

HTML

HTML (Hypertext Markup Language) 是一种标记语言，用于创建网页的结构和内容。它由一系列标签 (tags) 组成，每个标签都有特定的含义和功能。下面是一些 HTML 基础标签及其用法的示例：

1. ``<html>``：定义 HTML 文档的根元素。

```
``html
<html>
    <!-- HTML document content goes here -->
</html>
``
```

2. ``<head>``：定义 HTML 文档的头部，包含了元数据和链接到外部资源的信息。

```
``html
<head>
    <title>Page Title</title>
    <meta charset="UTF-8">
    <!-- Other meta tags, link tags, etc. -->
</head>
``
```

3. ``<title>``：定义网页的标题。

```
``html
<title>My Website</title>
``
```

4. ``<body>``：定义 HTML 文档的主体部分，包含可见的内容。

```
``html
<body>
    <h1>Hello, World!</h1>
    <p>This is a paragraph.</p>
    <!-- Other content -->
</body>
``
```

5. ``<h1>``, ``<h2>``, ``<h3>``, ``<h4>``, ``<h5>``, ``<h6>``：定义标题，级别从 1 到 6，1 是最高级别。

```
``html
<h1>Main Heading</h1>
<h2>Subheading</h2>
``
```

6. `

`: 定义段落。

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

7. `
`: 定义换行。

```
```html
<p>This is the first line.
This is the second line.</p>
```
```

8. ``: 定义超链接。

```
```html
Visit Example
```
```

9. ``: 定义图像。

```
```html

```
```

10. `

`, `

`, ` - `: 定义无序列表和有序列表。

```
```html

 Item 1
 Item 2

 First Item
 Second Item

```
```

这些是 HTML 的基础标签，通过它们可以创建简单到复杂的网页结构和内容。

`<td>` 是 HTML 中用于定义表格中的单元格（**table data**）的标签。它通常与 `|`（表格行）和 `

以下是 ` ` 标签的基本用法示例： |

```
```html
<table border="1">
 <tr>
 <td>Row 1, Column 1</td>
 <td>Row 1, Column 2</td>
 </tr>
</table>
```
```

```

    </tr>
    <tr>
        <td>Row 2, Column 1</td>
        <td>Row 2, Column 2</td>
    </tr>
</table>
'''

```

在这个示例中，`<table>`定义了一个表格，`<tr>`定义了两个表格行，而`<td>`则定义了每个单元格的内容。在网页上呈现时，这将显示一个带有两行两列的简单表格，每个单元格中包含一些文本。

你还可以在`<td>`标签中包含其他 HTML 元素，例如段落、图片、链接等，以创建更复杂的表格布局。

```

'''html
<table border="1">
    <tr>
        <td>
            <h2>Heading</h2>
            <p>This is a paragraph inside a table cell.</p>
            <a href="#">Link</a>
        </td>
        <td>
            
        </td>
    </tr>
</table>
'''

```

这将在表格中的一个单元格中显示一个带有标题、段落、链接和图像的内容。

当然，HTML 中还有许多其他常用的标签和概念。以下是一些常见的 HTML 知识：

1. ****表单（Form）：** ** HTML 表单用于收集用户输入的数据。常见的表单元素包括输入框、下拉框、复选框、单选按钮等。

```

'''html
<form action="/submit-form" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username"><br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password"><br>
    <input type="submit" value="Submit">
</form>
'''

```

...

2. ****媒体元素（Media Elements）：** ** HTML 支持嵌入多媒体内容，如音频和视频。

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

<video controls>
 <source src="video.mp4" type="video/mp4">
 Your browser does not support the video element.
</video>
```
```

3. ****注释（Comments）：** ** 在 HTML 中，可以使用注释来向代码添加注释，以便开发者阅读代码或暂时隐藏代码片段。

```
```html
<!-- This is a comment -->
```
```

4. ****链接（Links）：** ** 使用`<a>`标签可以创建超链接，连接到其他页面或资源。

```
```html
Visit Example
```
```

5. ****图像（Images）：** ** 使用``标签可以在页面中嵌入图像。

```
```html

```
```

6. ****元信息（Meta Information）：** ** 使用`<meta>`标签可以定义 HTML 文档的元信息，如字符编码、作者、描述等。

```
```html
<meta charset="UTF-8">
<meta name="description" content="Description of the page">
```
```

7. ****链接样式表（Linking Stylesheets）：** ** 使用`<link>`标签可以将外部样式表链接到 HTML

文档。

```
``html
<link rel="stylesheet" href="styles.css">
``
```

8. **分区元素（Semantic Elements）**：HTML5 引入了许多语义化的标签，如`<header>`、`<footer>`、`<nav>`、`<article>`、`<section>`等，用于更好地组织页面内容。

```
``html
<header>
  <h1>Website Header</h1>
</header>
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
<section>
  <article>
    <h2>Article Title</h2>
    <p>Article content goes here.</p>
  </article>
</section>
<footer>
  <p>Copyright © 2024. All rights reserved.</p>
</footer>
``
```

这些是 HTML 中一些常用的知识点，可以帮助你构建丰富而功能强大的网页。

CSS

CSS 盒模型是用于描述网页中元素布局和设计的基本概念之一。它将每个 HTML 元素视为一个矩形盒子，由内容区域、内边距、边框和外边距组成。每个部分都可以通过 CSS 属性进行控制和调整。

以下是 CSS 盒模型中常用的属性及其解释，并附带示例：

1. **Width（宽度）和 Height（高度）**：

- `width`属性定义元素的宽度。
- `height`属性定义元素的高度。

```
```css
.box {
 width: 200px;
 height: 100px;
}
```
```

2. **Padding（内边距）：**

- `padding`属性定义元素内容区域与边框之间的空间。
- 可以分别使用`padding-top`、`padding-right`、`padding-bottom`和`padding-left`来指定上、右、下、左方向的内边距。

```
```css
.box {
 padding: 20px;
}
```
```

3. **Border（边框）：**

- `border`属性定义元素的边框。
- 可以指定边框的样式、宽度和颜色。

```
```css
.box {
 border: 2px solid #000;
}
```
```

4. **Margin（外边距）：**

- `margin`属性定义元素与其相邻元素之间的空间。
- 可以分别使用`margin-top`、`margin-right`、`margin-bottom`和`margin-left`来指定上、右、下、左方向的外边距。

```
```css
.box {
 margin: 10px;
}
```
```

综合示例：

```
``html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Box Model Example</title>
  <style>
    .box {
      width: 200px;
      height: 100px;
      padding: 20px;
      border: 2px solid #000;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="box">This is a box with padding, border, and margin.</div>
</body>
</html>
``
```

在这个例子中，我们定义了一个带有宽度、高度、内边距、边框和外边距的盒子。这些属性的值可以根据需要进行调整，以创建符合设计需求的布局。

JS

以下是关于 JavaScript 变量的四个陈述，以及它们的真假情况：

1. Variables with block scope are declared using the let keyword.

- True (正确)

- 在 JavaScript 中，使用`let`关键字声明的变量具有块级作用域，这意味着它们在定义它们的块（例如，if 语句、for 循环、函数等）内部可见，而在块外部不可见。

2. The keyword const is used to declare variables that you should not change after they are assigned.

- True (正确)

- `const`关键字用于声明常量，即变量的值在声明后不可更改。尝试重新分配`const`变量的值会导致错误。

3. A variable declared outside a function definition using the let keyword is a global variable.

- False (错误)

- 使用`let`关键字在函数外部声明的变量不是全局变量，而是具有全局作用域的变量。它们可以在函数外部访问，但不会被添加到全局对象（如`window`对象）中。

4. A variable declared outside a function definition using the var keyword is a global variable.

- True (正确)

- 使用`var`关键字在函数外部声明的变量是全局变量，它们被添加到全局对象（如`window`对象）中，并且可以在代码的任何位置访问。然而，`var`声明的变量没有块级作用域，而是具有函数级作用域。

当然，JavaScript 有很多其他重要的知识点。以下是一些类似的 JavaScript 知识：

1. **函数（Functions）**：JavaScript 中的函数是一种可重复使用的代码块，可以在需要时被调用执行。函数可以接受参数并返回值。

```
```\javascript
function greet(name) {
 return "Hello, " + name + "!";
}

console.log(greet("John")); // 输出 "Hello, John!"
```
```

2. **作用域（Scope）**：作用域规定了变量和函数的可访问性和可见性。JavaScript 具有函数级作用域和块级作用域。

```
```\javascript
var x = 10; // 全局变量

function foo() {
 var y = 20; // 函数内的局部变量
 console.log(x); // 可以访问全局变量
}

console.log(y); // 错误，无法访问函数内的局部变量
```
```

3. **闭包（Closures）**：闭包是指一个函数能够访问其词法作用域之外的变量。闭包可以让函数记住并访问其创建时的上下文环境。

```
```\javascript
function outer() {
 var x = 10;

 function inner() {
```



```

 console.log(x); // 内部函数可以访问外部函数的变量
 }

 return inner;
}

var closureFunc = outer();
closureFunc(); // 输出 10
...

```

4. **\*\*异步编程（Asynchronous Programming）：**JavaScript 是一种单线程语言，但可以使用异步编程技术处理异步操作，如定时器、回调函数、Promise 和 async/await。

```

````javascript
// 使用定时器实现异步操作
setTimeout(function() {
    console.log("Hello after 1 second");
}, 1000);

// 使用 Promise 实现异步操作
const promise = new Promise((resolve, reject) => {
    setTimeout(() => {
        resolve("Resolved after 2 seconds");
    }, 2000);
});

promise.then((result) => {
    console.log(result);
});
...

```

5. ****面向对象编程（Object-Oriented Programming）：**JavaScript 是一种面向对象的语言，它支持创建对象、继承和多态等面向对象编程的特性。

```

````javascript
// 使用构造函数创建对象
function Person(name, age) {
 this.name = name;
 this.age = age;
}

Person.prototype.sayHello = function() {
 console.log("Hello, my name is " + this.name);
};

```

```
var john = new Person("John", 30);
john.sayHello(); // 输出 "Hello, my name is John"
...`
```

这些是 JavaScript 中的一些重要知识点，掌握它们可以帮助你更好地理解和应用 JavaScript 编程语言。

## JSON

JSON（JavaScript Object Notation）是一种轻量级的数据交换格式，用于在不同平台之间传输和存储结构化数据。以下是 JSON 的一些基础知识：

### 1. \*\*语法\*\*：

- JSON 采用键/值对的形式表示数据。
- 数据由逗号分隔。
- 花括号 `{}` 用于表示对象。
- 方括号 `[]` 用于表示数组。
- 冒号 `:` 用于分隔键和值。
- 字符串需要用双引号 `""` 包裹。
- 数字、布尔值、null 等类型的值不需要引号。

### 2. \*\*数据类型\*\*：

- 字符串：由双引号包裹的文本。
- 数字：整数或浮点数。
- 布尔值：true 或 false。
- 数组：由方括号包裹，包含有序的值列表。
- 对象：由花括号包裹，包含键/值对。

### 3. \*\*示例\*\*：

```
```json
{
  "name": "John",
  "age": 30,
  "isStudent": false,
  "hobbies": ["reading", "traveling", "cooking"],
  "address": {
    "street": "123 Main St",
    "city": "Anytown",
    "country": "USA"
  }
}
...`
```

4. **用途**:

- 在网络中传输数据: JSON 是 Web 开发中常用的数据格式, 用于在客户端和服务端之间传递数据。
- 配置文件: JSON 可以用于存储和读取应用程序的配置信息。
- 数据存储: JSON 可以用于存储和读取本地数据, 如浏览器的本地存储或数据库。

5. **注意事项**:

- JSON 只是一种数据格式, 不是编程语言, 因此不支持变量、注释等功能。
- JSON 的键名和字符串值必须是双引号包裹的, 而不能是单引号。
- JSON 不支持 JavaScript 特有的一些数据类型, 如日期对象、函数等。

通过了解 JSON 的基础知识, 你可以更好地处理和解析 JSON 数据, 以及与其他系统进行数据交互。

1.

Type the name of the HTML tag used to create a line break without ending the paragraph. (Type no spaces, nor angled brackets.)

键入用于创建换行符而不结束段落的 HTML 标记的名称。(不要键入空格, 也不要键入尖括号。)

br

2.

Complete the following HTML tag with one word (no spaces or other punctuation)

用一个单词(没有空格或其他标点符号)完成以下 HTML 标记

`dogs`

so that it creates a link that, when clicked, loads the page ``/dogs``

3.

Which of the following HTML fragments will make the words "Piano", "Guitar", and "Trumpet" appear as bullet points?

以下哪个 HTML 片段会使单词 "Piano"、"Guitar" 和 "Trumpet" 显示为要点?

``

`<td>Piano</td>`

`<td>Guitar</td>`

`<td>Trumpet</td>`

``

``

`Piano`

`Guitar`

`Trumpet`

``

```
<table>
<li>Piano</li>
<li>Guitar</li>
<li>Trumpet</li>
</table>
```

```
<ol type="A">
<li>Piano</li>
<li>Guitar</li>
<li>Trumpet</li>
</ol>
```

4. Consider the following sentence:

"Click here for more information."

Which of the following HTML snippets would make the word 'here' link to the page 'www.example.com'?

4. 考虑以下句子:

“单击此处获取详细信息。”

以下哪一个 HTML 片段会使单词“here”链接到页面“www.example.com”?

```
<p>
```

Click

```
<a href="www.example.com">here</a>
```

for more information.

```
</p>
```

```
<p>
```

Click

```
<href="www.example.com">here</href>
```

for more information.

```
</p>
```

```
<p>
```

Click

```
<a href="www.example.com">here<a />
```

for more information.

```
<p>
```

```
<p>
```

Click

```
<a url="www.example.com">here</a>
```

for more information.

```
</p>
```

5. Which of the following is not a property of the CSS Box Model?

6. 以下哪项不是 CSS 框模型的属性?

Margin
Border
padding
Column

6. Which of the following CSS rules is not correct?

#id { property: value }
.class { property: value }
tag { property: value }
\$div { property: value

7

Are the following statements about JavaScript variables true or false?

Variables with block scope are declared using the let keyword.

The keyword const is used to declare variables that you should not change after they are assigned.

A variable declared outside a function definition using the let keyword is a global variable. X

A variable declared outside a function definition using the var keyword is a global variable

以下关于 JavaScript 变量的语句是正确的还是错误的? 具有块作用域的变量是使用 let 关键字声明的。

关键字 const 用于声明在赋值后不应更改的变量。

在函数定义外部使用 let 关键字声明的变量是全局变量。十、

在函数定义外部使用 var 关键字声明的变量是全局变量

8

Type the output logged to the console when the following code is run. (Do not type any spaces.)

在运行以下代码时, 键入记录到控制台的输出。(不要键入任何空格。)

```
var x;  
console.log(typeof(x));
```

Undefined

9

Given the following JavaScript object, which of the following expressions cannot be used to determine whether the object has the property description?

给定以下 JavaScript 对象, 以下哪个表达式不能用于确定该对象是否具有属性描述?

```

let flowers = {
  type: "Spring flowers",
  region: "Europe",
  examples: [
    {
      name: "Monkey Orchid",
      appears: "April",
      colour: "lilac",
      habitat: "Dry, sunny grasslands on hills"
    },
    {
      name: "Common Rock-Rose",
      appears: "April",
      colour: "Yellow",
      habitat: "Forests, dry grasslands, waysides"
    },
    {
      name: "Star of Bethlehem",
      appears: "May",
      colour: "White",
      habitat: "Cultivated land, vines, grass lands"
    },
    {
      name: "Field Poppy",
      appears: "May",
      colour: "Red",
      habitat: "Along walls, hedges, paths"
    }
  ]
}

flowers.hasOwnProperty("description");
flowers.contains(description);
if ('description' in flowers) { return true; }
if (flowers.description !== undefined) { return true; }

```

10.

The following is not a valid JSON object.

```
{ name = David, prizes_won = 3 }
```

What needs to change to make it valid? Mark the following as true or false.

The keys (name, prizes_won) need to be double-quoted. T

The string value David needs to be double-quoted. T

The numeric value 3 needs to be double-quoted. F

The equals sign needs to be replaced by a different character. T

以下不是有效的 JSON 对象。

```
{name=David, prizes_won=3}
```

需要更改哪些内容才能使其有效？将以下内容标记为正确或错误。

密钥（名称，prizes_won）需要双引号。T

字符串值 David 需要使用双引号。T

数值 3 需要使用双引号。F

等号需要替换为其他字符。T

