1. What is HTML?

- HTML is used to define the structure and layout of a webpage. It consists of various elements that define headings, paragraphs, link, images, and more.

HTML是一种标记语言，用于创建网页的结构和布局。它由各种元素组成，用于定义标题、段落、链接、图像等。

- HTML is the standard markup language for creating web pages. It provides a set of rule(markup) for how content should be structured on a webpage.

HTML是创建网页的标准标记语言，提供一组规则（标记），规定了内容在网页上应该如何结构化。

- HTML use elements help the browser understand how to render and display the content to user.

HTML中的元素需要浏览器理解如何呈现和显示内容给用户。

2. Web Browser is used to read a HTML page and render.

-A web browser is a software application that is used to access and display content on the World Wide Web. It interprets HTML code, which is the standard markup language used to create web pages. The browser fetches the HTML file from web servers and renders them into a visually understandable format, which includes displaying text, image, link, and other media elements. In addition to HTML, Web browsers can interpret other language like CSS, for styling and JavaScript for interactivity.

Web浏览器是一种软件，用于访问和显示网页内容。它解释HTML代码，从服务器获取网页文件，并以可视化格式呈现，包括文本、图像、链接等。此外，浏览器还能处理CSS（样式）和JavaScript（交互）等语言。

3.HTML Heading

- HTML heading are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.

<h1> 定义最大字体(重要), 而 <h6> 定义最小(不重要的标题) 。

Heading 1 - H1
Heading 2 - H2
Heading 3 - H3
Heading 4 - H4
Heading 5 - H5
Heading 6 - H6

4. Table element styling

- Alignment is a valid attribute associated with HTML table elements. It determines how the content within the cells of a table is aligned (e.g. left,right, center).

对齐（位置）是与HTML表格元素对应的有效属性。它确定表格单元格中内容的对齐位置，例如左对齐、右对齐、居中对齐。

```html
<table>
  <tr class="table_1">
    <th>Header 1</th>
    <th>Header 2</th>
    <th>Header 3</th>
  </tr>
  <tr class="table_2">
    <td>Row 1, Cell 1</td>
    <td>Row 1, Cell 2</td>
    <td>Row 1, Cell 3</td>
  </tr>
  <tr class="table_3">
    <td>Row 2, Cell 1</td>
    <td>Row 2, Cell 2</td>
    <td>Row 2, Cell 3</td>
  </tr>
</table>
```

```css
.table_1{
    text-align: left;
}
.table_2{
    text-align: center;
}
.table_3{
    text-align: right;
}
```

Header 1	Header 2	Header 3
Row 1, Cell 1	Row 1, Cell 2	Row 1, Cell 3
Row 2, Cell 1	Row 2, Cell 2	Row 2, Cell 3

- While color can be applied to the content within a table using CSS, it is not an inherent attribute of HTML table elements. HTML itself does not have a specific attribute for setting colors in tables.
虽然可以使用CSS为表格中的内容添加颜色，但颜色并不是HTML表格元素的固有属性。HTML本身没有特定的属性用于在表格中设置颜色。

- Spanning it allows a cell to span multiple rows or columns, which can be useful for creating complex table layouts.
跨展 允许单元格跨越多行或多列，这对于创建复杂的表格布局非常有用。

```
<tr class="table_1">
  <th rowspan="2">Header 1</th>
  <th>Header 2</th>
  <th colspan="2">Header 3</th>
</tr>
```

rowspan(上下合体)	Header 2	colspan(左右合体)	
	Row 1, Cell 1	Row 1, Cell 2	Row 1, Cell 3
Row 2, Cell 1	Row 2, Cell 2	Row 2, Cell 3	Row 2, Cell 4

- Size can refer to the dimensions of the table itself or the size of individual cells or columns.
"Size" 可以指的是表格本身的尺寸，也可以指单个单元格或列的尺寸。

```css
<style>
  table {
    width: 100%; /* Set the width of the table to 100% of its container */
  }

  td {
    width: 50px; /* Set the width of individual cells */
  }

  th {
    height: 30px; /* Set the height of table headers */
  }
</style>
```

5. HTML Image Syntax.(照片格式)

- The HTML <img> tag is used to embed an image in a web page.

"<img>" 显示在 HTML 图像的格式 。

- Image are not technically inserted into a web page; Image are linked to web pages. The <img> tag creates a holding space for the referenced image.

从技术上讲，图像并非直接插入到网页中；图像与网页相链接。

<img> 标签创建了一个用于显示引用图像的空间。

- The <img> tag is empty, it contains attributes only, and does not have a closing tag.

标签是空的，只包含属性，不用关 。

- The <img> tag has two required attributes.

<img> 标签有两个必需的属性

i. Src - Specifies the path to the image.

指定图像的路径

ii. Alt - Specifies an alternate text for the image.

图像显示不出来的时候，显示照片的字体/(你所写的字)。

iii. Syntax - > <img src="*url*" alt="*alternatetext*">(格式·)

6. HTML <form> Elements `

- The <input> Elements

One of the most used form elements is the <input> elements.

The <input> elements can be displayed in several ways, depending on the type attribute.

<input> 是HTML 其中的一个元素。

拥有不同的 'type'。

-The <label> Element

The <label> element defines a label for several form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

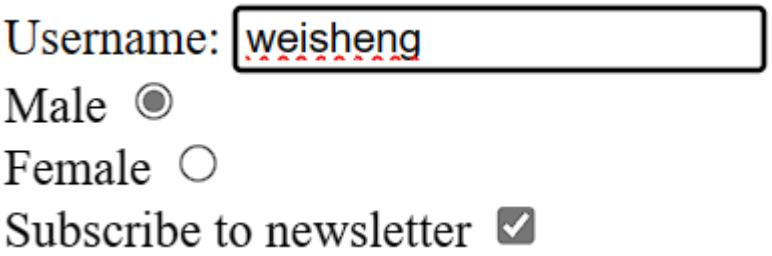
The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

'<label>' 是标题，方便阅读。在'<radio>'，'<checkbox>'的情况下

'<label>' 可以利用 'for' 和 'id'，它会切换单选按钮或复选框。

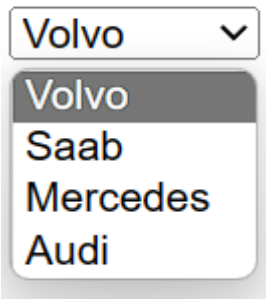
```
<!-- 示例：使用label标签关联文本输入框 -->
<label for="username">Username:</label>
<input type="text" id="username" name="username">

<!-- 示例：使用label标签关联单选框 -->
<label for="male">Male</label>
<input type="radio" id="male" name="gender" value="male">

<label for="female">Female</label>
<input type="radio" id="female" name="gender" value="female">

<!-- 示例：使用label标签关联复选框 -->
<label for="subscribe">Subscribe to newsletter</label>
<input type="checkbox" id="subscribe" name="subscribe" checked>
```



- The <select> Element
The <select> element defines a drop-down list:
The <option> element defines an option that can be selected.
By default, the first item in the drop-down list is selected.
To define a pre-selected option, add the selected attribute to the option:
Visible Values:
Use the size attribute to specify the number of visible values:
Allow Multiple Selections:
Use the multiple attribute to allow the user to select more than one value:

'<select>' 是元素用于 dropdown-list 。'<option>' 是元素用于选项（option）。他们是一个选项元素组合。

```
<!-- 示例：简单的下拉列表 -->
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```



使用 size 属性来指定下拉列表中可见的选项数量，使用 multiple 属性来允许多选。

```
<!-- 示例：指定可见值的数量 -->
<select name="colors" size="3">
  <option value="red">Red</option>
  <option value="green">Green</option>
  <option value="blue">Blue</option>
</select>

<!-- 示例：允许多选 -->
<select name="fruits" multiple>
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="orange">Orange</option>
</select>
```

- The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

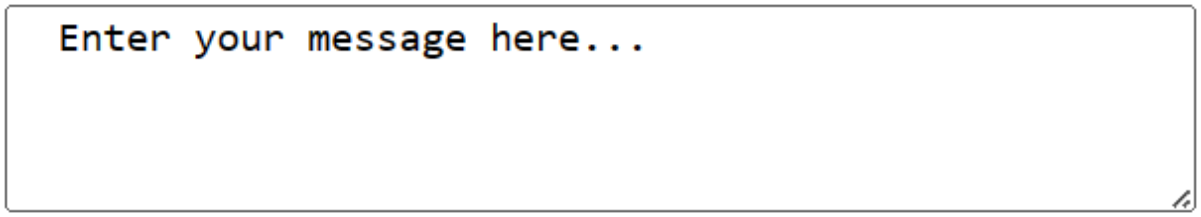
'<textarea>' 是元素用于定义多行输入字段和 'text' 会有点相似。

'rows' 属性指定文本区域中可见的行数。

'cols' 属性指定文本区域的可见宽度。

用户可以根据自己的字数来自定义'rows'和'cols'的长度与宽度。

```
<!-- 示例：简单的文本区域 -->
<textarea name="message" rows="4" cols="50">
  Enter your message here...
</textarea>
```



7. HTML List

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

```
Description List

<dl>
  <dt>Term 1</dt>
  <dd>Description for Term 1</dd>

  <dt>Term 2</dt>
  <dd>Description for Term 2</dd>

  <dt>Term 3</dt>
  <dd>Description for Term 3</dd>
</dl>
```

Unorderlisted

- Item 1
- Item 2
- Item 3

Orderlisted

1. Item 1
2. Item 2
3. Item 3

Description List

Term 1
: Description for Term 1

Term 2
: Description for Term 2

Term 3
: Description for Term 3

8. HTML Link

-HTML links are hyperlinks

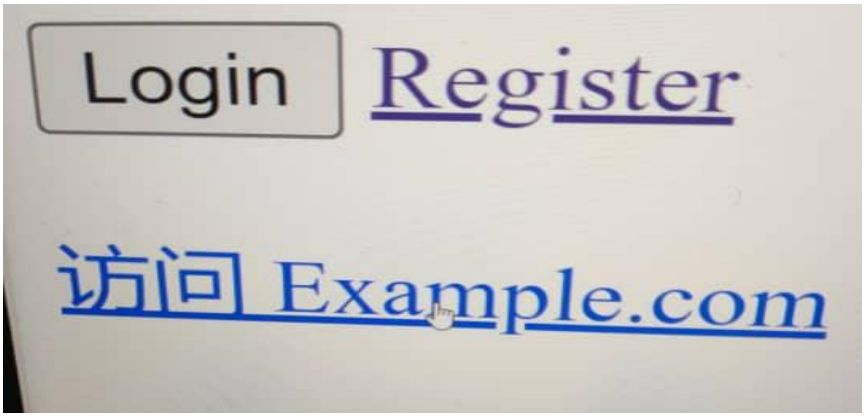
-You can click on a link and jump to another document.

'link' 链接用于创建'connect'，允许user单击并跳转到另一个文档(file)。

```
<a href="https://www.example.com">访问 Example.com</a>
```

-When you move the mouse over a link, the mouse arrow will turn into a little hand.

当将鼠标悬停在链接上时，鼠标会变成小手的形状，表示链接是可点击的。



HTML Links- Syntax.

```
<a href="url">link text</a>
```

-The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

<a> 元素最重要的属性是 href，它链接指定的目标(URL)。

-The link text is the part that will be visible to the reader.
-Clicking on the link text, will send the reader to the specified URL address.
单击链接 'link' 会将 点的人（读者）发送到指定的URL地址。

9. Bootstrap

- Bootstrap is a widely used front-end CSS framework that provides a set of pre-built, responsive design components and styles. It helps developers create websites that adapt and look good on various devices and screen sizes, from desktops to mobile devices.
'Bootstrap' 是一个大众使用的前端CSS框架，提供一组提前做好的响应式设计HTML和CSS。它帮助我们创建能够在各种设备（电脑，手机）和屏幕尺寸上适应并呈现良好的网站，从桌面到移动设备。

- Bootstrap simplifies the process of creating responsive and visually appealing web pages by providing a collection of pre-designed elements and a flexible grid system. This makes it a popular choice for web developers.
'Bootstrap' 通过提供一组预设计的元素和灵活的网格系统简化了创建响应式和视觉上吸引人的网页的过程。这使得它成为Web开发人员的流行选择。

-Between container-flex and container-grid in Bootstrap. Container-flex is designed for creating one-dimensional layouts (like rows or columns), while container-grid is used for two-dimensional layouts where you need to manage both rows and columns
`container-flex` 在创建 一维（1D）布局。它使用弹性盒子布局（Flexbox），这使得元素只可以在一条线上的空间排列。
`container-grid` 则用于 二维（2D）布局，其中需要同时管理行和列。它使用 CSS Grid 布局，允许在行和列的点上精确放置元素。

10. What is JavaScript?

- JavaScript is a scripting language that allows you to add interactivity to websites. It runs in the browser and enables you to manipulate the Document Object Model(DOM), handle events, and perform various operations to create dynamic and interactive web pages.

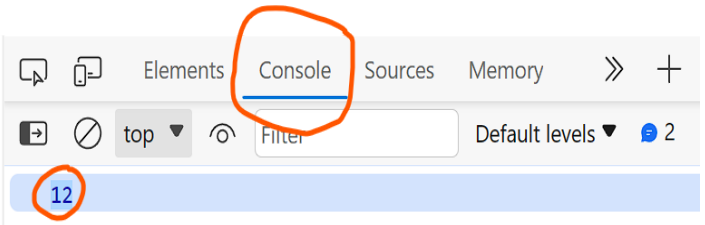
JavaScript (JS) 是一种脚本语言，它允许您为网站添加交互性。它运行在浏览器中，使您能够操作文档对象模型（DOM）、处理事件，并执行各种操作，从而创建动态和交互式的网页。

11. JavaScript Variable.

- Var = Var is function-scoped, meaning it is visible throughout the entirefunction it is declared in, regardless of block statement.

'Var' 不受块语句的限制,可以重复,但拿取的数值是最后的 (12) 。

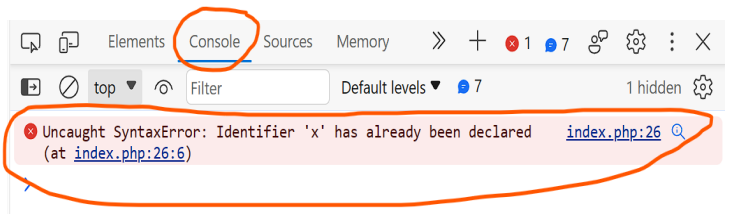
```
<script>

 var x = 10;
 var x = 11;
 var x = 12;

  console.log(x);

</script>
```



-Let = let is block-scoped, meaning it is only visible within the block(enclosed by curly braces) it is defined in.

'Let' 和Var 就不一样了,不能重复Let。

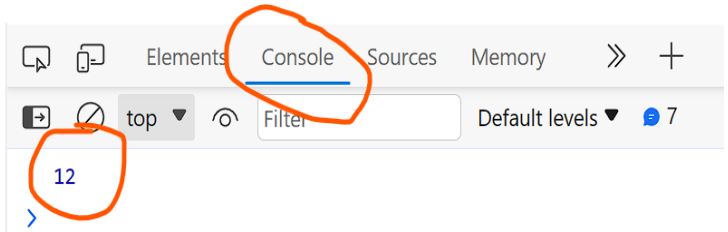
```
<script>

let x = 11;
let x = 12;

 console.log(x);

</script>
```



以上的是错误示范❌

当你要更改里面的数值，以下这才是Let正确用法。11变去12 。

```
<script>

let x = 11;
    x = 12;

 console.log(x);

</script>
```

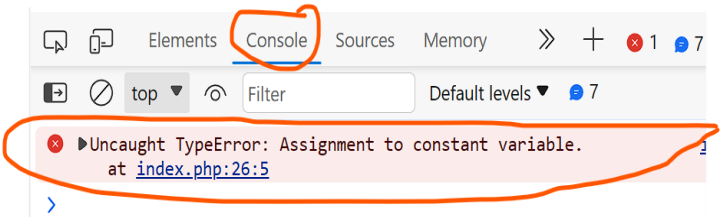


- Const = Const is also block-scoped and is used for declaring constants, which
means its value cannot be reassingned once it is set.

'Const' 和Let 就很相似但是 'Const' 不能更改里面的数值。

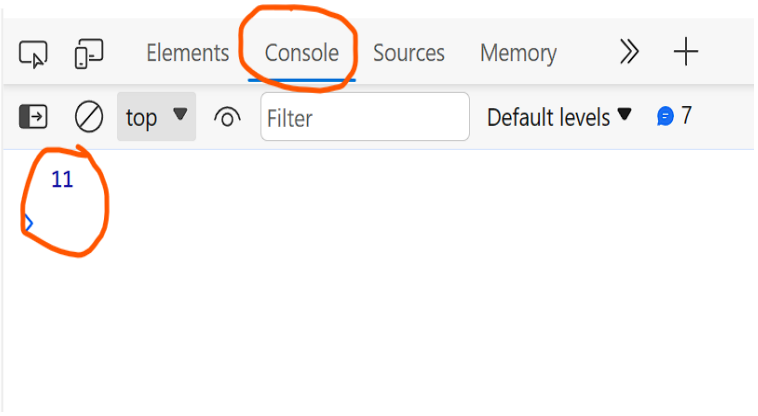
以下是个错误的示范❌

```
<script>

const x = 11;
    x = 12;

 console.log(x);

</script>
```



以下才是对的示范 。

```
<script>
 const x = 11;
  console.log(x);
</script>
```



12. JavaScript Comparison Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

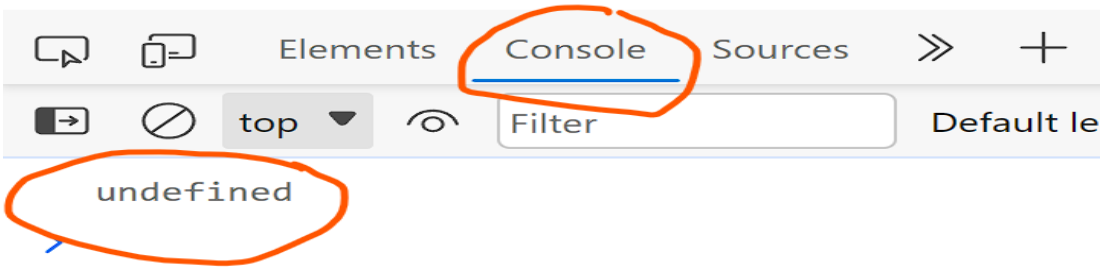
13. JavaScript Values

-Undefined

- If a variable has been declared, but its value has not been set, it will have the value 'undefined'.
- It is a type in JS, and it means that a variable has been declared but hasn't been assigned a value yet.
- It is often used to indicate that a variable or property exists but has not been given a value yet.
- It is also returned by functions when they do not have a 'return' statement.

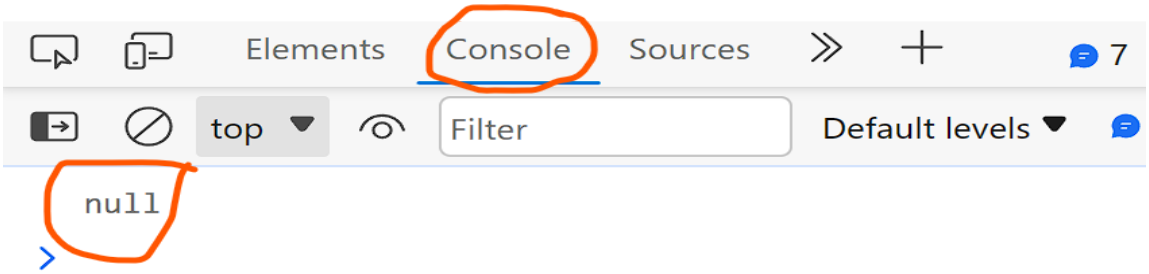
如果一个 'Variable' 已经声明，但 'value' 未设置，它将具有值 'undefined'。它是 JavaScript 中的一种类型,变量已经声明但尚未被赋值(undefined)。

```
let variable; // variable is undefined
function doSomething() {
  // No return statement, so it returns undefined
}
```



-Null
- “null” is a deliberate assignment value that indicates a variable or object intentionally does not have a meaningful value.
- It is typically assigned to a variable by a programmer to signify that it is intentionally empty or has no value.
- It is a type in JS and can be used in logical operations.
"null" 是一种有意赋值的值，表示变量或对象故意没有有意义的值。通常由程序员分配给一个变量，表示其故意为空或没有值。

```
let emptyValue = null; // variable is intentionally set
```



14. PHP String

- ` . ` This is the concatenation operator in PHP. It is used to join two strings or a string and a variable together.
` . ` 这个运算符可以用来连接两个字符串。

```php
$str1 = "Hello";
$str2 = "World";

// 使用 . 运算符连接两个字符串
$result = $str1 . $str2;
echo $result; // 输出：HelloWorld
```

$str1 . $str2 中间的那个点运算了两个字符串合了一体。

HelloWorld

15. PHP SESSION and COOKIE

COOKIE:-

1. Storage: Cookies are stored on the client's browser.
存储：Cookie 存储在客户端的浏览器中

2. Purpose: They are primarily used for client-side data storage.
目的：主要存储客户端的数据。

3. Size Limit: Cookies have a size limit (usually around 4KB per cookie).
大小限制：Cookie 有大小限制（通常每个 cookie 大约为 4KB）。

4. Expiration: Cookies can have an expiration date, after which they will be automatically deleted
过期：Cookie 可以设置过期日期，在此日期之后将被自动删除。

5. Security: They can be manipulated or read by the client, which can potentially lead to security concerns. You can set flags like Secure and Http Only to enhance security.

安全性：因为可以被客户端读取或控制，这可能导致安全问题。您可以设置标志如 **Secure** 和 **HttpOnly** 来提高安全性。

6. Usage: Cookies are commonly used for tasks like remembering user preferences, shopping cart items, tracking user behavior, and session management.
用途：**Cookie** 常用于记住用户首选项、购物车项目、跟踪用户行为和会话管理等任务。

SESSION:-
1. Storage: Sessions are stored on the server.
存储：会话存储在服务器上。

2. Purpose: They are used for server-side data storage.
目的：用于存储服务器端的数据。

3. Size Limit: Sessions can store larger amounts of data as they are stored on the server.
大小限制：由于存储在服务器上，所以会话可以存储更大量的数据。

4. Expiration: Sessions usually expire when the user closes the browser or after a certain period of inactivity (which can be configured).
过期：会话通常在用户关闭浏览器或一定时间的不活动后（可配置）过期。

5. Security: Data stored in sessions is not accessible or modifiable by the client, providing a higher level of security compared to cookies.
安全性：存储在会话中的数据不可被客户端访问或修改，提供了比 **Cookie** 更高的安全性。

6. Usage: Sessions are commonly used for tasks like user authentication, storing sensitive data, and maintaining user-specific information across multiple pages.
用途：会话常用于用户身份验证、存储敏感数据和在多个页面之间维护用户特定信息等任务。

Comparison:（比较）
● Cookies are more suitable for non-sensitive data and can be used for tasks like personalization and tracking.
● Sessions are better for storing sensitive information, managing user authentication, and maintaining application state.

- **Cookies** 更适合存储非敏感数据，可用于个性化和跟踪等任务。
- 会话 更适合存储敏感信息，管理用户身份验证和维护应用程序状态。

16. SQL CRUD

-CRUD stands for Create, Read, Update, and Delete. It represents the four basic operations that can be performed on data in a database or data storage system:

CRUD 代表创建(Create),读取(Read),更新(Update)和删除(Delete)。它对数据库或数据存储系统中的数据执行的四种基本操作

1. Create (C): This operation involves adding new data records or entries into the database. It is used to insert new information.

插入新信息，新信息添加到数据库中。

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. Read (R): This operation involves retrieving or querying existing data from the database. It is used to view or fetch information without altering it.

数据库中检索或查询现有数据，查看或获取信息。

```
SELECT * FROM users WHERE age > 30;
```

3. Update (U): This operation involves modifying existing data in the database. It is used to change or update information.

修改数据库中的现有数据，更改或更新信息。

```
UPDATE users SET email = 'new.email@example.com' WHERE id = 1;
```

4. Delete (D): This operation involves removing data records or entries from the database. It is used to permanently delete information

数据库中删除数据记录或信息，永久删除信息。

```
DELETE FROM users WHERE id = 2;
```

17. SQL JOIN

The JOIN keyword in MySQL is used to combine rows from two or more tables based on a related column between them. The purpose of using JOIN is to establish relationships between tables and retrieve data that is spread across different tables in a database.

在 **MySQL** 中，**JOIN** 用在两个或更多表之间变成一起 。它允许在单个查询中检索来自多个表的信息。使用 **JOIN** 的目的是建立表之间的关系并检索分布在数据库中不同表中的数据。

在 **MySQL** 中有几种类型的 **JOIN**：

1）INNER JOIN：This is the most common type of join. It returns only the records that have matching values in both tables.

这是最常见的 JOIN 类型。它仅返回两个表中具有匹配值的记录。

Example:

```
- SELECT *
- FROM table1
- INNER JOIN table2 ON table1.column_name =
  table2.column_name;
```

2）LEFT JOIN (or LEFT OUTER JOIN): This join returns all the records from the left table, and the matched records from the right table. The result will contain all records from the left table and the matching records from the right table. If there is no match, NULL values are returned from the right side.

此 JOIN 左表的所有记录和右表的匹配记录。如果没有匹配，从右侧返回 NULL 值。

Example:

```
- SELECT *
- FROM table1
- LEFT JOIN table2 ON table1.column_name =
  table2.column_name;
```

3）RIGHT JOIN (or RIGHT OUTER JOIN): This join is the opposite of the LEFT JOIN. It returns all records from the right table and the matched records from the left table. If there is no match, NULL values are returned from the left side.

这个JOIN 呢是 LEFT JOIN 的相反。它返回右表的所有记录和左表的匹配记录。如果没有匹配，从左侧返回 NULL 值。

Example:

```
- SELECT *
- FROM table1
- RIGHT JOIN table2 ON table1.column_name =
  table2.column_name;
```

4）FULL JOIN (or FULL OUTER JOIN): This join returns all records when there is a match in one of the tables. It returns NULL values for columns that do not have a match.

此 JOIN 在其中一个表中有匹配时返回所有记录。对于没有匹配的列，返回 NULL 值。

Example:

```
- SELECT *
- FROM table1
- FULL JOIN table2 ON table1.column_name =
  table2.column_name;
```

5）CROSS JOIN: This join returns the Cartesian product of both tables, which means it combines every row from the first table with every row from the second table.

此 JOIN 返回两个表的 'Cartesian product'，即它将第一个表的每一行与第二个表的每一行组合

Example:

```
- SELECT *
- FROM table1
- CROSS JOIN table2;
```

18. Mysql Primary Key and Foreign key

In MySQL tables, a primary key and a foreign key are both types of constraints that help establish relationships between different tables. They play a crucial role in maintaining data integrity and enforcing referential integrity within a database.

在 MySQL 表中，' Primary Key ' 和 ' Foreign key '都是的一种特定（重要性），帮助在不同表之间建立关系。维持数据完整性和强制引用完整性方面起着至关重要的作用。

Primary Key:

A primary key is a column (or a set of columns) in a table that uniquely identifies each record (or row) in that table.It must have a unique value for each record, meaning no two records can have the same value in the primary key column.A primary key column cannot contain NULL values.Only one primary key can be defined for each table.It is used to enforce entity integrity, ensuring that each record in a table is uniquely identifiable.

' Primary Key '是表中的 ' column ' (或一组' column ')，它唯一标识表中的每条记录（或行）。它必须对于每条记录具有唯一的值，这意味着' Primary Key ' 列中不能有两条记录具有相同的值。此外，' Primary Key ' 不能包含 NULL 值。确保表中的每条记录都是唯一可识别的。

Example:

```
-  CREATE TABLE students (
-      student_id INT PRIMARY KEY,
-      first_name VARCHAR(50),
-      last_name VARCHAR(50)
-  );
```

以上的 ' student_id ' 就是唯一性的（Primary Key）。

Foreign Key:

A foreign key is a column (or a set of columns) in a table that refers to the primary key in another table. It establishes a relationship between two tables. The foreign key column contains values that match the values in the primary key column of the referenced table. It allows you to perform operations like joins to retrieve related data from multiple tables.

' Foreign Key '是表中的 ' column ' (或一组' column ')，它引用另一表中的 ' Primary Key ' 。它建立了两个表之间的关系。' Foreign Key ' 列包含与引用表的' Primary Key '列中的值匹配的值。它允许执行操作，如连接，以从多个表中检索相关数据。

Example:

```
- CREATE TABLE orders (
-     order_id INT PRIMARY KEY,
-     customer_id INT,
-     order_date DATE,
-     FOREIGN KEY (customer_id) REFERENCES
  customers(customer_id)
- );
```

customer_id 是记录表中的' Foreign Key '，将 **customers** 表中的 **customer_id** 做为关系。

总结来说' Primary Key '是唯一性，' Foreign Key '是建立关系。

19. PHP Loop（循环）

Loop 支持多种循环语句，可以重复执行。例如：

1）for：This loop is used when you know in advance how many times you want to execute a block of code. It has a defined initialization, condition, and increment.

当您预先知道要执行一块代码多少次时可以使用此循环。它具有定义的初始化、条件和递增。

Example:

```
for ($i = 0; $i < 5; $i++) {
    echo $i;
}
```

01234

2) While: This loop is used when you want to execute a block of code as long as a specified condition is true.

当您希望在指定条件的情况下执行一块代码时使用此循环。

Example:

```
$i = 0;
while ($i < 5) {
    echo $i;
    $i++;
```

01234

3）do-while : Similar to the while loop, but it executes the block of code at least once, and then repeats it as long as a specified condition is true.

类似于 **while** 循环，但它至少执行一次代码块 。

Example:

```
$i = 0;
do {
    echo $i;
    $i++;
} while ($i < 5);
```

01234

4) foreach : This loop is used exclusively for arrays. It iterates over each key-value pair in an array.

专门用于数组（组合）。

Example:

```
$colors = array("red", "green", "blue");
foreach ($colors as $color) {
    echo $color;
}
```

red green blue

5）foreach (associative arrays): It can also be used to iterate over associative arrays, providing both the key and the value.

关联数组（例如：人物与名字）。

Example:

```
$person = array("first_name" => "John", "last_name" =>
"Doe");
foreach ($person as $key => $value) {
    echo "$key: $value";
}
```

first_name: John
last_name : Doe

6）break and continue: While not traditional looping constructs, break and continue are used within loops to alter the flow of execution. break terminates the loop early, while continue skips the rest of the loop and continues with the next iteration.

虽然不是传统的循环结构，但 break 和 continue 在循环内部用于更改执行流。break 提前stop循环，而 continue 跳过循环的其余部分并继续下一次语句。

Example:

```
for ($i = 0; $i < 10; $i++) {
    if ($i == 5) {
        break; // exit the loop when i is 5
    }
    if ($i % 2 == 0) {
        continue; // skip even numbers
    }
    echo $i;
}
```

1

3

以上的代码循环条件是 $i 等于 0，然后$i 小过 10 的话，正常来说会循环1，2，3，4，5，6，7，8，9，但是这边执行了 $i == 5 的时候 break 了。导致只会执行1，2，3，4。然后下面多了个条件continue就是$i % 2 == 0 导致只剩下1，3。%的意思是余数。