

HTML Tag Examples for OCR A-Level Computer Science

Tag	Example	Description
<html>	<html>...</html>	Root element of an HTML document.
<head>	<head>...</head>	Contains metadata and links to scripts/stylesheets.
<link>	<link rel="stylesheet" href="styles.css">	Links an external CSS file.
<title>	<title>My Page</title>	Sets the browser tab title.
<body>	<body>...</body>	Contains the visible content of the page.
<h1>	<h1>Main Heading</h1>	Largest heading (used for page titles).
<h2>	<h2>Subheading</h2>	Second level heading.
<h3>	<h3>Section Title</h3>	Third level heading.
		Displays an image with alt text and size.
<a>	Visit Site	Creates a hyperlink.
<div>	<div>This is a container</div>	Generic container for grouping content.
<form>	<form action="process.php" method="post" onsubmit="validateForm()">...</form>	Container for input elements to collect user data.
<input type="text">	<input type="text" name="username">	Textbox input for user to enter data.
<input type="submit">	<input type="submit" value="Submit">	Button to submit a form.
<p>	<p>This is a paragraph.</p>	Defines a paragraph of text.
	Item 1Item 2	Creates an unordered (bulleted) list.
	FirstSecond	Creates an ordered (numbered) list.
	List item	A single item in a list.
<script>	<script>alert('Hello!');</script>	Embeds or links to JavaScript code.

CSS

CSS to know for exam:

```
color: blue;  
background-color: green;  
background-color: lightblue; /* Named color */  
border-color: #FF5733; /* Hex color */  
border-style: solid; /* Border style */  
border-width: 3px; /* Border width */  
color: darkblue; /* Text color (named) */  
font-family: Arial, sans-serif; /* Font family */  
font-size: 18px; /* Font size */  
height: 150px; /* Height of box */  
width: 300px; /* Width of box */  
padding: 10px;
```

Learners are expected to be able to use CSS directly inside elements using the style attribute:

```
<h1 style="color:blue;">
```

and external style sheets. In the style sheets they should be able to use CSS to define the styling of elements:

```
h1{ color:blue; }
```

classes (class)

```
.infoBox{ background-color: green; }
```

Identifiers

```
#menu{ background-color: #A2441B; }
```

Example

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>CSS Example</title>  
  <style>  
    h1 {  
      color: blue;  
    }  
    #menu {  
      background-color: #A2441B;  
      color: white;  
      padding: 10px;  
    }  
    .infoBox {  
      background-color: green;  
      color: white;  
      padding: 10px;  
    }  
  </style>  
</head>  
<body>  
  <h1>Welcome to My Page</h1>  
  <div id="menu">  
    <p>This is the menu section.</p>  
  </div>  
  <div class="infoBox">  
    <p>This is an important message.</p>  
  </div>  
</body>  
</html>
```

Web design (HTML and CSS), both **classes** and **IDs** are used to identify and style elements — but they serve **different purposes** and have **different rules**.

Class

Used to apply the same style to **multiple elements**.

Example:

```
<p class="highlight">This is highlighted text.</p>
<div class="highlight">This box is also highlighted.</div>
```

CSS:

```
.highlight {
    background-color: yellow;
}
```

Key points:

- You can use the same class on **many elements**.
- You can assign **multiple classes** to one element:
- <p class="highlight bold-text">Example</p>
- In CSS, a class is selected with a **dot (.)**:
.classname { ... }

ID

Used to **uniquely identify** one specific element on a page. Unique elements like headers, main sections, or specific JavaScript hooks.

Example:

```
<p id="main-title">Welcome to My Website</p>
```

CSS:

```
#main-title {
    font-size: 24px;
    color: blue;
}
```

Key points:

- Each ID must be **unique** within a page (used only once).
- In CSS, an ID is selected with a **hash (#)**:
#idname { ... }
- Often used for JavaScript targeting (e.g.,
document.getElementById('main-title'))

Top Tips

- CSS code contained in {} ,
- Remember to use color
- don't worry about lack of semicolons ;
- HTML tags are not case sensitive

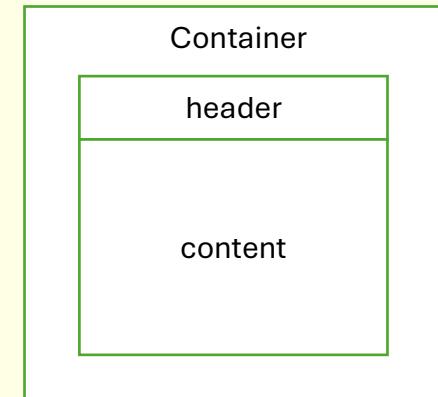
Using DIV tags

Example of using a range of styles to create a simple layout.

```
/* ID selectors (unique elements) */  
#container {  
    width: 80%;  
    margin: 20px auto;  
    border: 2px solid #ccc;  
    padding: 20px;  
    font-family: Arial, sans-serif;  
}  
  
#Header {  
    background-color: lightblue;  
    padding: 10px;  
    text-align: center;  
}  
  
#Content {  
    background-color: #f9f9f9;  
    padding: 15px;  
}  
  
/* Class selector (reusable) */  
.pink {  
    color: pink;  
    font-weight: bold;  
}
```

Any other properties used will be explained in the question.

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>My website</title>  
    <link href="styles.css" rel="stylesheet">  
</head>  
  
<body>  
  
<div id="container">  
  
    <div id="Header">  
        <h1>Welcome to My Website</h1>  
    </div>  
  
    <div id="Content">  
        <p>This is a paragraph tag</p>  
          
        <p class="pink">Example showing how to use class in HTML.</p>  
        <a href="contact.html">Go to the contact form</a>  
    </div>  
</div>  
  
</body>  
</html>
```



Using Forms

HTML

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

```
<form id="contactForm"" onsubmit="validateForm()">  
    <h2>Contact Us</h2>  
  
    <label for="name">Name:</label>  
    <input type="text" id="name" name="name" >  
  
    <label for="email">Email:</label>  
    <input type="text" id="email" name="email">  
  
    <label for="message">Message:</label>  
    <textarea id="message" name="message" rows="4" ></textarea>  
  
    <input type="submit" value="Submit">  
</form>
```

JavaScript

- Learners are expected to be able to follow and write basic JavaScript code.
- You will not be expected to commit exact details of syntax to memory.
- Questions in the exam will not penalise learners for minor inaccuracies in syntax.
- Learners will be expected to be familiar with the JavaScript equivalents of the structures listed in the pseudocode section (with the exception of input and output).
- They will not be expected to use JavaScript for Object Oriented programming or file handling.
- Questions will not be asked in JavaScript where something is passed to a subroutine by value or reference is relevant.

Input

Input will be taken in by reading values from a form. NB learners will not be expected to memorise the method for doing this as focus will be on what they do with that input once it is received.

Output

By changing the contents of an HTML element
`chosenElement = document.getElementById("example");
chosenElement.innerHTML = "Hello World";`

By writing directly to the document
`document.write("Hello World");`

By using an alert box
`alert("Hello World");`

Any other JavaScript used will be explained in the question.

Example JavaScript to check the form data client-side:

```
function validateForm() {  
    let name = document.getElementById("name").value.trim();  
    let email = document.getElementById("email").value.trim();  
    let message = document.getElementById("message").value.trim();  
    let errorMsg = document.getElementById("errorMsg");  
  
    errorMsg.text = "";  
  
    // Check for empty fields  
    if (name === "" || email === "" || message === "") {  
        errorMsg.text = "All fields are required.";  
        return false;  
    }  
  
    // Message length check  
    if (message.length < 10) {  
        errorMsg.text = "Message should be at least 10 characters long.";  
        return false;  
    }  
  
    alert("Form submitted successfully!");  
    return true;  
}
```

```
<form id="contactForm" onsubmit="return validateForm()">  
    <h2>Contact Us</h2>  
  
    <label for="name">Name:</label>  
    <input type="text" id="name" name="name" >  
  
    <label for="email">Email:</label>  
    <input type="text" id="email" name="email">  
  
    <label for="message">Message:</label>  
    <textarea id="message" name="message" rows="4"></textarea>  
  
    <input type="submit" value="Submit">  
    <p id="errorMsg" class="error"></p>  
</form>
```

Server-side

```
<form id="contactForm" action="contact.php" method="POST" onsubmit="return validateForm()">
```

Using innerHTML

```
<p id="text1">Hello</p>

<input id="change_text" type="button" value="Change Text">

<script>

    document.getElementById("change_text").onclick = function() {
        document.getElementById("text1").innerHTML = "This is new text!";
    }
</script>
```