ready(function(){  
  $("p").click(function(){  
    $("p").hide();  
  });  
});
```

test.html file

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>
<button onclick="changeMe()">Just Click Me</button>
<button onclick="getAlert()">Click Again</button>
<p id="p">Hello Bangladesh</p>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script type="text/javascript" src="assets/js/script.js"></script>
</html>
```

JavaScript এর Output কে কয়ভাবে দেখানো যায়

1. `window.alert()` ব্যবহার করে এলার্ট বক্সের মাধ্যমে
2. `document.write()` ব্যবহার করে এইচটিএমএল ডকুমেন্টে লিখে
3. `.innerHTML` ব্যবহার করে এইচটিএলএল এলিমেন্টে লিখে
4. `console.log()` ব্যবহার করে ব্রাউজার কনসোলের মধ্যে দেখানো যায়

window.alert() ব্যবহার

```
<!DOCTYPE html>  
<html>  
<body>  
<script>  
window.alert("Hello I'm From w3programmers.com");  
</script>  
</body>  
</html>
```

document.write() ব্যবহার

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write("<b>Hey! do you know? It's Best Platform to learn JavaScript!</b>");
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<button type="button" onclick="document.write(5 + 6)">click for Summation</button>
</body>
</html>
```

innerHTML ব্যবহার

HTML Element কে access করার জন্য JavaScript এ document নামক class এর getElementById() Method টি ব্যবহৃত হয়।

HTML এর ট্যাগ গুলোর id এট্রিবিউটটি HTML এর একটা নির্দিষ্ট এলিমেন্টকে নির্দেশ করে এবং innerHTML Property এর মাধ্যমে যেকোনো HTML কন্টেন্টকে নির্দেশ করতে পারে। চলুন একটা উদাহরণ দিয়ে বুঝে নেয়া যাক:

```
<!DOCTYPE html>
<html>
<body>
<p id="sum"></p>
<script>
document.getElementById("sum").innerHTML = 5 + 6;
</script>
</body>
</html>
```


console.log () ব্যবহার

JavaScript Code গুলোকে ডিবাগিং করার জন্য অনেক সময় আমাদেরকে বিভিন্ন রকমের তথ্য প্রদর্শন করতে হয় , তখন আমরা এর console Class এর log () মেথডটি ব্যবহার করি। অবশ্য এর জন্য আপনাকে **F12** এর মাধ্যমে ব্রাউজার কনসোল সক্রিয় করতে হবে অথবা মেনু থেকে “Console” নির্বাচন করতে হবে। অথবা সরাসরি **Ctrl+Shift+I** ব্যবহার করতে পারেন। চলুন উদাহরণে দেখা যাক:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>অবশ্য এর জন্য আপনাকে F12 এর মাধ্যমে ব্রাউজার কনসোল সক্রিয় করতে হবে অথবা মেনু থেকে "Console" নির্বাচন করতে হবে। অথবা সরাসরি Ctrl+Shift+I ব্যবহার করতে পারেন।</p>
```

```
<script>
```

```
console.log(5 + 6);
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript Variable

JavaScript এ variable কি ?

JavaScript এ Variable হচ্ছে কম্পিউটারের মেমোরিতে অস্থায়ী তথ্য সংরক্ষণের পাত্র বা পাত্রের নাম , যা আমরা কম্পিউটারের মেমোরিতে যেকোনো অস্থায়ী তথ্য সংরক্ষণের জন্য ব্যবহার করে থাকি। কোন একটা variable এ একবার তথ্য রেখে সেটা পুরো কোডজুড়ে বারবার ব্যবহার করতে পারেন, মূল তথ্য(value) টি বারবার রাখার পরিবর্তে।

JavaScript Variable

JavaScript এ variable তৈরী করতে হলে ডলার (\$) চিহ্ন যুক্ত অথবা ডলার (\$) চিহ্ন ছাড়া যেকোনো ইংরেজি বর্ণ (লেটার) (a-z, A-Z) অথবা আন্ডারস্কোর (_) দিয়ে শুরু করতে হয়।

উদাহরণ:

```
<!DOCTYPE html>

<html><body><p id="x"></p><p id="z"></p><p id="r"></p>
<script>
a="America";
$a="Australia";
_a="Arab Amirat";
document.getElementById("x").innerHTML = a;
document.getElementById("z").innerHTML = $a;
document.getElementById("r").innerHTML = _a;
</script>
</body>
</html>
```

JavaScript Variable

- JavaScript এ variable এর নাম case sensitive.যেমন JavaScript এর দৃষ্টিতে, a (\$a) এবং A (\$A) দুটি আলাদা ভেরিয়েবল।
- JavaScript এ Variable নামে কোনো ফাঁকা (স্পেস) থাকা যাবেনা। যদি নাম একের অধিক হয় তাহলে “__”underscore (first_name) অথবা বড় হাতের অক্ষরে(firstName) লিখতে হবে।
- এছাড়া JavaScript এ কোনো ভ্যারিয়েবল কে শুধু ফাংশন এর মধ্যে সীমাবদ্ধ (local variable) রাখার জন্য variable এর সামনে **var keyword** টি ব্যবহার করতে হয়। অন্যথায় variable টি global variable হিসেবে বিবেচিত হবে।
- JavaScript এ Variable এ যেকোনো value রাখতে হলে আপনাকে assignment operator (=) টি ব্যবহার করতে হবে।
যেমনঃ `var a=5;`
- JavaScript এ একটা statement এ একাধিক variable কে (comma (,) Seperated রেখে) ঘোষণা করতে পারেন।

JavaScript Variable (Let keyword)

JavaScript এ variable declare করার জন্য var কীওয়ার্ড এর মতোই let নামে আরেকটি keyword আছে , যার কাজ হচ্ছে variable টিকে function scope বা global scope এর পরিবর্তে block scope { } হিসেবে ব্যবহার করা যায়। অর্থাৎ আপনি চাইলে একই কোড পেজে অথবা একই ফাংশনের মধ্যে block দিয়ে একাধিক variable declare করতে পারেন। আর let কীওয়ার্ড টি ECMAScript 2015 অর্থাৎ ES6 থেকে JavaScript এ অন্তর্ভুক্ত হয়।

JavaScript Variable (Let keyword)

```
<!DOCTYPE html><html><body><script>
function test(){
  a=5;
  if(a==5){
    let a=7;
    document.write(a);
  }
  document.write("<br>");
  document.write(a);
}
test();
{
  let a=8;
  document.write("<br>");
  document.write(a);
}
</script></body></html>
```


JavaScript এ constant

JavaScript এ constant হচ্ছে Variable এর মতই কোনো একটা সাধারণ স্থায়ী Value বা মানের জন্য Identifier বা নাম, Constant এর সাথে Variable এর মূল পার্থক্য হচ্ছে Constant কে একবার Define করলে আর পরিবর্তন করা যায় না অথবা দ্বিতীয়বার আর declare বা ঘোষণা করা যাবে না। যেখানে Variable এর মান কে সমস্ত কোড জুড়ে যেকোনো জায়গায় পরিবর্তন করা যায়।

JavaScript এ constant

JavaScript এ **const** কীওয়ার্ড দিয়ে constant ঘোষণা করতে হয়। আর **const** কীওয়ার্ড টি ECMAScript 2015 অর্থাৎ ES6 থেকে JavaScript এ অন্তর্ভুক্ত হয়। একটি Valid Constant নাম শুরু হয় লেটার(letter) বা (__) Underscore দ্বারা। চাইলে Constant নামের পূর্বে ডলার(\$) সাইন দেওয়া যাবে। JavaScript এ Constant এর নাম case sensitive.যেমন JavaScript এর দৃষ্টিতে, a এবং A দুটি আলাদা কনস্ট্যান্ট। Constant নামে কোনো ফাঁকা (স্পেস) থাকা যাবেনা।যদি নাম একের অধিক হয় তাহলে “_”underscore (first_name) অথবা বড় হাতের অক্ষরে(firstName) লিখতে হবে। তা ছাড়া JavaScript এ Constant ঘোষণা দেওয়া কোনো নামকে পরবর্তীতে variable হিসেবেও ব্যবহার করা যাবেনা।

JavaScript constant

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
try {
  const a="America";
  a="Africa";
}
catch (err) {
  document.getElementById("demo").innerHTML = err;
}
</script>
</body>
</html>
```

JavaScript Comments

JavaScript অথবা যেকোনো Programming Language এ Comments হচ্ছে, একজন কোডার বা প্রোগ্রামারের সোর্স কোডের ব্যাখ্যা বা পাদটীকা। এটাকে আমরা Coding Documentation ও বলতে পারি। সাধারণত: যেকোনো Programming language এর Compiler এবং Interpreter গুলো comments কে কোড হিসাবে execute না করে এটাকে (Ignore) এড়িয়ে যায়। Comments System দিয়ে আমরা Coding Documentation ছাড়াও এই মুহূর্তে ব্যবহৃত হবেনা কিন্তু ভবিষ্যতে কাজে লাগতে পারে এমন Code গুলোও আমরা সাময়িক সময়ের জন্য hide করে রাখতে পারি।

Type Of Comments

- Single Line Comments (প্রত্যেক line এর জন্য আলাদা ভাবে comment চিহ্ন ব্যবহার করতে হয়)
- Multi Line Comments (একসাথে একাধিক লাইনে এর জন্য শুধু একবার কমেন্ট চিহ্ন ব্যবহার করতে হয়।)



JavaScript Operator

JavaScript এ Operators হচ্ছে কতগুলো symbol যা JavaScript Language দিয়ে কিছু নির্দিষ্ট mathematical, relational এবং logical সহ আরো অনেক ধরনের কার্য (operation) সম্পাদন করার নির্দেশনা পাঠায়। এবং নির্দেশনা অনুযায়ী JavaScript সেই নির্দিষ্ট কাজের ফলাফল প্রদর্শন করে।

Type of JavaScript Operators

- **Arithmetic Operators** বা গাণিতিক কাজের Operator: (+,-,*,/,%,**)
- **Assignment Operators** বা JavaScript Variable এ যেকোনো মান (Value) সংরক্ষণের Operator : (=)
- **Bitwise Operators** বা JavaScript এ Binary কাজের Operator :-(&,|,!,^,~,<<,>>,<<=,>>=,>>>=,&=, ^=,|=)
- **Comparison Operators** বা দুই বা ততোধিক Value এর মধ্যে তুলনা করার Operator :-(==,!=,===,!==,<,>,<=,>=)
- **Incrementing/Decrementing Operators** বা 1 করে বৃদ্ধি বা কমানোর Operator:-(++,- -)
- **Logical Operators** বা একাধিক অবস্থার বা condition এর উপর ভিত্তি করে সিদ্ধান্ত নেওয়ার Operator:-(&&,||,!)
- **String Operators** বা এক বা একাধিক string এর সাথে জোড়া লাগানোর Operator:-(+)
- **Type Operators** বা কোনো একটা JavaScript Variable কোনো class এর Instance কিনা তা চেক করার operator:- (instanceof) বা কোনো একটা variable বা value এর type কি তা চেক করার জন্য typeof Operator
- **Ternary Operator** বা if..else কার্য সম্পাদন করার অপারেটর।
- **delete Operator:** বা কোনো একটা Object এর নির্দিষ্ট property delete করার জন্য delete operator ব্যবহৃত হয়।
- **in Operator:** Object এর নির্দিষ্ট property খুঁজে বের করার জন্য ব্যবহৃত হয়।

JavaScript Operand

Operator এর দুই পাশে যেই সব variable অথবা value থাকে, JavaScript অথবা যেকোনো Programming Language এর পরিভাষায় একে Operand বলে।

```
<!DOCTYPE html>
<html>
<body>
<script>
var a=5;
document.write(a+5);
</script>
</body>
</html>
```

এখানে plus sign (+) এর দুই পাশের যথাক্রমে a এবং 5 এই দুইটা ই Operand আর plus sign (+) টি হচ্ছে Operator

Operator Precedence

JavaScript এ Operator Precedence হচ্ছে কিছু rules বা নিয়মের সমষ্টি যা দিয়ে JavaScript এ যে কোনো কাজে কোন কাজটি আগে সম্পাদিত হবে তা নির্ধারিত হয়।

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(1 + 5 * 3); //Output 16 But why not 18?
</script>
</body>
</html>
```

ব্যাখ্যা: উপরের কোডের ফলাফল 18 না এসে 16 আসবে। এর মূল কারণ হচ্ছে, আমাদের এখানে ব্যবহৃত দুটি operator এর মধ্যে Operator Precedence Rules অনুযায়ী addition (“+”) Operator টির চেয়ে multiplication (“*”) Operator টি বেশি অগ্রাধিকার প্রাপ্ত।

JavaScript array

JavaScript এ array এক ধরনের data type যা একটি single variable এ একাধিক value সংরক্ষণের সুযোগ দেয়।

আরেকটু ভালো ভাবে বুঝে নেয়া যাক , ধরুন আপনার কাছে যদি একাধিক value এর একটি list থাকে (উদাহরণস্বরূপঃ SAtech360 ছাত্রদের নামের লিস্ট) এবং সেই লিস্ট যদি শত সহস্র হয়ে থাকে , তাহলে সেগুলোকে আলাদা আলাদা ভ্যারিয়েবলের মধ্যে রেখে পরবর্তীতে ব্যবহার অনেক কষ্ট সাধ্য। আর এই কাজটি আমরা array তে খুব সহজে করতে পারি। কারণ array একই নামের অধীনে একাধিক ভ্যালু রাখতে পারে এবং আপনি index নম্বর ব্যবহার করে ভ্যালু এক্সেস করতে পারেন।

Note: JavaScript array এর index গুলো শুধুমাত্র numerical হয়ে থাকে, এবং কোনো ধরনের associative array বা string index support করেনা।

JavaScript এ কিভাবে array তৈরি করা হয় ?

- array literal উপায়ে
- সরাসরি array class এর Object তৈরি করে (new keyword ব্যবহার করে)

JavaScript এ array literal উপায়ে array তৈরি

Syntax:

```
var array_name = [item1, item2, ...];
```

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Arrays</h2>
<p id="student_list"></p>
<script>
var students = ["Azhar", "Sahin", "Rakib","Abdullah","Shanto"];
document.getElementById("student_list").innerHTML = students;
</script>
</body>
</html>
```

new keyword ব্যবহার করে

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Arrays</h2>
```

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

```
<script>
```

```
var students = new Array("Azhar", "Sahin", "Rakib","Abdullah","Shanto");
```

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

```
</script>
```

```
</body>
```

```
</html>
```


JavaScript এর কোন পদ্ধতিতে array declaration ভালো ?

এক কোথায় প্রথম পদ্ধতিতে অর্থাৎ `[]` দিয়ে এবং `new Array()` দিয়ে এরে তৈরী না করাই ভালো। এর কারণ হলো অনেক সময় `new Array()` দিয়ে তৈরী কোনো array তে শুধু একটা value থাকে তখন সেটি আমাদেরকে unexpected রেজাল্ট দেখায়।



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Arrays</h2>
```

```
<p>Avoid using new Array().</p>
```

```
<p id="demo1"></p>
```

```
<p id="demo2"></p>
```

```
<script>
```

```
var numbers1 = new Array(40);
```

```
var numbers2 = [50];
```

```
document.getElementById("demo1").innerHTML = numbers1[0];
```

```
document.getElementById("demo2").innerHTML = numbers2[0];
```

```
</script>
```

```
</body>
```

```
</html>
```

ব্যাখ্যা: লক্ষ্য করবেন, আমাদের প্রথম numbers1 array টি new Array দিয়ে তৈরী , এবং সেখানে শুধু একটা value থাকায় আমাদের কে undefined index error দেখাচ্ছে। কিন্তু দ্বিতীয় array numbers2 তে এই ধরনের কোনো সমস্যা নাই।

length property

JavaScript এ একটা array তে কতগুলো এলিমেন্ট আছে তা জানতে হলে আপনাকে JavaScript এর array Object এর length property টি ব্যবহার করতে হবে।

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

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

<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.length;
</script>

</body>
</html>
```

toString() Method and join() Method

JavaScript এ একটা array এর value গুলোকে string আকারে প্রদর্শন করতে হলে আপনাকে toString() Method অথবা join() Method ব্যবহার করতে হবে। এক্ষেত্রে toString() মেথড বাই ডিফল্ট array এর value গুলোকে কমা (,) দিয়ে প্রদর্শন করে , অন্যদিকে join() Method এ আপনি চাইলে নিজস্ব separator ব্যবহার করতে পারেন।

toString() Method and join() Method

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p id="demo1"></p>
```

```
<p id="demo2"></p>
```

```
<script>
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
document.getElementById("demo1").innerHTML = "<b>toString() Method Result:</b> "+fruits.toString();
```

```
document.getElementById("demo2").innerHTML = "<b>join() Method Result:</b> "+fruits.join("*");
```

```
</script>
```

```
</body>
```

```
</html>
```


JavaScript এ একটা array তে কিভাবে নতুন Element যুক্ত অথবা remove করবেন?

JavaScript এ একটা array তে কোনো Element যুক্ত অথবা রিমুভ করার জন্য JavaScript এর push,pop,shift,unshift এবং splice এই মেথড গুলো ব্যবহৃত হয়।

JavaScript push() Method

```
<!DOCTYPE html>
<html>
<body>
  <button onclick="pushValue()">Add value to the end</button>
  <p id="demo"></p><p id="demo1"></p>
  <script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = "<b>Current Array Before Push:</b> "+ fruits;

function pushValue() {
  fruits.push("Kiwi");
  document.getElementById("demo1").innerHTML = "<b>Current Array After Push:</b> "+fruits;
}
</script></body></html>
```

JavaScript pop() Method

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

<button onclick="popValue()">Remove value from end</button>
<p id="demo"></p><p id="demo1"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = "<b>Current Array Before POP:</b> " + fruits;

function popValue() {
  fruits.pop();
  document.getElementById("demo1").innerHTML = "<b>Current Array After Pop:</b> " + fruits;
}
</script></body></html>
```

JavaScript shift Method

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

<button onclick="shiftValue()">Remove value from Start</button>
<p id="demo"></p>
<p id="demo1"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = "<b>Current Array Before Shift:</b> "+ fruits;

function shiftValue() {
  fruits.shift();
  document.getElementById("demo1").innerHTML = "<b>Current Array After Shift:</b> "+fruits;
}
</script></body></html>
```

JavaScript unshift Method

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

<button onclick="unShiftValue()">Add value to Start</button>
  <p id="demo"></p><p id="demo1"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = "<b>Current Array Before Unshift:</b> "+ fruits;

function unShiftValue() {
  fruits.unshift("Guava","Pineapple");
  document.getElementById("demo1").innerHTML = "<b>Current Array After Unshift:</b> "+fruits;
}
</script></body></html>
```



JavaScript Control Structure

JavaScript এ Control Structure হচ্ছে প্রোগ্রাম বা Code গুলোর নিয়ন্ত্রণের কাঠামো অথবা এমন কিছু statement যা দিয়ে আমরা Program Flow Control করে থাকি বা Program এর পরবর্তী করণীয় কি তা নির্ধারণ করে দেই।

Control Structure List

- **Selection/Branching Statement**
 - **Conditional Statement**
 - **if**
 - **if...else**
 - **if...elseif/else if...n...else**
 - **switch Statement**
 - **Unconditional Statement**
 - **Continue**
 - **break**
 - **return**
- **Loops**
 - **for**
 - **while**
 - **do...while**
 - **for/in**

if statement

JavaScript এ if statement হচ্ছে, কোন একটি condition বা শর্ত পূরণ সাপেক্ষে Program এর কোন অংশ execute হবে কি হবে না ? তা নির্ধারণ করে দেয়। আরেকটু সহজ ভাবে বলি, ধরুন আপনি দোকানে যাবেন আপেল ফল কেনার জন্য, এবং সিদ্ধান্ত নিলেন যদি আপেল ফল না পান তাহলে ফিরে আসবেন বা তার পরিবর্তে অন্য কোনো ফল কিনবেন না। বাস্তব জীবনের এই চিত্র টি ই JavaScript এ if statement দিয়ে সমাধান করা হয়।

JavaScript এ If Statement কিভাবে কাজ করে?

- JavaScript এ if Statement তখন ই কাজ করবে, যখন কোনো Logical Condition true হয়,
- যখন logical Condition false হয় , browser if Statement ব্লক এর মধ্যের কোড গুলোকে skip বা এড়িয়ে যায়।
- If Statement কে One way decision বা একমুখী সিদ্ধান্ত ও বলা হয়।

if Statement Syntax

```
if(condition)
{
  here goes statements;
  .
  .
  .
  .
  here goes statements;
}
```

if Statement

```
<script>
var age = 18;
if(age > 17){
document.write("you are eligible for voting ");
}
document.write("<br>This is normal flow ");
</script>
```

```
<script>
var age = 18;
if(age > 17 && age<=30){
document.write ("you are eligible for apply BCS Job");
}
document.write("<br>This is normal flow ");
</script>
```

if-else Statement

JavaScript এ else statement হচ্ছে if এর বিপরীত অর্থাৎ যদি কোনো কারণে if statement শর্ত পূরণ করতে ব্যর্থ হয় অথবা if statement যদি true না হয়।, তাহলে Program তার বিকল্প কোন অংশ execute হবে, তা নির্ধারণ করার জন্য যেই statement ব্যবহৃত হয়, তাকে বলা হয় else statement । আরেকটু সহজ ভাবে বলি, ধরুন আপনি দোকানে যাবেন আপেল অথবা কমলা ফল কেনার জন্য, এবং সিদ্ধান্ত নিলেন যদি আপেল ফল না পান তাহলে আপেলের পরিবর্তে কমলা ফল কিনে নিয়ে আসবেন। বাস্তব জীবনের এই চিত্র টি ই JavaScript এ if-else statement দিয়ে সমাধান করা হয়।

JavaScript এ If-else Statement কিভাবে কাজ করে?

JavaScript এ যখন if statement এর logical Condition false হয় , browser if Statement ব্লক এর মধ্যের কোড গুলোকে skip বা এড়িয়ে যায় এবং else Statement ব্লক এর মধ্যের কোড গুলোকে run করবে।

Syntax

```
if(condition)
{
here go statements....
}
else
{
here go statements....
}
```


if-else Statement Example

```
<script>
var age=16;
if(age > 17)
{
document.write("<br>you are eligible for voting ");
}
else
{
document.write("<br>Sorry, you are not eligible for voting ");
}
document.write("<br>This is normal flow ");
</script>
```

if-else Statement Example

```
<script>
var age = 40;
if(age > 17 && age<=30){
document.write("you are eligible for apply BCS Job");
}
else{
document.write("<br>Sorry, you are not eligible for BCS Job");
}
document.write("<br>This is normal flow ");
</script>
```

switch statement

JavaScript এ switch হচ্ছে else if statement এর মতোই , পার্থক্য হচ্ছে else if statement ক্রমান্বয়ে সবগুলো condition বা শর্ত চেক করতে থাকে এবং প্রথম যেই শর্ত পূরণ হবে , তার প্রোগ্রাম execute করার সুযোগ দেয়। এবং পরবর্তী condition গুলো আর চেক করেনা। আর switch ক্রমান্বয়ে সবগুলো condition বা শর্ত চেক করার পরিবর্তে যেই কন্ডিশন মিলে তা লাফ দিয়ে execute করে। অন্য কোনো condition চেক করেনা।

Syntax

```
switch ( variable )  
{  
  case value1:  
    statement;  
    break;  
  case value2:  
    statement;  
    break;  
  default:  
    statement;  
}
```

Example

```
<script>
var action = "ADD";
switch (action) {
case "ADD":
document.write("Perform actions for adding.");
break;
case "MODIFY":
document.write("Perform actions for modifying.");
break;
case "DELETE":
document.write("Perform actions for deleting.");
break;
}
</script>
```

for loop

JavaScript এ for loop হচ্ছে একটি নির্দিষ্ট কাজ একটি নির্দিষ্ট শর্ত বা condition পূরণ না হওয়া পর্যন্ত বার বার করতে থাকা বা execute করে। for loop এ variable এর value initialization, condition এবং increment এর কাজ একই লাইনে করা হয়।

for loop Syntax

```
for (initialization; condition; increment/decrement){  
    code to be executed;  
}
```




```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Loops</h2>
```

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

```
<script>
```

```
var text="";
```

```
for (i = 1; i <=10; i++) {
```

```
    text += i + "<br>";
```

```
}
```

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

```
</script>
```

```
</body>
```

```
</html>
```

For/In loop

JavaScript এ For/In loop বিশেষ ভাবে JavaScript array এবং object এর প্রত্যেকটি element এ loop এর কাজ করার জন্য ব্যবহৃত হয় ।

```
<script>
var txt = "";
var person = ["Arif","Rafiq","Shafiq"];
var x;
for (x in person) {
    txt += x + ". " + person[x] + "<br>";
}
document.getElementById("demo").innerHTML = txt;
</script>

</body>
</html>
```

while loop

JavaScript এ while loop হচ্ছে for loop এর মতোই অর্থাৎ :একটি নির্দিষ্ট কাজ একটি নির্দিষ্ট শর্ত বা condition পূরণ না হওয়া পর্যন্ত বার বার করতে থাকা বা execute করে। পার্থক্য হচ্ছে while loop এ variable এর value initialization, condition এবং increment এর কাজ ভিন্ন ভিন্ন লাইনে করা হয়।

while loop Syntax

```
initialization  
while (condition){  
    statement  
    increment  
}
```

while loop

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var text = "";
var i = 0;
while (i < 10) {
  text += "<br>The number is " + i;
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

do-while loop

JavaScript এ do-while loop হচ্ছে while loop এর মতোই অর্থাৎ :একটি নির্দিষ্ট কাজ একটি নির্দিষ্ট শর্ত বা condition পূরণ না হওয়া পর্যন্ত বার বার করতে থাকা বা execute করে। while loop এ variable এর value initialization, condition এবং increment এর কাজ ভিন্ন ভিন্ন লাইনে করা হয়। পার্থক্য হচ্ছে, **while loop এ condition true না হলে while এর code block execute করতে পারেনা। এবং কোনো ফলাফল ও দেখায়না। কিন্তু do-while এ condition true না হলেও ১ বার code block execute করে, এবং ফলাফল দেখাতে পারে।**

do while loop Syntax

```
do {  
    // This code is run at least once  
} while ( condition )  
  
// This code is run after the loop finishes
```



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

```
<script>
```

```
var text = ""
```

```
var i = 0;
```

```
do {
```

```
  text += "<br>The number is " + i;
```

```
  i++;
```

```
}
```

```
while (i < 10);
```

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

```
</script>
```


Date time Method

Type	Example
ISO Date	"2015-03-25" (The International Standard)
Short Date	"03/25/2015"
Long Date	"Mar 25 2015" or "25 Mar 2015"

Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.



Function

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

- Function **parameters** are listed inside the parentheses () in the function definition.
- Function **arguments** are the **values** received by the function when it is invoked.

Function Return

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

```
<script>
```

```
var x = myFunction(4, 3);
```

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

```
function myFunction(a, b) {
```

```
    return a * b;
```

```
}
```

```
</script>
```

jQuery

জেকোয়েরি হচ্ছে জাভাস্ক্রিপ্টের একটা ফ্রেমওয়ার্ক বা ফাংশন লাইব্রেরি। এখানে শত শত ফাংশন আগে থেকেই তৈরী করা আছে যা আপনি ব্যবহার করে আপনার সাইটকে আরও প্রানবন্ত করতে পারেন। এই কাজগুলি শুধু raw জাভাস্ক্রিপ্ট ব্যবহার করেও করতে পারবেন তবে এতে অনেক বেশি কোড লিখতে হবে এবং প্রচুর সময় লাগবে। জেকোয়েরির মত আরও অনেক জাভাস্ক্রিপ্টের ফ্রেমওয়ার্ক আছে যেমন Mootools, Extjs, Dojo, Prototype ইত্যাদি। তবে জেকোয়েরি এবং এরপর মোটুলস এখন সবচেয়ে বেশি বিখ্যাত। জেকোয়েরি বিখ্যাত হওয়ার অনেক কারনের মধ্যে রয়েছে

1. -সব ব্রাউজার সাপোর্ট করে
2. -সহজ এবং বিস্তারিত ডকুমেন্টেশন বিদ্যমান
3. -প্রচুর প্লাগিন বিনামূল্যে পাওয়া যায়।
4. -সিএসএস ৩ এর সিলেক্টর সাপোর্ট
5. এছাড়াও আরও অনেক সুবিধা আছে।

jQuery

জেকোয়েরি দিয়ে যেসব করা হয়:

1. বিভিন্ন সাইটে ড্রপডাউন মেনু দেখেছেন তো? এসব সাধারণত জেকোয়েরি দিয়ে করা।
2. ইমেজ স্লাইডশো
3. ট্যাব সিস্টেম

*যদিও জেকোয়েরি ব্রাউজার স্ক্রিপ্টিং তবুও এখানে এজাক্স ব্যবহার করে ডেটাবেস থেকে ডেটা তুলে আনা যায়, সাইট পুনরায় লোড হওয়া ছাড়াই। যেমন বিভিন্ন সাইটে দেখবেন রেজিস্ট্রেশনের সময় মেইল ফিল্ডে মেইল ঠিকানা দেয়ার সাথে সাথেই বলে দেয় যে এই মেইল ঠিকানা ইতোমধ্যে ব্যবহার করা হয়েছে বা এই ইউজার নাম আগেই নেয়া হয়েছে। এসব জেকোয়েরি এবং এজাক্স দিয়ে করা যায়।

jQuery

```
$(document).ready(function(){  
$(".myheader").hide();  
$(".mypara").click(function(){  
$(".myheader").show();  
});  
});  
</script>
```

যেহেতু এইচটিএমএল এর এলিমেন্ট নিয়েই জেকোয়েরির কাজ তাই এইচটিএমএল আগে লোড করে তারপর জেকোয়েরিকে এক্সিকিউট করানো উচিত। এখানে প্রথম লাইন দ্বারা এই কাজটিই হচ্ছে অর্থ্যাৎ `$(document).ready()`; দ্বারা বলা হচ্ছে যে আগে পুরো এইচটিএমএল ডকুমেন্ট টুকু লোড হবে তারপর জেকোয়েরির ফাংশনগুলি কাজ শুরু করবে। (তাই দেখুন `ready` ফাংশনের প্রথম বন্ধনীর মধ্যে বাকি সবগুলি কোড বা ফাংশনগুলি রাখা হয়েছে)

jQuery Element Selection

একটা এইচটিএমএল ডকুমেন্টের যেকোন এলিমেন্ট বা কোন এলিমেন্টের আইডি/ক্লাস দিয়ে যেমন আমরা সেই এলিমেন্ট সিলেক্ট করে সেটার জন্য সিএসএস রুল লিখে থাকি।জেকোয়েরিতেও হুবহু একইভাবে একটা এলিমেন্টকে সিলেক্ট করতে হয়

```
<script src="/jquery_latest.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function(){
$('p').click(function(){
alert('You clicked on a paragraph');
});
});
</script>
```

```
<h1 id="tutorial">jQuery Tutorials</h1>
<p id="first_para">Bangladesh is a country of natural beauty</p>
<p class="satech"> DOT com is the largest bengali</p>
<p class="satech">There are 30 tutorial on JQuery in</p>
```

jQuery Event

ওয়েব পেজে ইউজার কখনও মাউস ক্লিক করে(কোন একটা লিংক বা যেকোন কিছু উপর), কিবোর্ড দিয়ে টাইপ করে (ফর্ম থাকলে), কখনও কোন একটা এলিমেন্টের উপর মাউস নিয়ে যেতে (মাউস হোভার) পারে এটা কোন লিংক, প্যারাগ্রাফ, ইনপুট ফিল্ড বা যেকোন কিছু উপরে হতে পারে। এগুলি হচ্ছে এক একটা ইভেন্ট। যেমন কোন একটা প্যারাগ্রাফ বা বাটনের উপর ক্লিক করলেন, তাহলে এটা একটা ইভেন্ট (ক্লিক ইভেন্ট)। কোন একটা এলিমেন্টের উপর মাউস নিয়ে গেলেন এটা একটা ইভেন্ট (মাউস হোভার ইভেন্ট)। এরূপ আরো ইভেন্ট আছে যেমন scroll ইভেন্ট, load ইভেন্ট, focus ইভেন্ট ইত্যাদি।

jQuery Event Handler

ওয়েব পেজে এমন অনেক ইভেন্ট ঘটে, এই ইভেন্টগুলি ধরে আমরা ইচ্ছে করলে কোন কাজ করতে পারি, কোন জেকোয়েরির কোড এক্সিকিউট করাতে পারি। জেকোয়েরিতে অনেক ফাংশন আছে যেগুলি শুধু এসব কোন ইভেন্ট ঘটলে এক্সিকিউট হবে এই ফাংশনগুলিকে বলে **ইভেন্ট হ্যান্ডলার**। `ready()`, `click()`, `hover()` এরূপ আরো অনেক ইভেন্ট হ্যান্ডলার আছে। যখনি একটা ইভেন্ট ঘটল, এটাকে টেকনিকাল ভাষায় বলে অমুক ইভেন্ট fire হল। একটা ইভেন্ট fire হওয়ার সাথে সাথে ঐ ইভেন্ট সংশ্লিষ্ট হ্যান্ডলার এর ভিতর কোড এক্সিকিউট হয়।

কোন একটা এলিমেন্ট সিলেক্ট করে এরপর যদি `ready()` ইভেন্ট হ্যান্ডলার ব্যবহার করেন তাহলে ঐ এলিমেন্টটি পেজে সম্পূর্ণ লোড হওয়ার পর `ready()` এর ভিতর কোডগুলি কাজ করা শুরু করবে। যদি এলিমেন্ট লোড হওয়া পূর্ণ না হয় তাহলে এর ভিতরের কোড কাজ করবেনা বরং অপেক্ষা করবে। সাধারনত জেকোয়েরির কোডের শুরুতেই এই হ্যান্ডলার ব্যবহার করে অর্থাৎ পুরো পেজ সম্পূর্ণ লোড হবে এরপর জেকোয়েরির অন্যসব কোডগুলি কাজ করা শুরু করবে।

অনুরূপ অন্যান্য ইভেন্ট হ্যান্ডলার এর ক্ষেত্রেও প্রযোজ্য যেমন `$('#test').click()` এভাবে থাকলে এর অর্থ হচ্ছে test আইডি সম্বলিত এলিমেন্টটি তে ক্লিক করার পর `click()` ইভেন্ট হ্যান্ডলার এর ভিতরের কোড এক্সিকিউট হবে।

ইভেন্ট	ইভেন্ট হ্যান্ডলার
ready	এটা ডকুমেন্ট ইভেন্ট। একটা ডকুমেন্ট (পেজ) সম্পূর্ণ লোড হয়ে রেডি হলে এরপর এই ইভেন্ট ফায়ার হয়। সাধারনত সব জেকোয়েরির কোড এই ইভেন্ট হ্যান্ডলার এর ভিতরে থাকে কারন আমরা চাই, পেজ রেডি হওয়ার পর আমাদের অন্যান্য জেকোয়েরি কাজ শুরু করুক। \$(document).ready(function(){//code..});
click	যেকোন এলিমেন্টে ক্লিক করলে এই ইভেন্ট ফায়ার হবে। যেমন "test" আইডি সম্বিলত এলিমেন্টে ক্লিক করলে যদি কিছু করতে চাই তাহলে \$('#test').click(function(){any jQuery code here});
submit	যেকোন ফর্ম সাবমিট করলে submit ইভেন্ট ফায়ার হবে। যেমন "test" আইডি সম্বিলত ফর্ম সাবমিট করলে যদি কিছু করতে চাই তাহলে \$('#test').submit(function(){any jQuery code here});
blur	এটাও একটা ফর্ম ইভেন্ট, ইনপুট বক্স (ফর্ম এলিমেন্ট) থেকে মাউস সরিয়ে অন্য কোথাও ক্লিক করলে (ফোকাস চলে যাবার সাথে সাথে) blur ইভেন্ট ফায়ার হয়। যেমন "test" আইডি ইনপুট বক্সে ব্লার করলে যদি কিছু করতে চাই তাহলে \$('#test').blur(function(){any jQuery code here});
change	ফর্ম ইভেন্ট। ফর্ম এলিমেন্টের মান (যেমন ড্রপ ডাউন অপশন) পরিবর্তন করার সময় ফায়ার হয়। যেমন "test" আইডি সম্বিলত ড্রপডাউন থেকে অপশন পরিবর্তন করলে যদি কিছু করতে চাই তাহলে \$('#test').change(function(){any jQuery code here});
focus	ফর্ম ইভেন্ট। ফর্ম এলিমেন্টে যখন ফোকাস হবে তখন ফায়ার হয়। যেমন "test" আইডি সম্বিলত এলিমেন্টে মাউস ফোকাস করলে যদি কিছু করতে চাই তাহলে \$('#test').focus(function(){any jQuery code here});
select	যেকোন এলিমেন্ট সিলেক্ট করার সময়ই ফায়ার হয়। যেমন "test" আইডি সম্বিলত এলিমেন্ট সিলেক্ট করলে যদি কিছু করতে চাই তাহলে \$('#test').select(function(){any jQuery code here});।
dblclick	এটা মাউস ইভেন্ট। কোন এলিমেন্টে ডাবল ক্লিক করলে এটা ফায়ার হবে। যেমন "test" আইডি সম্বিলত এলিমেন্টে ডাবল ক্লিক করলে যদি কিছু করতে চাই তাহলে \$('#test').dblclick(function(){any jQuery code here});
mouseover	মাউস ইভেন্ট। কোন এলিমেন্টের উপর মাউস নিয়ে গেলে ফায়ার হয়। যেমন "test" আইডি সম্বিলত এলিমেন্টে মাউস হোভার করলে যদি কিছু করতে চাই তাহলে \$('#test').hover(function(){any jQuery code here});
load	এটা উইন্ডো ইভেন্ট। ফায়ার হয় যখন পেজ লোড শেষ হয়। যেমন উইন্ডো লোড হলে যদি কিছু করতে চাই \$(window).load(function(){any jQuery code here});
resize	এটাও উইন্ডো ইভেন্ট। ফায়ার হয় যদি উইন্ডো রিসাইজ করা হয় তখন। যেমন উইন্ডো রিসাইজ করার সময় যদি কিছু করতে চাই \$(window).resize(function(){any jQuery code here});

jQuery conflict removal

অনেক সময় সব কোড ঠিক মত লিখবেন কিন্তু তারপরেও দেখবেন কাজ হচ্ছেনা। সাধারনত কোন প্রজেক্টে একসাথে জাভাস্ক্রিপ্টের অনেক লাইব্রেরি ব্যবহার করলে এমন হতে পারে। এ সমস্যার একটা সহজ সমাধান আছে জেকোয়েরিতে। মোটুলস, প্রোটোটাইপ বা যতগুলি ফ্রেমওয়ার্ক অথবা জাভাস্ক্রিপ্টের অন্য যতই লাইব্রেরি ব্যবহার করেননা কেন, জেকোয়েরির কোড তার নিজের মত কাজ করবে। কারও দ্বারা প্রভাবিত হবেনা। এরূপ ক্ষেত্রে জেকোয়েরির কোড লেখার সময় নিচের মত করে লিখতে হবে।

jQuery conflict removal

```
var abc = jQuery.noConflict();
abc(document).ready(function(){
  abc('table tr').each(function(i){
    if(i < 5){
      if(i % 2 == 0){
        abc(this).addClass('even');
      }else{
        abc(this).addClass('odd');
      }
    }
  });
});
```


jQuery conflict removal

- abc ভেরিয়েবলের নাম ইচ্ছেমত দিতে পারেন। পরবর্তীতে জেকোয়েরির সব কোডে যেখানে যেখানে \$ চিহ্ন আছে তা এই নাম দিয়ে প্রতিস্থাপন করতে হবে।
- জাভাস্ক্রিপ্টের অনেক লাইব্রেরী \$ এই চিহ্ন তাদের ভেরিয়েবল বা ফাংশন নামের জন্য ব্যবহার করে তাই একসাথে একাধিক ফ্রেমওয়ার্ক ব্যবহার করলে এই ধরনের সমস্যা হতে পারে।
- j ছোটহাতের এরপর Q বড়হাতের, এভাবেই লিখতে হবে নাহলে কাজ হবেনা।

ব্যাখ্যা:এখানে এইজটিএমএল এবং সিএসএস দিয়ে অনেক গুরুত্বপূর্ণ কাজ হয়ে গেছে তবে যেহেতু এগুলি শিখে এসেছেন তাই শুধু এখানে জেকোয়েরির কোডে যেসব নতুন মেথড ব্যবহার করেছি সেগুলির ব্যাখ্যা দিচ্ছি।

১ নম্বর লাইনে ডকুমেন্ট রেডি হয়েছে এরপর ২ নম্বর লাইনে #menu আইডি সম্বলিত যে ul আছে তার অধীনস্থ li সিলেক্ট করেছি।যখন এই li এর উপর মাউস হোভার হবে তখন শুধুমাত্র এই li এর অধীনে যদি কোন ul (অর্থ্যাৎ সাবমেনু) থাকে তাহলে সেটি find() মেথড দিয়ে খুঁজে বের করেছি এবং এরপর slideDown() মেথড ব্যবহার করে সেই ul দৃশ্যমান করা হয়েছে।এই মেথডটির ভিতর যে নাম্বার দেখছেন এটা হচ্ছে গতি।যত কম নাম্বার দিবেন গতি তত বেশি হবে।

**এখানে দেখুন ডট (.) দিয়ে একটা একটা করে মেথড সিলেক্টেড এলিমেন্টের উপর চালানো হয়েছে এটা হচ্ছে মেথড চেইনিং।এভাবে ডট দিয়ে দিয়ে যতগুলি প্রয়োজন ততগুলি মেথড চালাতে পারেন একটা সিলেক্টেড এলিমেন্টের উপর।

**প্রথম এবং দ্বিতীয় বন্ধনীর ব্যাপারটি জেকোয়েরিতে সবচেয়ে গুরুত্বপূর্ণ এবং বিরক্তিকর (প্রথম প্রথম)।কয়েক ডজন কোড লেখার পর অবশ্য আর না বুঝতে অসুবিধা হবে আর না লিখতে।

এখন মাউস সরিয়ে নিলে কি হবে এটা লেখা হয়েছে ৪ নম্বর লাইনে কমার (,) পর।এই ফাংশনটি ঐ ul টিই খুঁজে বের করে slideUp() মেথড দিয়ে উপরে উঠিয়ে নেবে।

৭ নম্বর লাইন থেকে যে কোডগুলি লিখেছি এগুলি অতিরিক্ত কিছু স্টাইলিং এর জন্য।এখানে children() দিয়ে li এর উপর হোভার করার সময় চেক করে দেখা হচ্ছে যে এই li এর অধীনে কোন সাবমেনু আছে কিনা।যদি থাকে তাহলে সেই li তে rem_radius নামে একটি ক্লাস যোগ করা হয়েছে এবং সেই ক্লাস সম্বলিত li এর a ট্যাগের বর্ডার রেডিয়াস মুছে দেয়া হয়েছে (শুধু নিচের দিকে ডানে এবং বামে)।

jQuery form validation

যেকোন সাইটের এপ্লিকেশন নিয়ে কাজ করার সময় সাধারনত ফর্ম লাগে।রেজিস্ট্রেশন ফর্ম,জরিপ ফর্ম,লগিন ফর্ম ইত্যাদি।যাইহোক এরূপ ফর্ম থাকলে ভেলিডেশন রাখতে হয়।যদি সার্ভার সাইড (পিএইচপি) ভেলিডেশন রাখেন তাহলে এটা ইউজারের জন্য বেশ বিরক্তিকর।যতক্ষন পর্যন্ত ডেটাবেস থেকে কোন মান চেক করতে না হয় ততক্ষন সার্ভার সাইড ভেলিডেশন না রাখাই ভাল।জেকোয়েরি দিয়েই সব ধরনের ভেলিডেশন করা সম্ভব।

jQuery form validation

আমি যে ফর্ম তৈরী করেছি সেটার আইডি reg_form.8 নম্বর লাইনে এই ফর্ম সাবমিট হচ্ছে কিনা তা submit মেথড দিয়ে চেক করা হচ্ছে। যখনি সাবমিট বাটনে ক্লিক করবে সাথে সাথে কয়েকটি ফাংশন এক্সিকিউট হবে। যদি সবগুলিই ফাংশন true রিটার্ন করে তাহলে ফর্ম সাবমিট হবে তানাহলে এই if স্টেটমেন্ট false রিটার্ন করবে এবং ফর্ম সাবমিট না হয়ে কি করতে হবে তা দেখাবে। এখানে ৩টি ফাংশন লিখেছি।

প্রথম ফাংশন: এই ফাংশনটির প্যারামিটার fieldName এবং id (১৩-৩৯ লাইন)

১৫-১৭ নম্বর লাইনের অর্থ হচ্ছে যে টেক্সট ফিল্ডটির নাম (name এট্রিবিউটের মান) আর্গুমেন্ট হিসেবে পাঠাবে সেই টেক্সট ফিল্ডটি সিলেক্ট করে এর মান val() মেথড দিয়ে বের করে এনে length প্রোপার্টি দিয়ে চেক করে দেখবে যে ১ এর চেয়ে ছোট কিনা (অর্থ্যাৎ ১ এর চেয়ে ছোট হলে ০, তার মানে কিছুই টাইপ করেনি), যদি হয় তাহলে text() মেথড ব্যবহার করে ঐ টেক্সট ফিল্ডের নিচে যে span ট্যাগ আছে সেখানে "Please enter any text" এই লেখাটি দৃশ্যমান করবে (ফাংশন কল করার সময় যে span এর আইডি দিবেন সেই span এ লেখাটি আসবে)। এবং সবশেষে error নামে একটা ক্লাস যোগ করে false রিটার্ন করে ফর্ম সাবমিট হতে দিবেনা। আর যদি কিছু টাইপ করে তাহলে উপরোক্ত লেখাটি error ক্লাস সহ (যদি থাকে) উঠে গিয়ে true রিটার্ন করবে এবং ফর্ম সাবমিট হবে।

jQuery form validation

দ্বিতীয় ফাংশন: ইমেইল ভেলিডেশন এর জন্য (৪২-৬৪ লাইন)

একই ব্যাখ্যা, এখানে শুধু test মেথড দিয়ে চেক করা হয়েছে যে ইনপুটকৃত মান বৈধ ইমেইল কিনা। filter এখানে রেগুলার এক্সপ্রেশন, যেকোন বৈধ ইমেইল এই ফিল্টারে আটকে যাবেনা।

তৃতীয় ফাংশন: নামের ভেলিডেশন (৬৭-৯৪ লাইন)

আগের দুটোর মতই, এখানে একটা নতুন মেথড আছে তাহল isNaN(). এই মেথড দিয়ে ইনপুটকৃত মান নামের কিনা দেখা হচ্ছে। যদি নামের হয় তাহলে false রিটার্ন করে।

এখানে মাত্র কয়েকটি ফিল্ড তাই এভাবে ফাংশন না লিখে সরাসরি কোড লেখাই ভাল ছিল। তবে যদি অনেক বড় ফর্ম হয় তাহলে প্রতিটি ফিল্ডের জন্য আলাদা আলাদা কোড না লিখে এভাবে একই ফাংশন শুধু প্যারামিটার পরিবর্তন করে কাজ করা যায় এবং এটাই উত্তম।

PHP

PHP হচ্ছে একটি সার্ভার সাইড স্ক্রিপ্টিং ভাষা, যা তৈরী করেন রাসমাস লারডোর্ফ ১৯৯৪ সালে। যা বিশেষ ভাবে যেকোনো ডাইনামিক এবং ইন্টারেক্টিভ ওয়েব ডেভেলপমেন্ট অথবা ওয়েব ভিত্তিক সফটওয়্যার এবং ওয়েব ভিত্তিক এপ্লিকেশন বানানোর জন্য ব্যবহৃত হয়। বর্তমানে এটিকে জেনারেল পারপাস প্রোগ্রামিং ল্যাঙ্গুয়েজ ও বলা হয়। অর্থাৎ, অন্যান্য প্রোগ্রামিং ভাষা দিয়ে যা যা করা যায়, PHP দিয়েও মোটামুটি তার সবই করা যায়। এছাড়া PHP ওপেনসোর্স হওয়ায় ওয়েব প্রোগ্রামারদের কাছে অধিক জনপ্রিয়। তবে মূলত, ওয়েব ডেভেলপমেন্টের, ওয়েব ভিত্তিক সফটওয়্যার এবং ওয়েব ভিত্তিক এপ্লিকেশন বানানোর কাজেই এটি সর্বাধিক ব্যবহৃত হয়। এছাড়া পি এইচপি ওয়েবে তথ্য সংরক্ষণের ক্ষেত্রে সর্বোচ্চ নিরাপত্তা প্রদান করে। পি এইচ পি কে এইচ টি এম এল (HTML) এর সাথেও ব্যবহার করা যায়। ওয়ার্ডপ্রেস, জুমলা সহ বেশ কিছু এডভান্সড ওপেনসোর্স কনটেন্ট ম্যানেজমেন্ট সিস্টেম (CMS) পি এইচ পি দ্বারা তৈরি। পি এইচ পি ব্যবহার করে কাস্টম কনটেন্ট ম্যানেজমেন্ট সিস্টেম তৈরি করা যায়। এছাড়া ই-কমার্স, ম্যানেজমেন্ট, অনলাইন ব্যাংকিং সফটওয়্যার ইত্যাদি তৈরিতেও পি এইচ পি ব্যাপকভাবে ব্যবহৃত হচ্ছে। পি এইচ পি খুবই জনপ্রিয় একটি ভাষা, w3techs এর তথ্য অনুযায়ী পৃথিবীতে যত ওয়েব সাইট আছে তার শতকরা ৮২ ভাগই সার্ভার সাইড স্ক্রিপ্টিংয়ে PHP ব্যবহার করে। বর্তমানে বাংলাদেশের ৯০% ফ্রীল্যান্সার (অর্থাৎ যারা ঘরে বসে বাহিরের দেশে কাজ করে) PHP ল্যাঙ্গুয়েজ দ্বারা কাজ করে।

PHP শেখার আগে কোন বিষয়ে আমার জানা অপরিহার্য

- ☐ কমপিউটার প্রোগ্রামিং কি?
- ☐ ওয়েব কি?
- ☐ ইন্টারনেট কি?
- ☐ ডেটাবেস কি?
- ☐ HTML কি?
- ☐ CSS কি?

PHP echo And print Statement

PHP তে সার্ভার থেকে কোনো কিছু অথবা যেকোনো ফলাফল ব্রাউজারে দেখাতে চাইলে আপনাকে echo অথবা print দুটির যেকোনো একটি ব্যবহার করতে হবে। PHP তে এই দুইটা হচ্ছে Language Construct । আর language construct গুলো সাধারণত [function](#) থেকে দ্রুততর হয়। কেননা language construct গুলো কোনো রকম কম্পাইল ছাড়াই প্রয়োজনীয় ফলাফল দেখাতে পা রে, অন্য দিকে function গুলো কম্পাইল হওয়ার পর ফলাফল দেখাতে পারে।

```
<?php  
echo "Hello world";  
echo ("Hello World");  
print "Hello World";  
print ("Hello World");  
?>
```

echo এবং print এর মধ্যে পার্থক্য

1. echo তে আপনি একাধিক parameter নিতে পারবেন, যদি parenthesis ব্যবহার না করেন। অন্যদিকে print এ আপনি শুধু একটি parameter নিতে পারবেন, হোক সেটা parenthesis সহ অথবা parenthesis ছাড়া।
2. echo এর কোনো return value নেই। অন্যদিকে print boolean true মানে 1 রিটার্ন করে।

```
<?php  
echo print (5); // Output 51  
?>
```

এখানে print এর মাধ্যমে 5 কে ফলাফল হিসাবে পাই। তারপর print এর return ফলাফল 1 আমরা echo এর মাধ্যমে পাই। যার জন্য আমরা এখানে ফলাফল হিসাবে একত্রে 51 পাই।

PHP Comments

PHP Comments: PHP অথবা যেকোনো Programming Language এ Comments হচ্ছে, একজন কোডার বা প্রোগ্রামারের সোর্স কোডের ব্যাখ্যা বা পাদটীকা। এটাকে আমরা Coding Documentation ও বলতে পারি। সাধারণত: যেকোনো Programming language এর Compiler এবং Interpreter গুলো comments কে কোড হিসাবে execute না করে এটাকে (Ignore) এড়িয়ে যায়। Comments System দিয়ে আমরা Coding Documentation ছাড়াও এই মুহূর্তে ব্যবহৃত হবেনা কিন্তু ভবিষ্যতে কাজে লাগতে পারে এমন Code গুলোও আমরা সাময়িক সময়ের জন্য hide করে রাখতে পারি।

Type of PHP Comments

1. Single Line Comments (প্রত্যেক line এর জন্য আলাদা ভাবে comment চিহ্ন ব্যবহার করতে হয়) #, //
2. Multi Line Comments (একসাথে একাধিক লাইনে এর জন্য শুধু একবার কমেন্ট চিহ্ন ব্যবহার করতে হয়।) /**/

PHP Variable

PHP Variable হচ্ছে কম্পিউটারের মেমোরিতে অস্থায়ী তথ্য সংরক্ষণের পাত্র বা পাত্রের নাম , যা আমরা কম্পিউটারের মেমোরিতে যেকোনো অস্থায়ী তথ্য সংরক্ষণের জন্য ব্যবহার করে থাকি। কোন একটা variable এ একবার তথ্য রেখে সেটা পুরো কোডজুড়ে বারবার ব্যবহার করতে পারেন, মূল তথ্য(value) টি বারবার রাখার পরিবর্তে। PHP তে variable ঘোষণা করতে হলে ডলার (\$) চিহ্ন এর পর যেকোনো ইংরেজি বর্ণ (লেটার) (a-z, A-Z) অথবা আন্ডারস্কোর (_) দিয়ে শুরু করতে হয়। PHP তে variable এর নাম case sensitive.যেমন পিএইচপি এর দৃষ্টিতে, \$a এবং \$A দুটি আলাদা ভেরিয়েবল। Variable নামে কোনো ফাঁকা (স্পেস) থাকা যাবেনা। যদি নাম একের অধিক হয় তাহলে “_”underscore (\$first_name) অথবা বড় হাতের অক্ষরে(\$firstName) লিখতে হবে। একটা variable এর মান স্ক্রিপ্টে যেকোন সময় পরিবর্তন হতে পারে।

Type Of Variable

- Predefined Variables বা PHP language কর্তৃক পূর্ব নির্ধারিত variable
- User Defined Variables বা Programmer বা Developer কর্তৃক নির্ধারিত variable

Predefined Variables বা PHP language কর্তৃক পূর্ব নির্ধারিত variable

PHP তে কিছু Super Global variable রয়েছে যে গুলোকে বলা হয় predefined variable বা PHP language কর্তৃক পূর্ব নির্ধারিত variable

- \$GLOBALS
- \$_SERVER
- \$_GET
- \$_POST
- \$_FILES
- \$_COOKIE
- \$_SESSION
- \$_REQUEST
- \$_ENV

User Defined Variables বা Programmer বা Developer কর্তৃক নির্ধারিত variable

একজন Programmer বা Developer বিভিন্ন অস্থায়ী তথ্য সংরক্ষণের জন্যে বিভিন্ন সময়ে সমস্ত কোড জোড়ে বিভিন্ন variable declare এবং তার value assign করে থাকেন, এবং সমস্ত কোড জোড়ে তা ব্যবহার করে থাকেন। প্রোগ্রামিং এর ভাষায় এইগুলোকে বলা হয় User Defined Variable. User Defined Variable Declare এর সময় to be careful থাকতে হয়। Because, সে গুলো যেন in any ways Pre defined variable নামের সাথে মিলে না যায়।

User Defined Variable কত প্রকার

1. variable scope বা স্থান ভিত্তিক variable
2. variable variables অর্থাৎ কোনো variable এর value কে variable করা।
3. Reference Variable বা একটা variable এর value তে ভিন্ন ভিন্ন variable নাম দিয়ে access করতে পারা।

What is PHP Constant

PHP constant হচ্ছে Variable এর মতই কোনো একটা সাধারণ স্থায়ী Value বা মানের জন্য Identifier বা নাম, Constant এর সাথে Variable এর মূল পার্থক্য হচ্ছে Constant কে একবার Define করলে আর পরিবর্তন করা যায় না অথবা দ্বিতীয়বার আর declare বা ঘোষণা করা যাবে না। যেখানে Variable এর মান কে সমস্ত কোড জুড়ে যেকোনো জায়গায় পরিবর্তন করা যায়।

PHP তে কিভাবে Constant ঘোষণা করবো?

PHP তে **define()** function অথবা **const** keyword দিয়ে constant ঘোষণা করতে হয়। একটি Valid Constant নাম শুরু হয় লেটার(letter) বা (__) Underscore দ্বারা। Constant নামের পূর্বে ডলার(\$) সাইন দেওয়া যাবে না। PHP তে Constant এর নাম case sensitive. যেমন PHP এর দৃষ্টিতে, a এবং A দুটি আলাদা কনস্ট্যান্ট। Constant নামে কোনো ফাঁকা (স্পেস) থাকা যাবে না। যদি নাম একের অধিক হয় তাহলে “_”underscore (first_name) অথবা বড় হাতের অক্ষরে(firstName) লিখতে হবে।

- `define("CONSTANT", "Hello world.");`
- `const TEST="Hello Bangladesh";`



PHP Operators

PHP অথবা যেকোনো Programming Language এ Operators হচ্ছে কতগুলো symbol যা PHP Interpreter কে কিছু নির্দিষ্ট mathematical, relational এবং logical সহ আরো অনেক ধরনের কার্য (operation) সম্পাদন করার নির্দেশনা পাঠায়। এবং নির্দেশনা অনুযায়ী PHP Interpreter সেই নির্দিষ্ট কাজের ফলাফল প্রদর্শন করে।

Type of PHP Operators

- Arithmetic Operators বা গাণিতিক কাজের Operator: (+,-,*,/,%,**)
- Assignment Operators বা PHP Variable এ যেকোনো মান (Value) সংরক্ষণের Operator : (=)
- Comparison Operators বা দুই বা ততোধিক Value এর মধ্যে তুলনা করার Operator :-(==,!=,===,!==,<,>,<=,>=,<>,<=>)
- Error Control Operators বা Error প্রদর্শনে বাধা দেওয়ার Operator :-(@)
- Execution Operators বা shell কমান্ডের মাধ্যমে PHP Code সম্পাদন করার Operator :- backticks (`)
- Incrementing/Decrementing Operators বা 1 করে বৃদ্ধি বা কমানোর Operator:-(++,- -)
- Logical Operators বা একাধিক অবস্থার বা condition এর উপর ভিত্তি করে সিদ্ধান্ত নেওয়ার Operator:-(&&||,!and,or,xor)
- String Operators বা এক বা একাধিক string এর সাথে জোড়া লাগানোর Operator:-(.)
- Array Operators বা এক বা একাধিক array কে বিভিন্ন রকমের compare, union করার operator:-(+,==,!=,===,!==,<>)

Operator Precedence কি?

PHP অথবা যেকোনো Programming Language এ Operator Precedence হচ্ছে কিছু rules বা নিয়মের সমষ্টি যা দিয়ে প্রোগ্রামিং অথবা যে কোনো গাণিতিক কাজে কোন কাজটি আগে সম্পাদিত হবে তা নির্ধারিত হয়।

```
<?php  
echo 1 + 5 * 3; //Output 16 But why not 18?  
?>
```

উপরের কোডের ফলাফল 18 না এসে 16 আসবে। এর মূল কারণ হচ্ছে, আমাদের এখানে ব্যবহৃত দুটি operator এর মধ্যে Operator Precedence Rules অনুযায়ী addition (“+”) Operator টির চেয়ে multiplication (“*”) Operator টি বেশি অগ্রাধিকার প্রাপ্ত।

Operator Associativity কি?

Operator Associativity হচ্ছে, PHP অথবা যেকোনো Programming Language এ parentheses অনুপস্থিতিতে একই Precedence এর Operator গুলোর মধ্যে কোন Operator টি অগ্রাধিকার প্রাপ্ত হবে তা নির্ধারণ করা।

PHP তে Operator Associativity কত প্রকার? এবং কি কি ?

PHP তে Operator Associativity দুই প্রকার :

১. Right to Left (অর্থাৎ: ডানদিকের Operator অগ্রগণ্য হবে।)
২. Left to Right (অর্থাৎ: বামদিকের Operator অগ্রগণ্য হবে।)

নিম্নে পিএইচপি Operator Precedence এর একটা তালিকা আছে। সবচেয়ে উপরের Operator টির অগ্রগণ্যতা সবচেয়ে বেশি এভাবে এক এক করে নিচে যাবে। এই তালিকাটি মনে রাখা খুব জরুরী। তা না হলে বিভিন্ন Mathematical এবং Programming এর সমস্যা সমাধান করতে গেলে কান্ডিত ফলাফল না ও পেতে পারেন।

Operator Precedence এর উপর ভিত্তি করে Operator গুলোকে ৯ টি ক্যাটাগরি তে করা হয়েছে

Category এর নাম	Operator এর নাম	Associativity
Unary	! ++ —	Right to left
Multiplicative	* / %	Left to right
Additive	+ —	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=	Right to left

PHP String

PHP তে String বলতে বুজানো হয় একাধিক ক্যারেক্টার এর ক্রম বা বিন্যাস (Sequence of Characters)। আরো সহজভাবে বলা যায় PHP তে কোন ম্যাসেজ, তথ্য, এমনকি কখনো কখনো সংখ্যাও String এর মাধ্যমে প্রকাশ করা হয়। PHP তে String নির্দেশ করার জন্য ডাবল কোটেশন (” “) এবং সিঙ্গেল কোটেশন (‘ ‘) ব্যবহার করা হয়। যেমন “PHP string by Double quotation.” অথবা ‘PHP string by Single quotation.’ যদি কখনো String এর মধ্যেই ডাবল কোটেশন (” “) অথবা সিঙ্গেল কোটেশন (‘ ‘) ব্যবহার করার প্রয়োজন হয় “Single ‘quoted text’ is displaying in a string.” এর অনুরূপ সিঙ্গেল কোটেশন এবং ‘Double “quoted text” is displaying in a string.’ অনুরূপ ডাবল কোটেশন ব্যবহার করা যাবে।

PHP তে Double Quoted String এবং Single Quoted String এর মধ্যে কোনো পার্থক্য আছে ?

PHP তে Double Quoted যেকোনো String এ PHP Variable ব্যবহার করলে , সেই Variable এর value Print হয়।

অন্যদিকে PHP তে Single Quoted যেকোনো String এ PHP Variable ব্যবহার করলে , সেই Variable এর value Print হয় না। অর্থাৎ সরাসরি variable এর নামটি ই print হয়। নিচের উদাহরণটি দেখুন :

```
<?php
$name="Sahab Uddin";
echo "Hello $name"; //Output: Sahab Uddin
echo "<br>";
echo 'Hello $name'; //Output: Hello $name
```



Control Structure

PHP অথবা যেকোনো Programming Language এ Control Structure হচ্ছে প্রোগ্রাম বা Code গুলোর নিয়ন্ত্রণের কাঠামো অথবা এমন কিছু statement যা দিয়ে আমরা Program Flow Control করে থাকি বা Program এর পরবর্তী করণীয় কি তা নির্ধারণ করে দেই।

Control Structure List

1. Selection/Branching Statement
 1. Conditional Statement
 1. if
 2. if...else
 3. if...elseif/else if...n...else
 4. switch Statement
 2. Unconditional Statement
 1. goto
 2. Continue
 3. break
 4. return
2. Loops
 1. for
 2. while
 3. do...while
 4. foreach
3. File Inclusion
 1. include
 2. require
 3. include_once
 4. require_once

switch	VS	else if/elseif
switch ক্রমান্বয়ে সবগুলো condition বা শর্ত চেক করার পরিবর্তে যেই কন্ডিশন মিলে তা লাফ দিয়ে execute করে। অন্য কোনো condition চেক করেনা।		আর else if/elseif statement ক্রমান্বয়ে সবগুলো condition বা শর্ত চেক করতে থাকে এবং প্রথম যেই শর্ত পূরণ হবে , তার প্রোগ্রাম execute করার সুযোগ দেয়। এবং পরবর্তী condition গুলো আর চেক করেনা।
switch সবসময় (equality) সমান কিনা তা দেখে		elseif/else if সবসময় boolean true কিনা তা দেখে
elseif এর তুলনায় switch faster		switch এর তুলনায় elseif slower
অনেক গুলো চয়েজ এর ক্ষেত্রে switch statement তুলনামূলক অনেক বেশি compact বা সহজবোদ্ধ।		অনেক গুলো চয়েজ এর ক্ষেত্রে elseif তুলনামূলক কঠিন।
switch case এ break statement ব্যবহার করা অপরিহার্য		else if/elseif statement এ break statement ব্যবহার করা লাগেনা।
switch এর expression এর মধ্যে ব্যবহৃত variable এর value বা মান শুধু integer data type সাপোর্ট করে।		elseif/else if এর expression এর মধ্যে ব্যবহৃত variable এর value বা মান integer এবং character উভয় data type সাপোর্ট করে।
switch statement একাধিক value এর range চেক করতে পারেনা		elseif/else if statement এ একাধিক value এর range চেক করতে পারে
switch statement এ মাল্টিপল চয়েজ এর ক্ষেত্রে শুধু মাত্র single expression ব্যবহার করতে পারবেন।		elseif/else if statement এ প্রত্যেকটি চয়েজ এর ক্ষেত্রে আলাদা expression ব্যবহার করতে পারবেন।

PHP তে foreach

PHP তে foreach বিশেষ ভাবে PHP array এবং object এর প্রত্যেকটি element এ loop এর কাজ করার জন্য ব্যবহৃত হয় ।

PHP foreach Syntax

syntax 1:

```
foreach ($array_or_object_handler as $value){  
    statement  
}
```

syntax 2:

```
foreach ($array_or_object_handler as $key=>$value){  
    statement  
}
```

syntax 3:

```
foreach ($array_or_object_handler as $value):  
    statement  
endforeach
```

PHP তে array কি

PHP তে array হচ্ছে ভিন্ন ভিন্ন Type এর data বা তথ্যের সমষ্টি, তথ্যের তালিকা অথবা মানচিত্র, আরেকটু সহজ ভাবে বলা যায়, array হচ্ছে একটা compound variable যা একই সময়ে একাধিক type এর value ধারণ করতে পারে, যেখানে একটি index এর বিপরীতে আমরা একটি value সংরক্ষণ করি। Index এর বিপরীতে value define করার জন্য আমরা => সিদ্ধল ব্যবহার করি। সাধারণত PHP array তে => চিহ্নের বামের অংশটুকুকে array key, array index অথবা array property বলে, এবং ডানের অংশটুকুকে array value বলে।

PHP তে কিভাবে array ঘোষণা করা হয়

১. array function এর মাধ্যমে
২. array short syntax এর মাধ্যমে

PHP array

print_r() function

PHP তে array এবং object এর property name এবং Property value প্রিন্ট করার জন্য print_r() function টি ব্যবহৃত হয়।

var_dump() function

PHP তে array এবং object এর টোটাল সাইজ, Property value size, Property name, Property value type এবং Property value প্রিন্ট করার জন্য var_dump() function টি ব্যবহৃত হয়।

PHP তে array কত প্রকার

PHP তে Dimension এর ভিত্তিতে array দুই প্রকার

১. Single Array

PHP তে যখন একটি array এর মধ্যে অন্য কোনো array না থাকে PHP এর পরিভাষায় একে single array বলে।

২. Nested Array

PHP তে যখন একটি array এর মধ্যে এক বা একাধিক array থাকে PHP এর পরিভাষায় একে Nested array বলে।

PHP তে index বা key এর ভিত্তিতে array তিন প্রকার

1. Numerical

PHP তে যেইসব array এর index শুধু মাত্র number সেগুলোকে Numerical / Enumerated Array. বলে।

2. Associative Array

PHP তে যেইসব array এর index শুধু মাত্র string সেগুলোকে Associative Array. বলে।

3. Hybrid Array

PHP তে যেইসব array এর index গুলো string এবং number মিলিত অবস্থায় থাকে সেগুলোকে Hybrid Array বলে।

Array General Functions

1. `foreach()`
2. `list()`
3. `each()`
4. `Extract()`
5. `count()`
6. `reset()`
7. `end()`
8. `current()`
9. `key()`
10. `next()`
11. `prev()`
12. `range()`
13. `array_unique`
14. `in_array()`
15. `array_key_exists()`
16. `array_rand ()`
17. `array_flip()`
18. `array_reverse()`
19. `array_keys()`
20. `array_values()`
21. `array_count_values()`

Array General Functions

PHP foreach() function এর মাধ্যমে আমরা খুব সহজেই array এর key এবং value গুলোকে print করতে পারি.

while loop এ list() function এবং each() function ব্যবহার

PHP list ফাংশনের কাজ হচ্ছে array এর প্রত্যেকটি value কে variable এ assign করা।

PHP each ফাংশনের কাজ হচ্ছে array এর current key এবং value রিটার্ন করা।

PHP count(), reset(), end(), current(), key(), next() এবং prev()

count()টির কাজ হচ্ছে array total element কতটি তা count করা, reset() টির কাজ হচ্ছে array টিকে reset করা অর্থাৎ array pointer কে শুরুতে নিয়ে যাওয়া। current() টির কাজ হচ্ছে array এর current value রাখা। key() টির কাজ হচ্ছে array এর current key রাখা। next() টির কাজ হচ্ছে array এর pointer কে একধাপ পরের ভ্যালু কে value তে point করা এবং prev() টির কাজ হচ্ছে array এর pointer কে একধাপ আগের value তে point করা

Array General Functions

PHP array array_count_values() function

PHP array এর মধ্যে কোন value টি কতবার আছে তা জানার জন্য আপনি array_count_values() function টি ব্যবহার করতে পারেন।

PHP array_keys() এবং array_values() function

php তে কোনো array এর শুধু column name গুলোকে নিয়ে আলাদা array করতে চাইলে আপনাকে array_keys এবং শুধু value গুলোকে নিয়ে আলাদা array করতে চাইলে আপনাকে array_values function টি ব্যবহার করতে হবে।

PHP array_flip() এবং array_reverse() function

php তে কোনো array এর key গুলোকে value তে রূপান্তর অথবা value গুলোকে key তে রূপান্তর করতে চাইলে আপনাকে array_flip() function এবং array এর সব গুলো value কে বিপরীত দিক থেকে সাজাতে চাইলে আপনাকে array_reverse() function টি ব্যবহার করতে হবে।

তবে array তে একই value যদি একাধিক বার থাকে, সেক্ষেত্রে শুধু শেষ value টি key হিসেবে থাকবে, অন্য গুলো replace হয়ে যাবে।

Array General Functions

PHP array_rand () এবং shuffle() function

php তে কোনো array এর value গুলো থেকে এক বা একাধিক value কে (এলো-মেলো করে) randomly পিক (বের) করতে চাইলে আপনাকে array_rand() function এবং array এর সব গুলো value কে এলো-মেলো (shuffling) করতে চাইলে আপনাকে shuffle() function টি ব্যবহার করতে হবে।

PHP array_key_exists() এবং in_array() function

php তে array এর মধ্যে কোনো নির্দিষ্ট value আছে কিনা তা চেক করতে চাইলে আপনাকে in_array() function এবং array এর মধ্যে কোনো key আছে কিনা তা চেক করতে চাইলে আপনাকে array_key_exists() function টি ব্যবহার করতে হবে।

PHP range() এবং array_unique function

PHP তে Number অথবা letter দিয়ে নির্দিষ্ট range এর array তৈরী করার জন্য আপনাকে range() function এবং একটি array এর মধ্যে অবস্থিত value গুলোর মধ্যে unique value নিয়ে নতুন array তৈরী করার জন্য array_unique() function ব্যবহার করতে হবে।

PHP functions



PHP অথবা যেকোনো প্রোগ্রামিং Language এ function হচ্ছে Program এর এমন একটা code block যা একটা নির্দিষ্ট কার্য সম্পাদনের এবং তা বার বার ব্যবহারের জন্যে ব্যবহৃত হয়। অর্থাৎ function এর মধ্যে এমন কিছু Instruction বা নির্দেশাবলী থাকে যা function ব্যবহারকারী থেকে এক বা একাধিক input নিয়ে এক বা একাধিক output বা result রিটার্ন করার জন্য ব্যবহৃত হয়। সাধারণত, program page লোড হওয়ার সাথে সাথেই ডিক্লেয়ার করা function execute হয়না, যতক্ষণ না function কে call করা না হয়। উদাহরণ হিসেবে উপরের juice machine টি হচ্ছে function আর কমলা গুলো হচ্ছে function এর input এবং function এর output বা রিটার্ন হচ্ছে কমলার শরবত।

PHP function

PHP তে function কত প্রকার

1. PHP Pre-built function বা PHP language কর্তৃক পূর্ব নির্ধারিত বিভিন্ন library function বা library ছাড়া function
2. User/Programmer defined function বা Programmer কর্তৃক নির্ধারিত function

PHP তে function কিভাবে লিখতে হয়?

1. PHP তে function define করার সময় letter অথবা underscore দিয়ে শুরু করতে হয়।
2. PHP তে function Case Insensitive অর্থাৎ একই নামে কোনো ফাংশন lowercase এবং uppercase এর কারণে আলাদা function হিসেবে বিবেচিত হবে না।

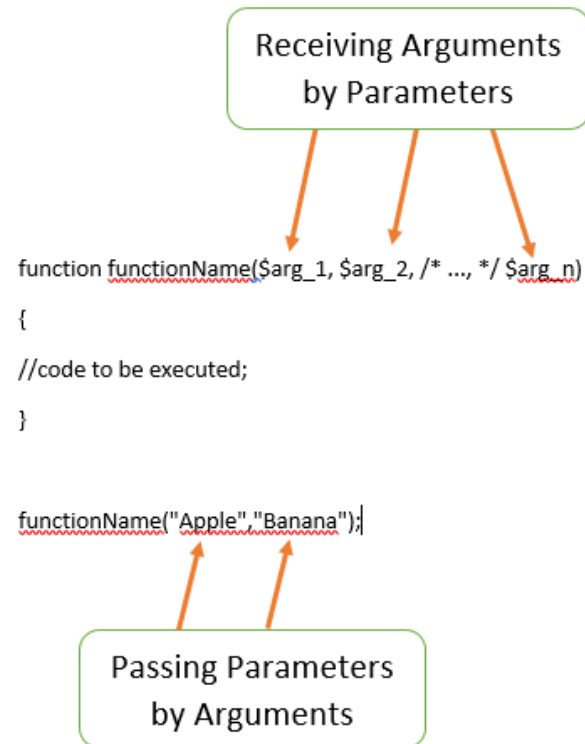
PHP তে function লেখার Syntax কি

```
<?php
function functionName($arg_1, $arg_2, /* ..., */ $arg_n)
{
    //code to be executed;
}
?>
```

PHP function

PHP তে function এর মধ্যে Parameter কি এবং argument কি ?

function declaration অথবা define এর সময় parentheses এর মধ্যে ব্যবহৃত একেকটি variable কে একেকটি function parameter বলা হয়। একই ভাবে function calling এর সময় parentheses এর মধ্যে ব্যবহৃত একেকটি variable কে একেকটি function argument বলা হয়।



PHP function

PHP তে functions এর arguments গুলোকে কয় ভাবে receive বা গ্রহণ করা যায়?

1. সরাসরি functions এর Parameter দিয়ে argument গুলো রিসিভ করা যায়।
2. functions এর Parameter ছাড়া argument গুলো রিসিভ করা যায়।

PHP তে Function এর মধ্যে function

যখন একটি function কে আরেকটি function এর মধ্যে declare করা হয়, PHP অথবা যেকোনো প্রোগ্রামিং এর পরিভাষায় একে বলা হয় functions within function বা function এর মধ্যে function. আর এর জন্য আপনাকে যা কাজ করতে হবে :

1. প্রথমে Parent function অর্থাৎ যেই function এর মধ্যে function টি declare করা হয়েছে, সেই function টি কল করতে হবে।
2. তারপর child function বা প্যারেন্ট function এর মধ্যের function টি কল করতে পারবেন।

PHP তে Conditional Function কি ?

যখন কোনো function, কোনো একটা condition true হওয়ার পর declare বা create হয়। PHP অথবা যেকোনো প্রোগ্রামিং এর পরিভাষায় একে বলা হয় conditional function.

PHP তে Recursive Function কি ?

function যখন নিজেই নিজেকে কল করে বা যখন function নিজের মধ্যেই নিজেকে কল করে , PHP অথবা যেকোনো Programming Language এর পরিভাষায় একে বলা হয় Recursive Function বা Recursion.

PHP functions Type hinting

PHP তে function call করা কালীন যেসব Parameter দিয়ে যেসব Argument গ্রহণ করব তার Data type এবং function থেকে যেসব value return করবো তার data type কি হবে, তা নির্ধারণ করার পদ্ধতিকে PHP তে Type Hinting বলে।

PHP তে Type Hinting কত প্রকার এবং কি কি ?

PHP তে Type Hinting দুই প্রকার :

- Argument Type Hinting
- Return Type Hinting

PHP তে Argument Type Hinting কি ?

PHP তে function call করা কালীন Parameter এর মাধ্যমে যেসব Argument গ্রহণ করব তার data type কি হবে, তা নির্ধারণ করার পদ্ধতিকে PHP তে Argument Type Hinting বলে।

PHP functions Type hinting

PHP তে Type Hinting এর জন্য কি কি data type সাপোর্ট করে?

PHP 7.1 পর্যন্ত ৯ ধরনের data type সাপোর্ট করে :

- Class/interface
- self
- array
- callable
- bool
- float
- int
- string
- iterable

PHP functions Type hinting

PHP তে Argument Type Hinting কিভাবে কাজ করে?

PHP তে Argument Type Hinting কিভাবে কাজ করে তা বুঝার জন্য নিচের উদাহরণটি দেখে নেয়া যাক :

```
function TypeHint(int $a){  
    var_dump($a);  
}  
TypeHint(5.5); //Output: (int)5
```

ব্যাখ্যা : উপরের typeHint function এ আমরা \$a parameter এর data type হিসেবে integer ঘোষণা করেছি , যার জন্য \$a paramter টি আমাদের argument value 5.5 কে integer এ convert করে শুধু 5 return করবে। আর যেহেতু PHP একটা loosely type language তাই কোনো রকম error দেখাবেনা। তবে আমরা চাইলে declare function দিয়ে PHP কে run time strict type করে নিতে পারি। আর তখন PHP একটা fatal error দেখাবে।

```
declare(strict_types=1);  
function TypeHint(int $a){  
    var_dump($a);  
}  
TypeHint(5.5);
```

PHP functions Type hinting

array দিয়ে type hinting এর একটা উদাহরণ দেখা যাক

```
declare(strict_types=1);
function TypeHint(array $a){
    return $a;
}
try {
    var_dump(typeHint([5,6,5.5]));
    echo "<br>";
    var_dump(typeHint(5));
} catch (TypeError $e) {
    echo 'Error: '.$e->getMessage();
}
```

ব্যাখ্যা :try block এর মধ্যে প্রথম typeHint function টি কাজ করবে, কারণ সেখানে আমরা function এর Parameter কে type hint হিসেবে array ঘোষণা করেছি , আর এর জন্য argument এর value ও array পাঠিয়েছি। আর try block এর মধ্যে দ্বিতীয় typeHint function টি কাজ করবেনা। কারণ সেখানে argument এর value পাঠিয়েছি interger.

PHP functions Type hinting

Return Type Hinting কি ?

PHP তে function call করা কালীন function থেকে যেসব value return করবে, তার data type কি হবে তা নির্ধারণ করার পদ্ধতিকে PHP তে Return Type Hinting বলে।

```
function sum($a, $b): int {  
    return $a + $b;  
}
```

// Note that a float will be returned.

```
var_dump(sum(3, 2.5)); //Result: 5
```

ব্যাখ্যা : উপরের sum function এ parenthesis এর পর return data type হিসেবে integer ঘোষণা করেছি , যার জন্য আমাদের return value 5.5 কে integer এ convert করে শুধু 5 return করবে। আর যেহেতু PHP একটা loosely type language তাই কোনো রকম error দেখাবেনা। তবে আমরা চাইলে declare function দিয়ে PHP কে run time strict type করে নিতে পারি। আর তখন PHP একটা fatal error দেখাবে।

```
declare(strict_types=1);
```


PHP functions Echo VS Return

পার্থক্য ১:

ধরুন আপনি দুটি function এর result নিয়ে summation বা যোগফল বের করবেন , সেক্ষেত্রে function দুটিতে return এর পরিবর্তে echo ব্যবহার করলে কি সমস্যা হতে পারে তা নিচের উদাহরণ দেখলেই বুঝতে পারবেন।

Result: 5 3 0

পাঠক লক্ষ্য করুন , আমরা আশা করেছিলাম add1 এবং add2 function দুটির রেজাল্ট দিয়ে তৃতীয় একটা যোগফল বের করব, কিন্তু function দুটিতে return এর পরিবর্তে echo ব্যবহার করায়, আমাদের কে 5 এবং 3 আলাদা প্রিন্ট করে দিয়েছে , আর আউটপুট আসছে শুধু 0, এবার আমরা echo এর পরিবর্তে return ব্যবহার করব। আর তাতেই আমাদের আশা অনুযায়ী add1 এবং add2 function দুটির রেজাল্ট দিয়ে তৃতীয় একটা যোগফল বের করব :

Result: 8

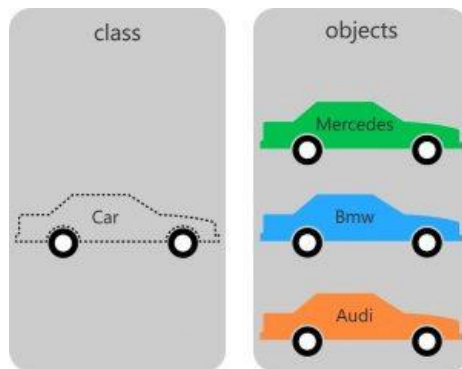
পার্থক্য ২:

function এর মধ্যে echo একাধিক বার ব্যবহার করা যায়, কিন্তু return শুধু মাত্র একবার ব্যবহার করা যায় অর্থাৎ function এর মধ্যে return এর পর আর কোনো code execute করেনা।

পার্থক্য ৩:

function এর মধ্যে echo ব্যবহার করলে function calling এর সময় আবার echo দিতে হয়না। কিন্তু function এর মধ্যে return ব্যবহার করলে এবং function থেকে output দেখাতে চাইলে , function calling এর সময় echo দিতে হয়।

PHP Object Oriented Programming



Object কি?

আমরা আমাদের চার পাশে যা কিছু দেখি তার সবই Object বা বস্তু। এই ধরুন আপনি আপনার নিজেকে এবং আপনার আসে-পাশের দিকে খেয়াল করুন , আপনার গায়ের জামা, আপনার পরনের প্যান্ট, হাতের মোবাইল, পড়ার টেবিল,বসার চেয়ার ,কম্পিউটার এই সবই একেকটি Object বা বস্তু। এমনকি আপনি / আমি নিজেও একটা Object.

class কি?

Object সম্পর্কে তো জানা হলো , চলুন এবার জানি class কি? class হচ্ছে Object এর replica (অবিকল প্রতিরূপ) বা blueprint (নীলনকশা)। ধরুন আমরা একটি চেয়ার বানাতে চাই, শুরুতে আমরা কোন রকম চিন্তা ভাবনা না করে ফু দিয়ে একটা চেয়ার বানিয়ে ফেলতে পারি না। আমরা এর জন্যে পরিকল্পনা করি- চেয়ারটা দেখতে কেমন হবে, এটি দৈর্ঘ্য, প্রস্থ এবং উচ্চতা কতটুকু হবে, চেয়ারটি কি কি কাজ করবে ইত্যাদি। এই পরিকল্পনা গুলো আমরা আমরা কোথাও লিখে রাখি। আমাদের এই লেখা ডকুমেন্টটি আসলে class. আরেকটু ব্যাখ্যা করে বলতে গেলে আমরা বলতে পারি যে একটা অবজেক্ট কেমন হবে তা যেখানে নির্ধারণ করা থাকে তার নাম class। এই ক্লাস দিয়ে তৈরি অবজেক্টকে ঐ ক্লাসের অস্তিত্ব(Instance) বলে।

PHP OOP Basics

PHP OOP Basics: Object Oriented Programming কি?

এখন প্রশ্ন হল, তাহলে PHP (OOP) Object Oriented Programming কি? Computer Programming এ class এবং Object এর ধারণাকে কাজে লাগিয়ে যে Programming করা হয় তাকেই বলা হয় PHP (OOP) Object Oriented Programming। অবজেক্ট ওরিয়েন্ট কনসেপ্ট তিনটি ধারণার উপর প্রতিষ্ঠিত:

- **Inheritance**-ইংরেজি শব্দ inheritance অর্থ হচ্ছে উত্তরাধিকার। Object Oriented Programming এ এক class থেকে কিছু ফাংশনালিটি (Methods) ও বৈশিষ্ট্য (Property) অন্য class এ শেয়ার করার একটা পদ্ধতি হচ্ছে Inheritance. বা কোন Class বা Object এ অন্য কোন Class বা Object এর বৈশিষ্ট্য অর্জন করে থাকে যে প্রক্রিয়ায় তাকে উত্তরাধিকার সূত্র বা Inheritance বলে। আরো সহজ ভাবে বলা যায়, একটা ক্লাসকে (Parent Class) Inherit (অনুসরণ) করে তার কিছু বৈশিষ্ট্য আরেকটি উত্তরসূরি (child class) এর মধ্যে ব্যবহার করার প্রক্রিয়াকে Inheritance বলা হয়।
- **encapsulation**-অর্থাৎ সব কিছু encapsulate অবস্থায় থাকা। অনেকটা ক্যাপসুল মেডিসিন এর মতোই, যেভাবে একটা capsule এর বাইরে একটা আবরণ দিয়ে সব কিছু ভেতরে আটকানো থাকে। একই ভাবে Object Oriented Programming এ একটা class এর আবরণ দিয়ে সব কিছু ভেতরে আটকানো থাকে।
- **polymorphism** – বহুরূপিতা। অর্থাৎ একটা অবজেক্ট নানা সময় নানা রকম রূপ ধারণ করতে পারে বা বহুরূপে ব্যবহৃত হতে পারে।

PHP OOP Basics

- **Parent Class:** একটি class থেকে যখন অন্য একটা class inherits বা উত্তরাধিকার সূত্রে তৈরী হয় , তখন যেই class থেকে নতুন class টি তৈরী হয়, তাকে Parent class বলা হয়। Parent class কে base class বা super class ও বলে।
- **Child Class:** একটি class যখন অন্য একটা class থেকে inherits বা উত্তরাধিকার সূত্রে তৈরী হয় , তখন তাকে child class বলা হয়। child class কে subclass বা derived class ও বলে।
- **Visibility/Access modifier:**
ইংরেজি শব্দ visibility অর্থ দৃশ্যমানতা, অর্থাৎ class এর মধ্যে যেকোনো property, constant অথবা Method এ কে access পাবে তা visibility দিয়ে নির্ধারিত হয়। Visibility গুলো সাধারণত class এর মধ্যে constant, property, method এর আগে ঘোষণা করতে হয়।

PHP OOP Basics

PHP তে Visibility তিন ধরনের:

- **private** :Class এর মধ্যে constant, property, method গুলোকে যদি শুধু একই class বা class নিজের মধ্যে ব্যবহার সীমিত রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে private ঘোষণা করতে হয়।
- **protected**: Class এর মধ্যে constant, property, method গুলোকে যদি শুধু একই class বা class নিজের মধ্যে এবং তার child class এর মধ্যে ব্যবহার সীমিত রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে protected ঘোষণা করতে হয়।
- **public**: Class এর মধ্যে constant, property, method গুলোকে যদি একই class বা class নিজের মধ্যে , child class এবং class এর বাহির থেকে ব্যবহার রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে public ঘোষণা করতে হয়।

PHP OOP Basics

- **Property:** PHP OOP তে class এর মধ্যে অবস্থিত variable গুলিকে Property বলা হয়। Property গুলোকে “attributes” বা “fields” নামেও ডাকা হয়। PHP তে Property গুলোকে যেকোনো একটা visibility অর্থাৎ : public, private, protected এর যেকোনো একটি visibility সহ ডিফাইন করতে হয়।
- **Method:** PHP তে class এর মধ্যে অবস্থিত function গুলিকে Method বলা হয়। PHP তে Method গুলোকে যেকোনো একটা visibility অর্থাৎ : public, private, protected এর যেকোনো একটি visibility সহ declare বা ঘোষণা করতে হয়।
- **static keyword:** class এর যেকোনো property এবং method এ কোন রকম instance বা object ছাড়া access করার সুযোগ দিতে চাইলে বা সরাসরি class দিয়ে access করার সুযোগ দিতে চাইলে static keyword ব্যবহৃত হয়।
- **“::” scope resolution operator** PHP তে class এর মধ্যে অবস্থিত যেকোনো static property, static method এবং constant কে class এর ভিতরে অথবা বাহিরে যেকোনো কাজে ব্যবহার করতে চাইলে “:: scope resolution operator” ব্যবহার করতে হয়।
- **“\$this” pseudo-variable:** “\$this” pseudo-variable টি মূলত current class এর object কে ধারণ করে। PHP তে class এর মধ্যে অবস্থিত যেকোনো Non-static property এবং Non-static method কে class এর মধ্যেই যেকোনো কাজে ব্যবহার করতে চাইলে “\$this” pseudo-variable ব্যবহার করতে হয়।

PHP তে class declare বা ঘোষণা করার নিয়ম

PHP তে “class” keyword এর পর letter অথবা underscore দিয়ে class name শুরু করতে হয়। class name এর পর curly braces দিয়ে class শুরু করতে হয়, এবং curly braces দিয়ে class শেষ করতে। curly braces এর ভিতরের অংশকে বলা হয় class body . class name হিসাবে PHP এর reserved word দেওয়া যাবেনা।

```
<?php
class SimpleClass // class Declaration
{ //class start

    /* .....
    Class Body
    ..... */

} //class end
?>
```

PHP তে class থেকে object তৈরির নিয়ম কি?

PHP তে class থেকে অবজেক্ট তৈরী করতে হলে class এর বাহিরে class এর নামের পূর্বে new keyword দিতে হয়। অর্থাৎ new className অথবা new className(). আর object কে যেই variable এ রাখা হয় তাকে object handler বলা হয়।

```
<?php  
class SimpleClass  
{ //class start
```

```
    /* .....  
    Class Body  
    .....*/
```

```
} //class end
```

```
$objHandler=new SimpleClass(); // Object Call  
?>
```

PHP তে class এর মধ্যে constant, property এবং method ঘোষণা করার নিয়ম কি?

PHP তে class এর মধ্যে constant, property এবং method ঘোষণা করতে হলে constant, property এবং method গুলোর নামের পূর্বে আপনার প্রয়োজন অনুসারে যেকোনো একটা visibility দিতে হয়। তবে constant এর বেলায় visibility ঐচ্ছিক বা optional.constant এ কোন visibility ঘোষণা না করলে বাই ডিফল্ট public থাকে।

```
<?php
class SimpleClass
{
    public $var = 'a default value'; // public property declaration
    private const a="America"; // private constant Declaration

    // public method declaration
    public function displayVar() {
        echo $this->var;
    }
}
?>
```


PHP তে class এর মধ্যে অবস্থিত property, method এবং constant ব্যবহারের নিয়ম কি ?

PHP তে class এর মধ্যে অবস্থিত property, method এবং constant গুলো দুইভাবে ব্যবহৃত হতে পারে।

১. class এর বাহিরে।
২. class এর মধ্যে।

property, method এবং constant কে class এর বাহিরে ব্যবহার

object handler বা object variable দিয়ে : PHP তে class এর মধ্যে অবস্থিত যেকোনো public Non-static property এবং public Non-static method কে class এর বাহিরে যেকোনো কাজে ব্যবহার করতে চাইলে সরাসরি object handler বা object variable দিয়ে ব্যবহার করা যায়।



```
<?php
class SimpleClass
{
    // property declaration
    public $var = 'Hello Simple Class';

    // method declaration
    public function displayVar() {
        echo $this->var;
    }
}
$obj=new SimpleClass;
$obj->displayVar();
/*
access class method using Object Variable $obj
Result: Hello Simple Class
*/
?>
```


“::” scope resolution operator PHP তে class এর মধ্যে অবস্থিত যেকোনো public [static](#) property, public [static](#) method এবং public constant কে class এর বাহিরে যেকোনো কাজে ব্যবহার করতে চাইলে **“:: scope resolution operator”** ব্যবহার করতে হয়।

```
<?php
class SimpleClass
{
    // property declaration
    public $var = 'Hello Simple Class';
    public const a="America";

    // Static method declaration
    public static function sayHello() {
        echo "Hello World";
    }
}
SimpleClass::sayHello();
echo SimpleClass::a;
/*
access class method using directly class name and scope resolution Operator ::
Result: Hello World
*/
?>
```

property, method এবং constant কে class এর মধ্যেই ব্যবহার

“\$this” pseudo-variable ব্যবহার করে।

```
<?php
class SimpleClass
{
    // property declaration
    public $var = 'Hello Simple Class';

    // method declaration
    public function sayHello() {
        echo $this->var; // use $this to show $var value
    }
}
$obj=new SimpleClass;
$obj->sayHello(); // Result: Hello Simple Class
?>
```

“::” scope resolution operator PHP তে class এর মধ্যে অবস্থিত যেকোনো [static](#) property, [static](#) method এবং constant কে class এর মধ্যেই যেকোনো কাজে ব্যবহার করতে চাইলে **“:: scope resolution operator”** ব্যবহার করতে হয়।

```
<?php
class SimpleClass
{
    // property declaration
    static public $var = 'Hello Simple Class';
    const a="America";

    // method declaration
    public function sayHello() {
        echo self::$var; // use self with scope resolution Operator
        echo "<br>", self::a;
    }
}
$obj=new SimpleClass;
$obj->sayHello();

?>
```

PHP OOP Method Chaining

PHP তে Method Chaining কি?

PHP তে যখন অনেক গুলো method কে একটি single instruction এর মধ্যে call করা হয়, PHP এর পরিভাষায় একে বলা হয় Method Chaining . আরেকটু সহজ ভাবে বুঝিয়ে বলি, ধরুন আপনার Person নামে একটা class আছে, যেখানে আপনার **\$name** এবং **\$age** নামে দুটি **private property** আছে। আর **\$name** এবং **\$age** property দুটিতে value set করার জন্য আছে যথাক্রমে **setName()** এবং **setAge()** নামক দুটি method. এবং সর্বশেষে **\$name** এবং **\$age** property দুটির value দেখানোর জন্য আছে **getInfo()** নামক method. এবং ধরুন আমাদের object টির নাম হচ্ছে **\$person**. এখন আমরা যদি এই তিনটি method কে আলাদা আলাদা ভাবে object দিয়ে call না করে একটি single instruction এর মধ্যে ঠিক এইভাবে call করতে চাই : **\$person->setName("Sahab Uddin")->setAge(24)->getInfo()**. PHP এর পরিভাষায় এই পদ্ধতিকে বলা হয় Method Chaining . Method Chaining এ class এর মধ্যের প্রত্যেকটি Method অর্থাৎ যেই class এর method সেই class এর object কে return করে। Method Chaining এর জন্য class এর মধ্যে যেকোনো value return এর পরিবর্তে **return \$this;** কথাটি লিখতে হয়।

```
<?php
class person
{
    private $name="";
    private $age="";

    public function setName($name="")
    {
        $this->name=$name;
        return $this;
    }
    public function setAge($age="20")
    {
        $this->age=$age;
        return $this;
    }

    public function getInfo()
    {
        echo "Hello, My name is ".$this->name." and my age is ".$this->age." years.";
    }
}

$person = new person();
$person->setName("Masud Alam")->setAge("33")->getInfo();
?>
```


ব্যাখ্যা : লক্ষ্য করুন আমরা setName() Method এবং setAge() Method দুটির মধ্যে \$this pseudo-variable টি রিটার্ন করি। আর আমরা জানি “\$this” pseudo-variable টি মূলত current class এর object কে ধারণ করে। প্রথমে আসা যাক \$person->setName(‘Sahab Uddin’) এ . এটির কাজ হচ্ছে setName() Method কে call করবে এবং Name Property তে Masud Alam নামকে value হিসেবে সেট করবে। তারপর setName() মেথড \$this রিটার্ন করবে , অর্থাৎ এটি Object person কে রিটার্ন করবে।

এরপরে আশা যাক ->setAge(‘24’) তে , যেহেতু আমরা আগের setName() method থেকে chain object পেয়েছি , সুতরাং সেই person object দিয়েই আমরা setAge(24) কে call করা এবং age Property তে 24 age কে value হিসেবে সেট করবে। আবার এখানেও আমরা \$this রিটার্ন করায়, একই ভাবে এখানেও মূলত আমাদের Object person কে chain রিটার্ন করবে।

সর্বশেষে ->getInfo() method দিয়ে আমরা পূর্বের ধারাবাহিকতায় chain রেজাল্ট পাই।



Static Method Chaining

Static Method Chaining সাধারণ Method Chaining এর মতোই। তবে এখানে প্রত্যেকটি static Method এর মধ্যে `return $this` এর পরিবর্তে `return new static;` কথাটা লিখতে হবে।

```
<?php
class oop{
    public static $val;

    public static function init($var){
        static::$val=$var;
        return new static;
    }

    public static function add($var){
        static::$val+=$var;
        return new static;
    }


    public static function sub($var){
        static::$val-=$var;
        return new static;
    }

    public static function out(){
        return static::$val;
    }
}

echo oop::init(10)->add(2)->sub(5)->out(); //Result: 10+2-5=7
echo "<br>";
echo oop::sub(2)->out(); //Result: 7-2=5;
```

PHP তে Constructor Method কি?

Class থেকে নতুন কোনো object তৈরী হওয়ার সময় যে মেথড স্বয়ংক্রিয় ভাবে এবং সবার আগে execute হয়, PHP তে তাকে constructor Method বলে। PHP তে Constructor Method একটি PHP Language কর্তৃক Pre-defined Public Method অর্থাৎ PHP language কর্তৃক পূর্ব নির্ধারিত Public Method , যা public function এর পর একসাথে দুইবার underscore (__) দেওয়ার পর construct কথাটি লিখতে হয় অর্থাৎ public function __construct() নাম দিয়ে লিখতে হয়।



```
<?php
class oop{
    public function __construct(){
        echo "Welcome to PHP World!";
    }
    public function sayHello(){
        echo "Hello World";
    }
}

$obj=new oop; /* Now Automatically Execute __construct Method
               And Will show "Welcome to PHP World " First.*/

echo "<br>";

$obj->sayHello(); // Result: Hello World


?>
```


যদি কোনো class এর Constructor Method এ Parameter দেওয়া থাকে, তাহলে সেই Constructor Method এ Argument পাঠাতে হলে class object এর সাথে parenthesis “()” দিয়ে আর্গুমেন্ট পাঠাতে হবে।

```
class Car {  
    private $model;  
  
    //__construct  
    public function __construct ($model)  
    {  
        $this->model = $model;  
    }  
  
    public function getCarModel()  
    {  
        return ' The car model is: ' . $this->model;  
    }  
}  
  
//We pass the value of the variable once we create the object  
$car1 = new Car("Mercedes"); // Passing Argument to Constructor Parameter  
  
echo $car1 -> getCarModel();
```

PHP তে Destructor Method কি?

Object এর কাজ শেষ হওয়ার সময় class এর মধ্যে যে মেথড স্বয়ংক্রিয় ভাবে execute হয় বা সবার শেষে execute হয়, PHP তে তাকে Destructor Method বলে। PHP তে Destructor Method একটি PHP Language কর্তৃক Pre-defined Public Method অর্থাৎ PHP language কর্তৃক পূর্ব নির্ধারিত Public Method , যা public function এর পর একসাথে দুইবার underscore (__) দেওয়ার পর destruct কথাটি লিখতে হয় অর্থাৎ public function __destruct() নাম দিয়ে লিখতে হয়।



```
<?php
class oop{
    public function __construct(){
        echo "Welcome to PHP World!";
    }
    public function sayHello(){
        echo "Hello World";
    }

    public function __destruct(){
        echo "<br>I'm about to disappear - bye bye!";
    }
}
$obj=new oop;

echo "<br>";

$obj->sayHello();


?>
```

PHP তে Destructor Method কোনো argument receive করতে পারে ?

না পারেনা এবং সেটার দরকার ও নাই। তবে অন্য যেকোনো method দিয়ে Argument Receive করে যেকোনো property তে রেখে সেই property কে constructor Method দিয়ে রিটার্ন করতে পারবেন।

PHP তে Destructor Method কেন ব্যবহার করবো?

কোনো কিছু fininshing বা শেষ করতে চাইলে destructor Method ব্যবহার করা হয়, যেমন : file closing, Database Connection closing এবং Session Destroy এর মতো কাজগুলো Destructor Method দিয়ে করা হয়।



```
<?php
class oop{
private $name;
public function __construct($val){
    return $this->name=$val;
}
public function sayHello(){
    echo "Hello World";
}

public function __destruct(){
    echo "<br>Bye bye $this->name!";
}

}
$obj=new oop("Imran");

echo "<br>";

$obj->sayHello();

?>
```


Visibility or Access Modifier in PHP

PHP তে Visibility বা Access modifier কি ?

ইংরেজি শব্দ Visibility অর্থ দৃশ্যমানতা, অর্থাৎ class এর মধ্যে যেকোনো property, constant অথবা Method এ কে access পাবে তা visibility দিয়ে নির্ধারিত হয়। Visibility গুলো সাধারণত class এর মধ্যে constant, property, method এর আগে ঘোষণা করতে হয়।

PHP তে Visibility বা Access modifier কয়টি এবং কি কি ?

- Public
- Protected
- Private

Public

Class এর মধ্যে constant, property, method গুলোকে যদি একই class বা class নিজের মধ্যে , child class এবং class এর বাহির থেকে ব্যবহার রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে public ঘোষণা করতে হয়।

আমাদের GrandFather class এর \$name নামক Property টি public হওয়ায়, \$name property টি আমরা একই সাথে current class, child class এবং class এর বাহির থেকেও access করতে পেরেছি।

Visibility or Access Modifier in PHP

```
<?php
class GrandFather
{
    public $name='sahabuddin'; // A public variable
    public function getName(){
        return $this->name;
    }
}

class Daddy extends GrandFather // Inherited class
{
    function displayGrandFaName()
    {
        return $this->name; // The public variable will be available to the inherited class
    }
}

$grandFa=new GrandFather;

echo $grandFa->getName(),"\n";

// Inherited class Daddy wants to know Grandpas Name
$daddy = new Daddy;
echo $daddy->displayGrandFaName(),"\n"; // Prints 'sahabuddin'

// Public variables can also be accessed outside of the class!
$outsiderWantstoKnowGrandpasName = new GrandFather;
echo $outsiderWantstoKnowGrandpasName->name; // Prints 'sahabuddin'
?>
```

Visibility or Access Modifier in PHP

protected:

Class এর মধ্যে constant, property, method গুলোকে যদি শুধু একই class বা class নিজের মধ্যে এবং তার child class এর মধ্যে ব্যবহার সীমিত রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে protected ঘোষণা করতে হয়।

আমাদের GrandFather class এর \$name নামক Property টি protected হওয়ায়, \$name property টি আমরা একই সাথে current class এবং child class থেকে access করতে পেরেছি কিন্তু current class অথবা child class এর বাহির থেকে access করতে পারবোনা, করলে আপনাকে fatal error দেখাবে।

Visibility or Access Modifier in PHP

private :

Class এর মধ্যে constant, property, method গুলোকে যদি শুধু একই class বা class নিজের মধ্যে ব্যবহার সীমিত রাখতে চাই , তাহলে সেইসব constant, property, method গুলোকে private ঘোষণা করতে হয়।

আমাদের GrandFather class এর \$name নামক Property টি private হওয়ায়, \$name property টি আমরা শুধু মাত্র current class এর ভিতরে ব্যবহার করতে পেরেছি, কিন্তু current class এর বাহির থেকে access বা ব্যবহার করতে পারবোনা, করলে আপনাকে error দেখাবে।

Visibility or Access Modifier in PHP

```
<?php
class GrandFather
{
    private $name='sahabuddin'; // A Protected variable
    public function getName(){
        return $this->name;
    }
}

class Daddy extends GrandFather // Inherited class
{
    function displayGrandFaName()
    {
        return $this->name; // The Private Property will not available to the inherited class and will show notice
    }
}

$grandFa=new GrandFather;

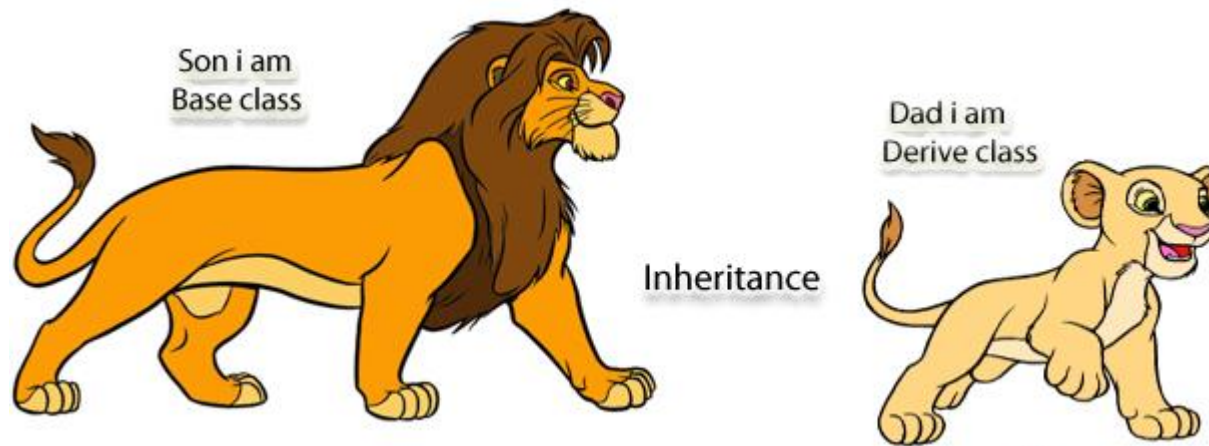
echo $grandFa->getName(),"\n"; //Prints 'Sahabuddin'
echo $grandFa->name, "\n"; //Results in a Notice

// Inherited class Daddy wants to know Grandpas Name
$daddy = new Daddy;
echo $daddy->displayGrandFaName(),"\n"; // Results in a Notice

// Protected Property can not be accessed outside of the class!
$outsiderWantstoKnowGrandpasName = new GrandFather;
echo $outsiderWantstoKnowGrandpasName->name; // // Results in a Fatal Error

?>
```


PHP OOP Inheritance



Inheritance কি?

ইংরেজি শব্দ inheritance অর্থ হচ্ছে উত্তরাধিকার। Object Oriented Programming এ এক class থেকে কিছু ফাংশনালিটি (Methods) ও বৈশিষ্ট্য (Property) অন্য class এ শেয়ার করার একটা পদ্ধতি হচ্ছে Inheritance. বা কোন Class বা Object এ অন্য কোন Class বা Object এর বৈশিষ্ট্য অর্জন করে থাকে যে প্রক্রিয়ায় তাকে উত্তরাধিকার সূত্র বা Inheritance বলে। আরো সহজ ভাবে বলা যায় , একটা ক্লাসকে (Parent Class) Inherit (অনুসরণ) করে তার কিছু বৈশিষ্ট্য আরেকটি উত্তরসূরি (child class) এর মধ্যে ব্যবহার করার প্রক্রিয়াকে Inheritance বলা হয়।

PHP OOP Inheritance

PHP তে কিভাবে child class তৈরী হয় ?

child class তৈরী করতে হলে প্রথমে class কীওয়ার্ড এর পর child class এর নাম দিতে হয়, তারপর extends কীওয়ার্ড লিখে Parent Class এর নাম লিখতে হয়।

```
<?php
class ParentClass {
    // properties and methods here
}

class ChildClass extends ParentClass {
    // additional properties and methods here
}
?>
```

PHP OOP Inheritance

PHP তে Inheritance কিভাবে কাজ করে?

মূলত Parent Class এবং Child Class ধারণার উপর ভিত্তি করে Inheritance ব্যাপারটি আসে। আপনি যখন একটা child class তৈরী করবেন , সেটি তার parent class এর সব protected এবং public Property, Method, Constant গুলো inherit করবে। আরো সহজ ভাবে বলা যায় , একটি class যখন আরেকটি class কে extend করে তখন Parent Class এর সব Protected এবং public Property, Method এবং constant, child class এ define না করেও ব্যবহার বা access করা যায়। পাশাপাশি child class আরো নতুন নতুন property এবং method অন্তর্ভুক্ত করতে পারবে।

PHP OOP Inheritance

```
<?php
class Member {
    public $username = "";
    private $loggedIn = false;
    public function login() {
        $this->loggedIn = true;
    }
    public function logout() {
        $this->loggedIn = false;
    }
    public function isLoggedIn() {
        return $this->loggedIn;
    }
}
?>
```

PHP OOP Inheritance

ব্যাখ্যা: উপরের Member Class টি একটি Web Forum Application এর জন্য করা একটা Class। যেখানে ফোরাম মেম্বারদের জন্য login(), logout(), isLoggedIn() method তৈরী করা হয়েছে। যেহেতু ফোরাম অ্যাডমিনিস্ট্রেটররাও Member, এখন আপনি Member Class থেকে একটি নতুন Child class তৈরী করতে পারেন, যেটার নাম হবে Administrator Class। আর তাতে সুবিধা হচ্ছে Administrator class, Member class এর সব protected এবং public property, Method এবং constant গুলোকে Inherit বা উত্তরাধিকার সূত্রে পাবে। চলুন এবার Member Class থেকে child class হিসেবে Administrator Class টি তৈরী করা যাক:

PHP OOP Inheritance

```
<?php
class Administrator extends Member {
    public function createForum( $forumName ) {
        echo "$this->username created a new forum: $forumName<br>";
    }
    public function banMember( $member ) {
        echo "$this->username banned the member: $member->username<br>";
    }
}
?>
```

PHP OOP Inheritance

ব্যাখ্যা: লক্ষ্য করুন Administrator class এ আমাদের অনেক কাজ নতুন করে করা লাগবেনা , Administrator Class তার Member Parent Class দিয়ে login, logout এবং user logged In কিনা সেগুলো inherit করে নিজে ব্যবহার করতে পারবে। এখন আপনি administrator স্পেসিফিক Method গুলো যেমন : createForum(), deleteForum() এবং banMember() আপনার administrator class এ যোগ করতে পারবেন। আর এতে সুবিধা হচ্ছে আপনি administrator class এ একই সাথে সাধারণ Member Class এর সুবিধা পাচ্ছেন, পাশাপাশি Administrative সুবিধা গুলোও পাচ্ছেন। আর এটাই inheritance. চলুন এবার Parent এবং Child Class দুটির উদাহরণ একসাথে দেখা যাক

PHP OOP Inheritance

```
<?php
class Member {
    public $username = "";
    private $loggedIn = false;
    public function login() {
        $this->loggedIn = true;
    }
    public function logout() {
        $this->loggedIn = false;
    }
    public function isLoggedIn() {
        return $this->loggedIn;
    }
}

class Administrator extends Member {
    public function createForum( $forumName ) {
        echo "$this->username created a new forum: $forumName<br>";
    }
    public function banMember( $member ) {
        echo "$this->username banned the member: $member->username<br>";
    }
}

// Create a new member and log them in
$member = new Member();
$member->username = "Sahab Uddin";
$member->login();
echo $member->username . " is " . ( $member->isLoggedIn() ? "logged in" : "logged out" ) . "<br>";

// Create a new administrator and log them in
$admin = new Administrator();
$admin->username = "Farhan";
$admin->login();
echo $admin->username . " is " . ( $member->isLoggedIn() ? "logged in" : "logged out" ) . "<br>";
// Displays "Mary created a new forum: SAtech360"
$admin->createForum( "SAtech360" );
// Displays "Farhan banned the member: Sahab Uddin"
$admin->banMember( $member );
?>
```

Method Overriding কি?

সাধারণত যখন একটা child class তৈরী হয়, তখন সেটি তার parent class এর সব Protected Property এবং Method গুলোকে Inherit করে নিয়ে আসে। কিন্তু কখনো কখনো আমাদেরকে Parent Class এর যেকোনো Method কে Child Class এ নতুন করে একই নামে লেখার দরকার হয়, এটা সাধারণত তখনই দরকার হয়, যখন দেখা যায়, Parent Class এর Method টি child class এর সব পারপাস পূরণ করতে পারছেনা। অর্থাৎ, আরো কিছু feature add করার দরকার হয়। Object Oriented Programming এর পরিভাষায় এটাকে বলা হয় Method Overriding. এক কোথায়, Parent Class এর কোনো Method কে Child Class এ একই নামে এবং সমান সংখ্যক parameter দিয়ে পুনরায় declare করলে Object Oriented Programming এর পরিভাষায় বলা হয় Method Overriding.

এবার একটা practical উদাহরণ দেওয়া যাক : দরুন আমাদের Member class টির login Method টি শুধু Login কাজটি করে। এখন যদি Administrator Class চায়, কখন কে লগইন করলো তার একটা log Entry রাখা দরকার। আর এর জন্য আমরা Method Overriding Feature টি ব্যবহার করবো। চলুন দেখা যাক :

Method Overriding

```
<?php
class Member {
    public $username = "";
    private $loggedIn = false;
    public function login() {
        $this->loggedIn = true;
    }
    public function logout() {
        $this->loggedIn = false;
    }
}
class Administrator extends Member {
    public function login() {
        $this->loggedIn = true;
        echo "Log entry: $this->username logged in<br>";
    }
}
// Create a new member and log them in
$member = new Member();
$member->username = "Sahab Uddin";
$member->login();
$member->logout();
// Create a new administrator and log them in
$admin = new Administrator();
$admin->username = "Farhan";
$admin->login();      // Displays "Log entry: Farhan logged in"
$admin->logout();
```


PDO (PHP Data Object)

PHP তে PDO হচ্ছে একটা Database Extension বা class যা দিয়ে MySQL, Oracle, MSSQL, PGSQL, DB2 এবং SQLITE সহ বিভিন্ন Database থেকে Data তুলে আনা, Data সম্পাদনা করা, ডিলিট করা এবং নতুন Data ঢুকানোর জন্য ব্যবহৃত হয়। PHP তে PDO ছাড়াও প্রত্যেকটি Database System এর জন্য আলাদা আলাদা Extension যেমন: MySQL Database এর জন্য mysqli, Oracle Database এর জন্য OCI8 ইত্যাদি আছে। PHP তে PDO কে Plug & Play টাইপের Database extension বলা যায়। কেননা PDO extension টি একাই একসাথে সবচেয়ে জনপ্রিয় Database গুলো support করে। আর এতে সুবিধা হল যেকোন সময় এক Database থেকে যদি অন্য কোন ডেটাবেসে সুইচ করতে হয় তাহলে শুধু অল্প কিছু কোড সম্পাদনা করলেই কাজ হয়ে যাবে। তাছাড়া PDO অন্যান্য Database Extension এর তুলনায় অনেক বেশি নিরাপদ এবং দ্রুত।

PDO (PHP Data Object)

PHP তে PDO extension দিয়ে MySQL Database Server এর সাথে connect করার উপায় কি?

PDO দিয়ে MySQL Database Server এর সাথে connect করতে চাইলে, প্রথমে আপনাকে MySQL data source name or DSN : অর্থাৎ, MySQL Database Server এর address দিতে হবে । অর্থাৎ সার্ভারের IP অথবা server name দিতে হবে। যেমন : **127.0.0.1** অথবা **localhost** ব্যবহার করতে পারেন।

তারপর আপনাকে **new** কীওয়ার্ড দিয়ে PDO class এর একটা object তৈরী করতে হবে, এবং PDO class এর constructor Method এ আপনার DSN কে Argument হিসেবে পাঠাতে হবে।

PDO (PHP Data Object)

```
<?php
$dsn="mysql:host=localhost";
try {
    $dbh = new PDO($dsn);
    echo "Connection Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

Note: PHP PDO তে error handling, program flow control এর জন্য try....catch ব্যবহৃত হয়। এটার বিশেষ সুবিধা হচ্ছে error গুলোকে খুব সহজ এবং কার্যকর ভাবে হ্যান্ডলিং করা যায়। আর এতে Project যেমন secured হয় , এবং performance ও ভালো হয়। try...catch এর সাধারণ নিয়ম হচ্ছে আপনার code run হবে try block এর মধ্যে , যদি code এ কোনো সমস্যা হয়, তাহলে program স্বয়ংক্রিয় ভাবে catch block এ যাবে, PDOException Class এর বিভিন্ন Method দিয়ে customize error message দেখানো যাবে।

PDO (PHP Data Object)

PHP তে PDO extension দিয়ে MySQL Database Server এ Database তৈরী করার উপায় কি?

Database Server এর সাথে তো connect করলেন? কিন্তু Database তৈরী করবেন না? আর হ্যাঁ Database তৈরী করতে চাইলে Database তৈরী করার পারমিশন আছে এমন যেকোনো একটা user হিসেবে access থাকতে হবে। আর Database তৈরী করার পারমিশন আছে এমন যেকোনো একটা user হিসেবে access করতে হলে আপনাকে new PDO Object এ আরো দুটি parameter হিসেবে সেই user এর নাম এবং পাসওয়ার্ড দিতে হবে।

```
<?php
$dsn="mysql:host=localhost";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    echo "Connection Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

PDO (PHP Data Object)

Database Server এর সাথে connect হলো, user হিসেবে access হলো , এবার চলুন DSN, Username, Password ব্যবহার করে PHP PDO দিয়ে একটা Database তৈরী করা যাক। আর PHP দিয়ে MySQL Database Server এ Database তৈরী সহ যেকোনো কাজ করতে চাইলে আপনাকে PDO class এর query Method এর মধ্যে SQL (Standard Query Language) লিখে একে call করতে হবে। এখানে আমরা testDB নামে একটা Database তৈরী করার জন্য “CREATE DATABASE IF NOT EXISTS testDB” query টি ব্যবহার করব।

```
<?php
$dsn="mysql:host=localhost";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    $dbh->query('CREATE DATABASE IF NOT EXISTS testDB') or die("Database Create Fail!");
    echo "Database Create Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```


PDO (PHP Data Object)

PHP তে PDO extension দিয়ে MySQL Database এ table তৈরী করার উপায় কি?

Database Server এর সাথে connect হলো, user হিসেবে access হলো , Database ও তৈরী হলো, এখন PHP PDO দিয়ে যেকোনো MySQL Database এ table তৈরী করতে চাইলে, প্রথমে আপনাকে Database কে connect করতে হবে । আর database connect করার নিয়ম হচ্ছে \$dsn string এ hostname বা IP এর পর semicolon (;) দিয়ে Database এর নাম দিতে হবে।

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    echo "Database Connection Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

PDO (PHP Data Object)

Database এর সাথে যদি successfully connect হয় , তাহলে এবার আপনার কাজ হচ্ছে একটা table তৈরী করা । আর PHP দিয়ে MySQL Database এ table তৈরী করতে চাইলে Database তৈরির মতোই আপনাকে PDO class এর query Method এর মধ্যে Table তৈরির SQL (Standard Query Language) লিখে একে call করতে হবে।

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";$user="root";$pass="";$sql=<<<"table"
CREATE TABLE `students` (
  `id` INT NOT NULL AUTO_INCREMENT , `name` VARCHAR(20) NOT NULL , `email` VARCHAR(100) NOT NULL ,
  `mobile` VARCHAR(20) NOT NULL , PRIMARY KEY (`id`(4)))
ENGINE = InnoDB;
table;
try {
  $dbh = new PDO($dsn,$user,$pass);
  $dbh->query($sql) or die("Table Create fail");
  echo "Table Create Success";
} catch (PDOException $e) {
  print "Error!: " . $e->getMessage() . "<br/>";
  die();
}
```

PDO (PHP Data Object)

Prepared Statement কি?

Database Management System এ Prepared Statement হচ্ছে অনেকটা SQL Query Template এর মত। যা SQL query এর মাধ্যমে বিভিন্ন Parameter এর actual Value এর পরিবর্তে ভিন্ন ভিন্ন Placeholder (“?” or :placeholderName) দিয়ে Database এ পাঠায়। পরবর্তীতে Database Management System তখন SQL Query Template টির উপর Parse, Compile, Syntax Check এবং Query Optimization করে এবং Execute ব্যতীত ফলাফলটি পরবর্তীতে ব্যবহারের জন্য সংরক্ষণ করে। পরবর্তীতে application টি value গুলোকে Parameter এ Bind করে, এবং DBMS স্টেটমেন্টটিকে Execute করে। Application টি statement কে প্রত্যেকবার ভিন্ন ভিন্ন value দ্বারা যতবার ইচ্ছে execute করতে পারে।

PDO (PHP Data Object)

Prepared Statement ব্যবহার করলে কি সুবিধা পাওয়া যাবে?

- Prepared statement পার্সিং টাইমকে কমিয়ে দেয় কারণ এটি query একবারই execute করে (যদিও স্টেটমেন্টটি অসংখ্যবার execute করা যায়)। ব্যাপারটা অনেকটা প্রতিদিন বাজার করার জন্য ঘর থেকে বাহির হওয়ার দরকার নাই, ঘরে বসে online এ অর্ডার দিলেই সেই পণ্যটি ঘরেই চলে আসবে, আর এতে সময় এবং শ্রম দুইটাই বাঁচবে।
- Bind parameter গুলো সার্ভারের Bandwidth load কমিয়ে দেয়, যেহেতু আপনাকে শুধু Parameter গুলোকেই বার বার পাঠাতে হয়, সম্পূর্ণ query পাঠাতে হয়না।
- Prepared statement গুলো SQL Injection এর বিরুদ্ধে খুবই কার্যকর ভূমিকা রাখে। SQL Injection হলো এক ধরনের Database Injection কোড যা Database এ আক্রমণ ও তথ্য চুরির কাজে ব্যবহার করা হয়। আর Prepared Statement এর প্যারামিটারের value গুলি যেহেতু সরাসরি বহিরাগত input থেকে আসেনা যা পরবর্তীতে একটি ভিন্ন প্রোটোকল ব্যবহার করে প্রেরন করা হয়। যেহেতু মূল SQL এর value গুলো সরাসরি বহিরাগত input থেকে পাওয়া যায় না। তাই SQL Injection ঘটতে পারে না।

PDO (PHP Data Object)

আমরা একটা Table তৈরী করব এবং Table এ কিছু Data Insert করব। আর এর জন্য আপনাকে নিচের কোডটিতে আপনার Database Username, Password এবং dsn এ localhost এর পর সেমিকোলন (;) দিয়ে Database এর নামটি সঠিকভাবে দিয়ে execute করলেই হবে।

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
$sql=<<<<"table"
CREATE TABLE IF NOT EXISTS `students` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(20) NOT NULL ,
  `email` VARCHAR(100) NOT NULL ,
  `mobile` VARCHAR(20) NOT NULL ,
  PRIMARY KEY (`id`(4)))
ENGINE = InnoDB;
INSERT INTO `students` (`id`, `name`, `email`, `mobile`) VALUES
  (NULL, 'Noman Khan', 'noman@mail.com', '01899009988'),
  (NULL, 'Sahin Khan', 'sahin@mail.com', '01900887766'),
  (NULL, 'Firoj Khan', 'firon@mail.com', '01900998877'),
  (NULL, 'Rashed Khan', 'rashed@test.com', '01566778899'),
  (NULL, 'Sohag Khan', 'sohag@mail.com0', '1788993344')
table;
try {
  $dbh = new PDO($dsn,$user,$pass);
  $dbh->query($sql) or die("fail");
  echo "Table Create Success";
} catch (PDOException $e) {
  print "Error!: " . $e->getMessage() . "<br/>";
  die();
}
```


PDO (PHP Data Object)

bindColumn function

PHP Prepared Statement এ SQL statement এর মধ্যে Column name অথবা Column Number এর বিপরীতে যেকোনো variable কে bind করার জন্য bindColumn() function ব্যবহৃত হয়।

```
<?php
function readData($dbh) {
    $sql = 'SELECT name, colour, calories FROM fruit';
    try {
        $stmt = $dbh->prepare($sql);
        $stmt->execute();

        /* Bind by column number */
        $stmt->bindColumn(1, $name);
        $stmt->bindColumn(2, $colour);

        /* Bind by column name */
        $stmt->bindColumn('calories', $cals);

        while ($row = $stmt->fetch(PDO::FETCH_BOUND)) {
            $data = $name . "\t" . $colour . "\t" . $cals . "\n";
            print $data;
        }
    }
    catch (PDOException $e) {
        print $e->getMessage();
    }
}
readData($dbh);
?>
```

PDO (PHP Data Object)

bindParam() Function

PHP Prepared Statement এ SQL statement এর মধ্যে Name placeholder অথবা Question Mark (“?”) Placeholder এর বিপরীতে যেকোনো variable কে bind করার জন্য bindParam() function ব্যবহৃত হয়।

bindValue() Function

PHP Prepared Statement এ SQL statement এর মধ্যে Name placeholder অথবা Question Mark (“?”) Placeholder এর বিপরীতে যেকোনো value কে bind করার জন্য bindValue() function ব্যবহৃত হয়।

PHP PDO তে bindParam এবং bindValue এর মধ্যে কোনো পার্থক্য আছে?

যদিও bindValue এবং bindParam function দুটির কাজ হচ্ছে SQL statement এর মধ্যে Name placeholder অথবা Question Mark (“?”) Placeholder এর বিপরীতে যেকোনো variable কে bind করা , কিন্তু খুবই গুরুত্বপূর্ণ একটা পার্থক্য আছে, আর তা হচ্ছে, bindValue শুধু variable এর value কে bind করে, অনেকটা hard copy বা static এর মতো, অর্থাৎ , একবার যদি PDO Prepared Statement এ value bind হয়ে যায় , তাহলে variable এর মান পরিবর্তন হলেও SQL Statement এর আউটপুট এ কোনো পরিবর্তন আসবেনা।

অন্যদিকে bindParam function টি Variable এর reference কে bind করে। অর্থাৎ ,PDO Prepared Statement এ Variable এর value bind হয়ে যাওয়ার পর যদি আবার variable এর মানে কোনো পরিবর্তন আসে , তাহলে SQL Statement এর ফলাফলেও পরিবর্তন আসবে।

PDO (PHP Data Object)

bindParam function দিয়ে উদাহরণ:

```
$sex = 'male';  
$s = $dbh->prepare('SELECT name FROM students WHERE sex = :sex');  
$s->bindParam(':sex', $sex); // use bindParam to bind the variable  
$sex = 'female';  
$s->execute(); // executed with WHERE sex = 'female'
```

bindValue function দিয়ে উদাহরণ:

```
$sex = 'male';  
$s = $dbh->prepare('SELECT name FROM students WHERE sex = :sex');  
$s->bindValue(':sex', $sex); // use bindValue to bind the variable's value  
$sex = 'female';  
$s->execute(); // executed with WHERE sex = 'male'
```

PDO (PHP Data Object)

Insert Operation এ placeholder name দিয়ে Prepared statement এর ব্যবহার

INSERT Operation এ placeholder name দিয়ে Prepared Statement ব্যবহার করতে চাইলে, আপনাকে PDO Class এর prepare method এর মধ্যে INSERT Query টি লিখতে হবে এবং INSERT Query এর মধ্যে Value গুলোর পরিবর্তে placeholder name দিতে হবে। তারপর আবার PDOStatement class এর bindParam Method দিয়ে প্রত্যেকটি placeholder এর বিপরীতে value গুলোকে bind করে দিতে হবে।

নোট: PDO Class এর query,prepare method call করার পর pdoStatement Object তৈরী হয়।

PDO (PHP Data Object)

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    $stmt = $dbh->prepare("INSERT INTO students (name, email, mobile) VALUES (:name, :email, :mobile)");
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':email', $email);
    $stmt->bindParam(':mobile', $mobile);
    // insert one row
    $name = 'Habibur Rahman';
    $email = "habib@mail.com";
    $mobile = "01899887766";
    $stmt->execute();
    echo "Insert First Row Success<br>";
    // insert another row with different values
    $name = 'Tareq Rahman';
    $email = "tareq@mail.com";
    $mobile = "01659887766";
    $stmt->execute();
    echo "Insert Second Row Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```


PDO (PHP Data Object)

ব্যাখ্যা:লাইন নম্বর ৭ এ লক্ষ্য করুন আমরা INSERT Query এর মধ্যে সরাসরি value না লিখে placeholder Name দিয়ে Query টি কে Prepare করে নিয়েছি, তারপর লাইন নম্বর ৮,৯,১০ এ placeholder গুলোতে variable গুলোকে bind করে দিয়েছি। তারপর আমরা ১৫ এবং ২১ নম্বর লাইনে আমরা নতুন নতুন value assign করে execute করি। আর এতে সুবিধা হচ্ছে আমাদের প্রত্যেক বার SQL টি call করা লাগছেনা , শুধু value পাঠিয়ে দিলেই হচ্ছে।

PDO (PHP Data Object)

Insert Operation এ (“?”) দিয়ে Prepared statement এর ব্যবহার

INSERT Operation এ (“?”) দিয়ে Prepared Statement ব্যবহার করতে চাইলে, আপনাকে PDO Class এর prepare method এর মধ্যে INSERT SQL টি লিখতে হবে এবং INSERT SQL এর মধ্যে Value গুলোর পরিবর্তে (“?”) দিতে হবে। তারপর আবার PDOStatement class এর bindParam Method দিয়ে প্রত্যেকটি placeholder Position Number এর বিপরীতে value গুলোকে bind করে দিতে হবে।

ব্যাখ্যা:লাইন নম্বর ৭ এ লক্ষ্য করুন আমরা INSERT Query এর মধ্যে সরাসরি value না লিখে (“?”) দিয়ে Query টি কে Prepare করে নিয়েছি, তারপর লাইন নম্বর ৮,৯,১০ এ placeholder গুলোতে variable গুলোকে bind করে দিয়েছি। তারপর আমরা ১৫ এবং ২১ নম্বর লাইনে আমরা নতুন নতুন value assign করে execute করি। আর এতে সুবিধা হচ্ছে আমাদের প্রত্যেক বার SQL টি call করা লাগছেনা , শুধু value পাঠিয়ে দিলেই হচ্ছে।

PDO (PHP Data Object)

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    $stmt = $dbh->prepare("INSERT INTO students (name, email, mobile) VALUES (?, ?, ?)");
    $stmt->bindParam(1, $name);
    $stmt->bindParam(2, $email);
    $stmt->bindParam(3, $mobile);
    // insert one row
    $name = 'Habibur Rahman';
    $email = "habib@mail.com";
    $mobile = "01899887766";
    $stmt->execute();
    echo "Insert First Row Success<br>";
    // insert another row with different values
    $name = 'Tareq Rahman';
    $email = "tareq@mail.com";
    $mobile = "01659887766";
    $stmt->execute();
    echo "Insert Second Row Success";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

PDO (PHP Data Object)

SELECT Operation এ placeholder name দিয়ে Prepared statement এর ব্যবহার

SELECT Operation এ placeholder name দিয়ে Prepared Statement ব্যবহার করতে চাইলে, আপনাকে PDO Class এর prepare method এর মধ্যে SELECT Query টি লিখতে হবে এবং SELECT Query এর মধ্যে WHERE condition এর Value গুলোর পরিবর্তে placeholder name দিতে হবে। তারপর আবার PDOStatement class এর bindParam Method দিয়ে প্রত্যেকটি placeholder এর বিপরীতে value গুলোকে bind করে দিতে হবে।

PDO (PHP Data Object)

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    $stmt = $dbh->prepare("SELECT * FROM students WHERE id=:id");
    $stmt->bindParam("id",$id);
    // insert one row
    $id = 2;
    $stmt->execute();
    echo "<pre>";
    print_r($stmt->fetch(PDO::FETCH_ASSOC));
    echo "</pre>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```


PDO (PHP Data Object)

Select Operation এ (“?”) দিয়ে Prepared statement এর ব্যবহার

SELECT Operation এ (“?”) দিয়ে Prepared Statement ব্যবহার করতে চাইলে, আপনাকে PDO Class এর prepare method এর মধ্যে QUERY টি লিখতে হবে এবং SELECT QUERY এর মধ্যে Value গুলোর পরিবর্তে (“?”) দিতে হবে।

তারপর আবার PDOStatement class এর bindParam Method দিয়ে প্রত্যেকটি placeholder Position Number এর বিপরীতে value গুলোকে bind করে দিতে হবে।

PDO (PHP Data Object)

```
<?php
$dsn="mysql:host=localhost;dbname=testDB";
$user="root";
$pass="";
try {
    $dbh = new PDO($dsn,$user,$pass);
    $stmt = $dbh->prepare("SELECT * FROM students WHERE id=?");
    $stmt->bindParam(1,$id);
    // insert one row
    $id = 2;
    $stmt->execute();
    echo "<pre>";
    print_r($stmt->fetch(PDO::FETCH_ASSOC));
    echo "</pre>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

CRUD

CRUD কি?

Computer Programming এ CRUD হচ্ছে **Create+Read+Update+Delete** এই চারটি শব্দের প্রথম (যথাক্রমে Create=C, Read=R, Update=U, Delete=D) Letter দিয়ে গঠিত সংক্ষিপ্ত রূপ। অর্থাৎ আপনি যখন কোনো web site অথবা সফটওয়্যার তৈরী করবেন, তখন software এর তথ্য গুলো Database এ রাখার জন্য প্রথমে Database, table এবং কিছু Data Create বা তৈরী করে নিতে হয়, আর তখন এটাকে আমরা CRUD এর **Create Operation** বলি। আবার আমাদের Create করা Data কে যখন browser এ দেখাই বা Read করি, তখন এটাকে আমরা CRUD এর **Read Operation** বলি। একই ভাবে Create করা Data কে যখন browser দিয়ে **Update** এবং **Delete** করি, তখন এটাকে আমরা CRUD এর **Update** এবং **Delete Operation** বলি। Computer Programming এ CRUD Operation কে যেকোনো software অথবা Web Development এর Engine হিসেবে বিবেচনা করা হয়।

OOP Encapsulation এবং Polymorphism ধারণাকে কাজে লাগিয়ে CRUD Class তৈরী করা :

প্রথমে বুঝা যাক, Encapsulation এবং Polymorphism এর ধারণা বলতে কি বুঝি ? Encapsulation-অর্থাৎ সব কিছু capsule অবস্থায় থাকা। অনেকটা ক্যাপসুল মেডিসিন এর মতোই , যেভাবে একটা capsule এর বাইরে একটা আবরণ দিয়ে সব কিছু ভেতরে আটকানো থাকে । আর Polymorphism মানে হচ্ছে বহুরূপিতা। অর্থাৎ Class এবং এর Method গুলো বহুরূপে ব্যবহৃত হতে পারে। আর এইভাবে ভাবে এখানে Encapsulation এর ধারণাকে কাজে লাগিয়ে CRUD এর সব Operation (যথাক্রমে C => Create, R => Read, U => Update এবং D => Delete) গুলোকে আমরা Capsulate করে রাখবো। তারপর Polymorphism ধারণাকে কাজে লাগিয়ে এগুলোর বহুরূপে ব্যবহার নিশ্চিত করব।

CRUD

PDO দিয়ে MySQL Database Host এর সাথে Connect

PHP PDO class থেকে একটা Object তৈরী করার মধ্যে দিয়ে আমরা MySQL Database Host এর সাথে কানেক্ট করতে পারি। এর জন্য আপনাকে PDO Class এর Parameter হিসেবে DSN (Data Source Name), Database hostname, Username এবং Password দিতে হবে। আর এই কাজটি আমরা একটা Crud নামক class এর constructor মেথড দিয়ে করব। তবে আপনি চাইলে আলাদা মেথড এর মধ্যেও কাজটি করতে পারেন।

ব্যাখ্যা: আমরা চাচ্ছি class থেকে object তৈরী করার সাথে সাথে Database Host এর connect হব। আর তাই Database এর connection কাজটি constructor Method এ করেছি। এবং Constructor Method এর Parameter হিসেবে DSN (Data Source Name), Database hostname, Username এবং Password গ্রহণ করি। এবং \$dsn variable টি তৈরী করি। তারপর new PDO object তৈরী করি এবং Database Host এর সাথে connect হই। তবে যদি কোনো কারণে Database Host এর সাথে connect না হয়, তাহলে নিচের মতো আউটপুট দেখাবে।

CRUD

```
<?php
class Crud{
    private $conn;
    public function __construct($host,$user,$pass){
        $dsn="mysql:host=$host;";
        try {
            $this->conn = new PDO($dsn,$user,$pass);
            echo "Connection Success";
        }catch(PDOException $e) {
            die("Error!: ". $e->getMessage());
        }
    }
}

$db=new Crud("localhost","root","");
?>
```


CRUD

PDO দিয়ে Database তৈরী

PHP PDO দিয়ে Database তৈরী করতে হলে আপনাকে PDO Class এর prepare() অথবা query() Method এর মধ্যে “CREATE DATABASE DatabaseName” Query রান করার মাধ্যমে Database তৈরী করতে হবে। আমরা আমাদের Crud Class এর মধ্যে নতুন Database তৈরির সুবিধা দেওয়ার জন্য একটা createDatabase নামে একটি method তৈরী করব।

ব্যাখ্যা: লাইন নম্বর ১৪ তে আমরা createDatabase Method এ user থেকে Database Name এবং collation গ্রহণ করার জন্য \$dbName এবং \$collation নামে দুইটি Parameter ব্যবহার করেছি। তারপর \$sql variable এর মধ্যে database তৈরির SQL তৈরী করি। লাইন নম্বর ২০ এবং ২১ এ যথাক্রমে prepare এবং execute করি। এবং ২২-২৮ নম্বর লাইনে আমরা database তৈরী হয়েছে কিনা সেটা চেক করব।

CRUD

```
<?php
class Crud{
    private $conn;
    public function __construct($host,$user,$pass){
        $dsn="mysql:host=$host;";
        try {
            $this->conn = new PDO($dsn,$user,$pass);
            //echo "Connection Success";
        }catch(PDOException $e) {
            die("Error!: ". $e->getMessage());
        }
    }

    public function createDatabase($dbName,$collation="utf8_general_ci"){
        $sql=<<<"db"
        CREATE DATABASE $dbName
        DEFAULT CHARACTER SET utf8
        DEFAULT COLLATE $collation;
db;
        $stmt=$this->conn->prepare($sql);
        $stmt->execute();
        if($stmt->errorCode()=="00000"){
            echo "Database Create Success";
        }
        else{
            die($stmt->errorInfo()[2]);
        }
    }
}

$db=new Crud("localhost","root","");
$db->createDatabase("crud2");
?>
```

CRUD

PDO দিয়ে MySQL Database এর একটি নির্দিষ্ট Database কে ব্যবহার বা Select
PHP PDO দিয়ে একটি নির্দিষ্ট Database Select অথবা Use করতে হলে আপনাকে PDO Class এর prepare() অথবা query() Method এর মধ্যে "USE DatabaseName" Query রান করার মাধ্যমে Database সিলেক্ট করতে হবে। আমরা আমাদের Crud Class এর মধ্যে নতুন Database সিলেক্ট সুবিধা দেওয়ার জন্য একটা selectDatabase নামে একটি method তৈরী করব। যার কাজ হবে এই class ব্যবহারকারী নিজের ইচ্ছা মতো database তৈরী করতে পারবে।

CRUD

```
<?php
class Crud{
    private $conn;
    public function __construct($host,$user,$pass){
        $dsn="mysql:host=$host;";
        try {
            $this->conn = new PDO($dsn,$user,$pass);
            //echo "Connection Success";
        }catch(PDOException $e) {
            die("Error!: ". $e->getMessage());
        }
    }

    public function selectDatabase($dbName){
        $sql="use $dbName";
        $stmt=$this->conn->prepare($sql);
        $stmt->execute();
        if($stmt->errorCode()!="00000"){
            die($stmt->errorInfo()[2]);
        }
        return "Database Selected";
    }
}

$db=new Crud("localhost","root","");
echo $db->selectDatabase("crud2");
?>
```

CRUD

PDO দিয়ে একটি নির্দিষ্ট Database এর মধ্যে একটি table তৈরী করা:

PHP PDO দিয়ে একটি নির্দিষ্ট Database এ যেকোনো table তৈরী করতে হলে আপনাকে PDO Class এর prepare() অথবা query() Method এর মধ্যে CREATE TABLE tableName” Query রান করার মাধ্যমে table তৈরী করতে হবে । আমরা আমাদের Crud Class এর মধ্যে একটি নির্দিষ্ট Database এর মধ্যে নতুন table তৈরীর সুবিধা দেওয়ার জন্য একটা createTable নামে একটি method তৈরী করব ।

ব্যাখ্যা: আমরা createTable Method এর ২৫ থেকে ৩০ নম্বর লাইনে আমাদের সিলেক্টেড database এ টেবিল তৈরী করার জন্য একটা SQL কোয়েরি তৈরী করি। তারপর সেটিকে ৩১ এবং ৩২ নম্বর লাইনে prepare এবং execute করি। তারপর ৩৩ এবং ৩৪ নম্বর লাইনে আমরা আমাদের টেবিল তৈরী ঠিকভাবে হয়েছে কিনা তা চেক করি।

CRUD

```
<?php
class Crud{
    private $conn;
    public function __construct($host,$user,$pass){
        $dsn="mysql:host=$host;";
        try {
            $this->conn = new PDO($dsn,$user,$pass);
            //echo "Connection Success";
        }catch(PDOException $e) {
            die("Error!: " . $e->getMessage());
        }
    }

    public function selectDatabase($dbName){
        $sql="use $dbName";
        $stmt=$this->conn->prepare($sql);
        $stmt->execute();
        if($stmt->errorCode()!="00000"){
            die($stmt->errorInfo()[2]);
        }
        //return "Database Selected";
    }

    public function createTable(string $table, array $fields,$primary_key=""){
        $sql = "CREATE TABLE `$table` (";
        foreach ($fields as $definition) {
            $sql.= $definition['column_name']. ' '.$definition['column_type']. ',';
        }
        $sql.= 'date_added' . " DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,";
        $sql.= "PRIMARY KEY ($primary_key));";
        $stmt=$this->conn->prepare($sql);
        $stmt->execute();
        if($stmt->errorCode()!="00000"){
            die($stmt->errorInfo()[2]);
        }
        return "table $table Create Success";
    }
}

$db=new Crud("localhost","root","");
echo $db->selectDatabase("crud2");

$column_data = array(
    array(
        'column_name' => 'id',
        'column_type' => 'INT UNSIGNED NOT NULL AUTO_INCREMENT'
    ),
    array(
        'column_name' => 'name',
        'column_type' => 'VARCHAR(20)'
    ),
    array(
        'column_name' => 'color',
        'column_type' => 'VARCHAR(20)'
    )
);
echo $db->createTable("fruits",$column_data,"id");
?>
```

CRUD

PDO দিয়ে একটি নির্দিষ্ট table এর মধ্যে Data Insert:

PHP PDO দিয়ে একটি নির্দিষ্ট Database এ যেকোনো table এ Data Insert করতে হলে আপনাকে PDO Class এর prepare() অথবা query() Method এর মধ্যে “INSERT INTO tableName” Query রান করার মাধ্যমে table এ Data Insert করতে হবে। আমরা আমাদের Crud Class এর মধ্যে একটি নির্দিষ্ট Database এর মধ্যে একটি নির্দিষ্ট table এ Data Insert করার সুবিধা দেওয়ার জন্য Insert() নামে একটি method তৈরী করব।

ব্যাখ্যা লাইন নম্বর ২৬ এ আমরা Insert Method এ দুটি parameter নেই। তারমধ্যে প্রথমটি হচ্ছে Table এর নাম, এবং দ্বিতীয়টি হচ্ছে একটা associative array, যেই array টির index এর নাম গুলো টেবিল এর column এর নামের সাথে মিলে রাখবো। আর এই array টাকেই আমরা লাইন নম্বর ২৭-৩৭ SQL Query তে convert করি, যা পরবর্তীতে লাইন নম্বর ৩৯ এ SQL টিকে Prepare করি, এবং লাইন নম্বর ৪০-৪২ এ dynamically bindParam করি। finally লাইন নম্বর ৪৩ এ execute করি এবং লাইন নম্বর ৪৪ এ data insert হয়েছে কিনা তা চেক করি।

CRUD

```
<?php
class Crud{
    private $conn;
    public function __construct($host,$user,$pass){
        $dsn="mysql:host=$host;";
        try {
            $this->conn = new PDO($dsn,$user,$pass);
            //echo "Connection Success";
        }catch(PDOException $e) {
            die("Error!: ". $e->getMessage());
        }
    }

    public function selectDatabase($dbName){
        $sql="use $dbName";
        $stmt=$this->conn->prepare($sql);
        $stmt->execute();
        if($stmt->errorCode()!="00000"){
            die($stmt->errorInfo()[2]);
        }
        //return "Database Selected";
    }

    public function Insert($table,array $data){
        $sql="INSERT INTO $table ( ";
        foreach($data as $col=>$val){
            $sql.=" $col,";
        }
        $sql= substr($sql,0,-1);
        $sql.=") VALUES ( ";
        foreach($data as $col=>$val){
            $sql.=" :$col,";
        }
        $sql= substr($sql,0,-1);
        $sql.=" )";

        $stmt = $this->conn->prepare($sql);
        foreach($data as $column=>$value){
            $stmt->bindParam($column, $value);
        }
        $stmt->execute();
        if($stmt->rowCount()>0){
            echo "Data Insert Success";
        }
        else{
            die("Data Insert Fail!");
        }
    }
}

$db=new Crud("localhost","root","");
echo $db->selectDatabase("crud2");
$data=array(
    "name"=>"Banana",
    "color"=>"yellow"
);

$db->Insert("fruits",$data);
?>
```

PHP OOP FORM Builder

বিভিন্ন software এবং Web Development Project এ অনেক সময় আমাদেরকে একই কাজ বারবার করতে হয়, আর সেই জন্য একজন Programmer কে একই code বারবার লিখতে হয়। আর একই কোড বারবার লিখার কারণে আমাদের Project Development এর সময় যেমন বেশি লাগে একই ভাবে খরচ ও অনেক বেশি লাগে। আমরা যদি বারবার করা লাগে এমন কাজ গুলো কোনো class এর method এ process করে রাখি, তাহলে আমরা খুব সহজে সেই class এর Object দিয়ে method গুলোকে call করতে পারি। এতে আমাদের Project Development এর সময় এবং খরচ কমিয়ে আনতে পারি। CRUD Operation এর Form এর কাজ গুলো সাধারণত বারবার ব্যবহার হয়, আর তাই form এর কাজ গুলো আমরা নতুন একটা FormBuilder Class এর মধ্যে আলাদা আলাদা Method এ লিখে রাখব। যা দিয়ে আমরা Form এর মধ্যে বিভিন্ন input field তৈরী করতে পারব।

PHP OOP FORM Builder

PHP Constructor এবং Destructor Method দিয়ে Form Start এবং Form End

HTML input field গুলো তৈরী করার আগে আমাদেরকে প্রথমে form start এবং শেষে form close করে নিতে হয়। আর এই দুটি কাজের মধ্যে form start টা আমরা class এর constructor Method দিয়ে করব। কারণ আমরা জানি constructor Method সবার আগে execute হয়, একই ভাবে form close টা destructor Method দিয়ে করব। কেন destructor Method টা সবার শেষে execute হয়।

ব্যাখ্যা: পাঠক লক্ষ্য করুন , আমরা সাধারণ ভাবে যখন কোনো ফর্ম তৈরী করার সময় যেইভাবে form শুরু এবং শেষ করি, এখানেও আমরা constructor এবং destructor method এ যথাক্রমে form শুরু এবং শেষ করি। তবে form start করার জন্য যেসব তথ্য দরকার তা আমরা constructor method এর Parameter হিসেবে গ্রহণ করবো। যেমন আমরা এখানে action, method, form name এবং form এর জন্য css_class গ্রহণ করেছি।

PHP OOP FORM Builder

```
<?php
class FormBuilder{
    public function __construct($action,$method='post',$name='form',$css_class='form-control'){
        echo '<form class="'. $css_class. '" action="'. $action. '" method="'. $method. '" id="'. $name. '">'. "\n";
    }

    public function __destruct(){
        echo '</form>'. "\n";
    }
}

$form=new FormBuilder("process.php");
?>
```

PHP OOP FORM Builder

<input type="..." \> তৈরী করার Method

এখন আমরা inputField নামে একটা method বানাবো, যার কাজ হচ্ছে <input> দিয়ে যত form input বানানো হয় তার সবগুলো তৈরী করবে।

```
public function inputField($label="", $type, $name, $value="", $input_css_class='form-control'){  
    $form_input=$label.':<input type="'. $type.'" class="'. $input_css_class.'" name="'. $name.'" id="'. $name.'" value="'. $value.'" />';  
    echo $form_input;  
}
```

ব্যাখ্যা: উপরের inputField Method এর মধ্যে যেকোনো ধরনের input field তৈরী করার জন্য যত রকমের input field যেমন : label, input type, name, value, এবং input field এর জন্য css class সব গুলোকে আমরা parameter হিসেবে গ্রহণ করার সুযোগ রেখেছি, এবং যেগুলো Non-mandatory সে গুলুতে একটা default value দিয়ে দিয়েছি।

PHP OOP FORM Builder

<textarea>....</textarea> তৈরী করার Method

এখন আমরা textareaField নামে একটি method বানাবো, যার কাজ textarea form input বানানো।

```
public function textareaField($label="", $name, $value, $rows=3, $cols=20, $input_css_class='form-control'){  
  
    $form_input=$label.': <textarea class="'. $input_css_class.'" rows="'. $rows.'" $cols="'. $cols.'" name="'. $name.'" id="'. $name.'">'. $value.'</textarea>';  
    echo $form_input;  
  
}
```

ব্যাখ্যা: উপরের textareaField Method এর মধ্যে textarea field তৈরী করার জন্য যত রকমের input field যেমন : label, name, value, rows, column এবং input field এর জন্য css class সব গুলোকে আমরা parameter হিসেবে গ্রহণ করার সুযোগ রেখেছি, এবং যেগুলো Non-mandatory সে গুলোতে একটি default value দিয়ে দিয়েছি।

PHP OOP FORM Builder

<select>....</select> তৈরী করার Method

```
public function selectField($label,$name, array $value=array(),$default="", $input_css_class='form-control'){
    $r =$label.': <select class="form-control" name="'.$name.'" id="'.$name.'">';
    if(is_array($value)){
        foreach($value as $k=>$v){
            if($k === $default)
                $r .= "<option value='$k' selected>$v</option>\n";
            else
                $r .= "<option value='$k'>$v</option>\n";
        }
    }
    $r .='</select>';
    echo $r;
}
```

ব্যাখ্যা: উপরের selectField Method এর মধ্যে select field তৈরী করার জন্য যত রকমের input field যেমন : label,value এবং input field এর জন্য css class সব গুলোকে আমরা parameter হিসেবে গ্রহণ করার সুযোগ রেখেছি, এবং যেগুলো Non-mandatory সে গুলুতে একটা default value দিয়ে দিয়েছি।

PHP OOP FORM Builder

```
<?php
class FormBuilder{
    public function __construct($action,$methode='post',$name='form',$css_class='form-control'){
        echo '<form class="'.$css_class.'" action="'.$action.'" method="'.$methode.'" id="'.$name.'">';
    }

    public function __destruct(){
        echo '</form>';
    }

    public function inputField($label="", $type, $name, $value="", $input_css_class='form-control'){
        $form_input=$label.<input type="'.$type.'" class="'.$input_css_class.'" name="'.$name.'" id="'.$name.'" value="'.$value.'" />;
        echo $form_input;
    }

    public function textareaField($label="", $name, $value, $rows=3, $cols=20, $input_css_class='form-control'){
        $form_input=$label.<textarea class="'.$input_css_class.'" rows="'.$rows.'" cols="'.$cols.'" name="'.$name.'" id="'.$name.'">'.$value.</textarea>;
        echo $form_input;
    }

    public function selectField($label, $name, array $value=array(), $default="", $input_css_class='form-control'){
        $r=$label.<select class="form-control" name="'.$name.'" id="'.$name.'">;
        if(is_array($value)){
            foreach($value as $k=>$v){
                if($k== $default)
                    $r.="<option value='$k' selected>$v</option>\n";
                else
                    $r.="<option value='$k'>$v</option>\n";
            }
        }
        $r.<\/select>;
        echo $r;
    }
}

$form=new FormBuilder("process.php");

$form->inputField("Name :", "text", "firstName");
echo "<br>";
$form->inputField("Email :", "email", "email");
echo "<br>";
$form->inputField("Password :", "password", "password");
echo "<br>";
echo "Gender :";
echo "Male " . $form->inputField("", "radio", "gender", "Male");
echo "Female " . $form->inputField("", "radio", "gender", "Female");
echo "<br>";
$form->textareaField("Address", "address", "", 3, 20);
echo "<br>";
$form->selectField("Country", "country", ["Bangladesh", "India", "Pakistan", "Nepal", "Srilanka"]);
echo "<br>";
echo "Remember Me " . $form->inputField("", "checkbox", "remember");
echo "<br>";
$form->inputField("", "submit", "submit", "Submit");
$form->inputField("", "reset", "reset", "Reset");
```




PHP OOP FORM Builder

[illegible]

WordPress Theme Development

WordPress Theme হচ্ছে কিছু ফাইল এর কালেকশন যা ওয়ার্ডপ্রেস দিয়ে তৈরী ওয়েবসাইট ব্যবহারকারীদের কে সাইটের কনটেন্ট প্রদর্শন এবং বৈচিত্র প্রদর্শনের জন্যে একসঙ্গে কাজ করে । WordPress এর জন্য তৈরী একেকটি থিম আমাদেরকে একেক রকমের বৈচিত্র্য প্রদর্শন করে এবং ব্যবহারকারীর সামনে সাইটের কনটেন্ট গুলো দেখতে কেমন হবে তা নির্ধারণ করে ।

মূলতঃ থিম গুলোই ব্যবহারকারীকে wordpress এর বৈচিত্রকে প্রদর্শন করে । যেমন: একটি থিম দিয়ে ওয়ার্ডপ্রেস CMS কে সাধারণ একটা ব্লগ এ যেমন রূপান্তর করতে পারেন । একই সাথে আপনি চাইলে একটি থিম দিয়ে WordPress কে শপিং সাইট, পোর্টফোলিও সাইট, মেগাজিন সাইট সহ আরো অনেক ধরনের ওয়েব সাইটে রূপান্তর করতে পারেন । প্রকৃতপক্ষে ওয়ার্ডপ্রেস থিমগুলি ডিজাইনারদের সৃজনশীলতার সঙ্গে, ডেভেলপারদের, এবং সাধারণ ব্যবহারকারীদের একত্রিত করে ।

What is Template?

WordPress এর সমগ্র Theme জুড়ে ব্যবহৃত File গুলিকে বলা হয় Template বা Template File. এবং এই ফাইল গুলো আপনার সাইট কিভাবে প্রদর্শিত হবে তা নির্ধারণ করে। আর Template File গুলি মূলতঃ সব PHP File যা HTML, টেম্পলেট এর Tags এবং PHP কোড এর সমন্বয়ে হয়ে থাকে। আপনি যখন আপনার থিম তৈরি করবেন। তখন আপনি আপনার ওয়েবসাইটের বিভিন্ন অংশের লেআউট এবং ডিজাইন করার জন্য টেমপ্লেট ফাইল গুলো ব্যবহার করতে হবে। উদাহরণস্বরূপ, আপনি header তৈরি করার জন্য header.php টেমপ্লেট ব্যবহার করতে হবে, অথবা আপনার ওয়েব সাইট এ comment সিস্টেম যুক্ত করার জন্য comments.php template file ব্যবহার করতে হবে।



Wrodpress Theme

WordPress এর সবগুলো থিমই ওয়ার্ডপ্রেস এর /wp-content/Themes/ ডিরেক্টরি তে থাকে। উদাহরণস্বরূপ, আপনার একটি থিম “satech360” নামে আছে , তাহলে সেটি এবং তার সমস্ত ফাইল যেমন Template Files, stylesheets , JavaScript Files, image, fonts ইত্যাদি আপনি পাবেন wp-content / themes / satech360/ ফোল্ডারে।

WordPress Theme Development

WordPress এর Theme এ কি কি Template File থাকে ?

1. index.php

মূল template file, এটি সব theme এর জন্য বাধ্যতামূলক।

2. style.css

প্রধান stylesheet File, এটি সব theme এর জন্য বাধ্যতামূলক, এটিতে আপনার থিম এর Design ছাড়াও Theme এর header information যেমন : Theme name, author name, Theme Version, Date এবং Description সহ আরো অনেক তথ্য থাকে।

3. rtl.css

ওয়েব সাইট এর কনটেন্ট যদি right to left (ডান থেকে বামে) প্রদর্শনের দরকার হয় , তাহলে এটি ব্যবহৃত হয়।

WordPress Theme Development

comments.php

এটি ওয়েব সাইট এর comment section এর layout design এর জন্য ব্যবহৃত করতে পারেন।

front-page.php

ওয়েবসাইটের Front Page এর জন্য ব্যবহৃত হয় যদি এটি বিদ্যমান থাকে, এবং সেটি **Admin > Settings > Readingz** এ গিয়ে সেট করতে হয়।

home.php

home page হচ্ছে WordPress এর default front page. আপনি যদি কোনো static front page কে home page হিসেবে সেট না করেন , তাহলে এটি বাই ডিফল্ট latest post গুলো দেখাবে।

WordPress Theme Development

header.php

header template file সাধারণত আপনার সাইটের document type, meta information, stylesheet ,javascript সহ অন্যান্য তথ্য সংযুক্ত থাকে।

singular.php

wordpress যখন single.php file অথবা page.php file টি পায়না তখন singular template ব্যবহৃত হয়। আর যদি singular.php file টিও পাওয়া না যায় তখন index.php file টি ব্যবহৃত হয়।

single.php

single post template ব্যবহার করা হয় যখন একজন ভিজিটর single post এর জন্য request করেন।

WordPress Theme Development

single-{post-type}.php

যখন ভিসিটর একটা কাস্টম পোস্ট টাইপ থেকে একটা single post কে রিকোয়েস্ট করে। উদাহরণস্বরূপ single-book.php template file টি single post এর জন্য ব্যবহৃত হবে যখন বুক book নামে কোনো custom টেমপ্লেট ফাইল থাকবেনা।

archive-{post-type}.php

Archive Post Type টেমপ্লেটটি ব্যবহার করা হয় যখন visitor একটি custom post type আর্কাইভের অনুরোধ করেন।

উদাহরণস্বরূপ, archive-book.php template file টি আর্কাইভ পোস্ট গুলো দেখানোর জন্য ব্যবহার করা হবে যখন books নামে একটি কাস্টম পোস্টের জন্য। archive.php টেমপ্লেট ফাইলটি ব্যবহার করা হবে , যখন archive-post-type.php template file টি থাকবেনা।

page.php

যখন ভিজিটর কোনো ইন্ডিভিজুয়াল পেজ কে রিকোয়েস্ট করে তখন page template file টি ব্যবহৃত হয়। যা বিল্টইন টেমপ্লেট।

WordPress Theme Development

page-{slug}.php

ভিজিটর যখন একটি নির্দিষ্ট পেজ কে রিকোয়েস্ট করে যেমন “about” তখন slug page template request কৃত পেজ এর বিপরীতে একটা নির্দিষ্ট page template প্রদর্শন করবে। যেমন /about পেজ পরিবর্তে page-about.php ফাইল।

category.php

যখন ভিজিটর ক্যাটাগরির ভিত্তিতে post কে request করে , তখন category template ব্যবহৃত হয়।

tag.php

যখন ভিজিটর ট্যাগের ভিত্তিতে post কে request করে , তখন category template ব্যবহৃত হয়।



taxonomy.php

যখন ভিজিটর একটি কাস্টম taxonomy term ভিত্তিতে post কে request করে , তখন taxonomy template ব্যবহৃত হয়।

author.php

যখন ভিজিটর একটি author page কে load করে , তখন author page template ব্যবহৃত হয়।

date.php

যখন ভিজিটর একটি post কে date অনুযায়ী request করে , তখন date/time template ব্যবহৃত হয়।

archive.php

যখন ভিজিটর একটি post কে category, author অথবা date অনুযায়ী request করে , তখন archive template ব্যবহৃত হয়।

উল্লেখ্য: এই টেম্পলেট টি বর্তমানে ব্যবহৃত template file যেমন-category.php, author.php এবং date.php file এর পরিবর্তে ব্যবহৃত হয়।



search.php

ভিজিটর এর search result দেখানোর জন্য search result template টি ব্যবহৃত হয়।

attachment.php

ভিজিটর কে attachment file যেমন image, pdf অথবা অন্যান্য media file দেখানোর জন্য attachment template টি ব্যবহৃত হয়।

image.php


ভিজিটর কে attachment file যেমন image কে দেখানোর জন্য image attachment template টি ব্যবহৃত হয়। তবে image attachment file টি না থাকলে শুধু attachment.php file দিয়ে দেখানো হয়।

404.Php

যখন wordpress কোনো post খুঁজে না পায় তখন 404 template টি ব্যবহৃত হয়।

WordPress এ Template Tags কি?

WordPress এ Template Tags হচ্ছে কিছু PHP Function যা ওয়ার্ডপ্রেস এর থিম এর মধ্যে বিভিন্ন ইনফরমেশন কে ডাইনামিক্যালি গ্রহণ এবং প্রদর্শনের জন্য ব্যবহৃত হয়। WordPress এর থিম ডেভেলপাররা Theme এর Template File গুলোর মধ্যে ডাইনামিক ইনফরমেশন গ্রহণ এবং প্রদর্শনের জন্য Template tags গুলো ব্যবহার করে থাকেন। ওয়ার্ডপ্রেস এর থিম ব্যবহারের জন্য অসংখ্য বিল্ট-ইন template tags রয়েছে। আবার ওয়ার্ডপ্রেস থিম এবং প্লাগিন গুলোও তাদের নিজস্ব template tags তৈরী এবং ব্যবহার করে থাকেন।



`<?php the_author(); ?>`

author template tag এটি মূলতঃ ওয়ার্ডপ্রেস পোস্ট এ সেই পোস্টের অথোরের নাম প্রদর্শনের জন্য ব্যবহৃত হয়।

`<p>This post is written by <?php the_author(); ?></p>`

wordpress theme এর template file গুলোতে বহুল ব্যবহৃত template tags গুলো হচ্ছে নিম্নরূপ:

get_header()- Themeএর index এ header.php file কে include করার জন্য ব্যবহৃত হয়।


get_sidebar()-Themeএর index এ sidebar.php file কে include করার জন্য ব্যবহৃত হয়।

get_footer()- Themeএর index এ footer.php file কে include করার জন্য ব্যবহৃত হয়।

WordPress Theme Development

wordpress theme development এবং Theme Customization এর মধ্যে পার্থক্য কি?

WordPress এর Theme Development বলতে বুঝানো হয় নতুন করে WordPress Theme তৈরী করা। যেখানে Theme এর মধ্যে Template File গুলো সম্পূর্ণ নতুন অথবা পুরাতন কনসেপ্ট এর উপর ভিত্তি করে নতুন ভাবে গড়ে উঠবে। আর Theme Customization বলতে বুঝানো হয় existing বা পুরাতন যেকোনো theme এর feature বা সুবিধা গুলো অক্ষুন্ন রেখে তার template file গুলোর ডিজাইন পরিবর্তন করা এবং নতুন সুবিধা (Feature) যুক্ত করা।



Sahab Uddin
Phone: 01867033550
Email: sahabuddinriyaj984@gmail.com
Web: satech360.com



THANK
YOU