وزارة التعليم العالي والبحث العلمي

جامعة برج العرب التكنولوجية
BORG AL ARAB TECHNOLOGICAL UNIVERSITY

Lecture Four:

Continue CSS

COURSE TITLE: WEB PROGRAMMING 1

TOPIC: CONTINUE CSS

SEMESTER: 1 2025/2026 DURATION: 10 WEEKS

The display Property

The display property specifies if/how an element is displayed.

The default display value is block or inline.

Block-level Elements

1. A block-level element always starts on a new line
2. takes up the full width available
3. Accept margin – padding – width - Height

Examples of block-level elements:

-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

Inline-level Elements

1. An inline element does not start on a new line
2. takes up as much width as necessary.
3. This is an inline element inside a paragraph.

Examples of inline elements:
•
•<a>
•<img>

```
span {
  display: block;
}
```

Override The Default Display Value

A common example is making inline <li> elements for horizontal menus:

```
li {
  display: inline;
}
```

The following example displays elements as block elements:

```
span {
  display: block;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

Display a list of links as a horizontal menu:

HTML CSS JavaScript

```html
<!DOCTYPE html>
<html>
<head>
<style>
span {
  display: block;
}
</style>
</head>
<body>

<h1>Display span elements as block elements</h1>

<span>A display property with</span> <span>a value of "block" results in</span>
<span>a line break between each span elements.</span>

</body>
</html>
```

Display span elements as block elements

A display property with
a value of "block" results in
a line break between each span elements.

The following example displays <a> elements as block elements:

```
a {
  display: block;
}
```

The following example displays <img> elements as block elements:

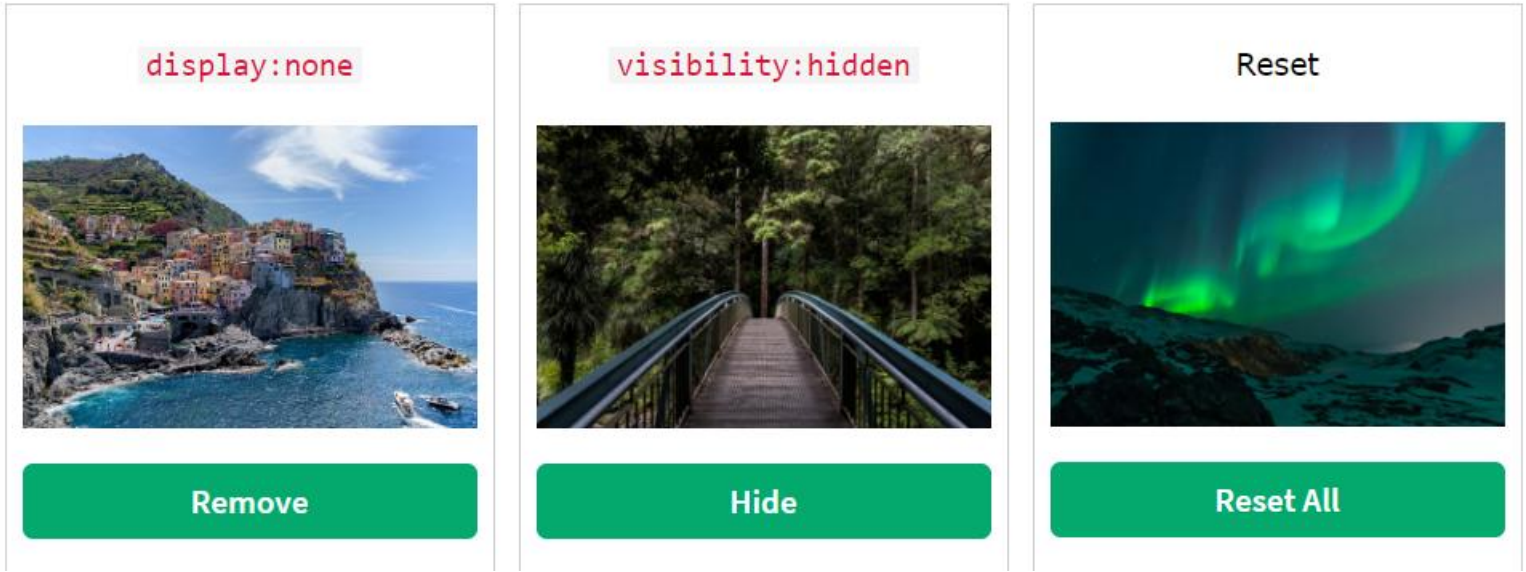
```
img {
  display: block;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
a {
  display: block;
}
</style>
</head>
<body>

<h1>Display links as block elements</h1>

<a href="/html/default.asp" target="_blank">HTML</a>
<a href="/css/default.asp" target="_blank">CSS</a>
<a href="/js/default.asp" target="_blank">JavaScript</a>

</body>
</html>
```

Display links as block elements

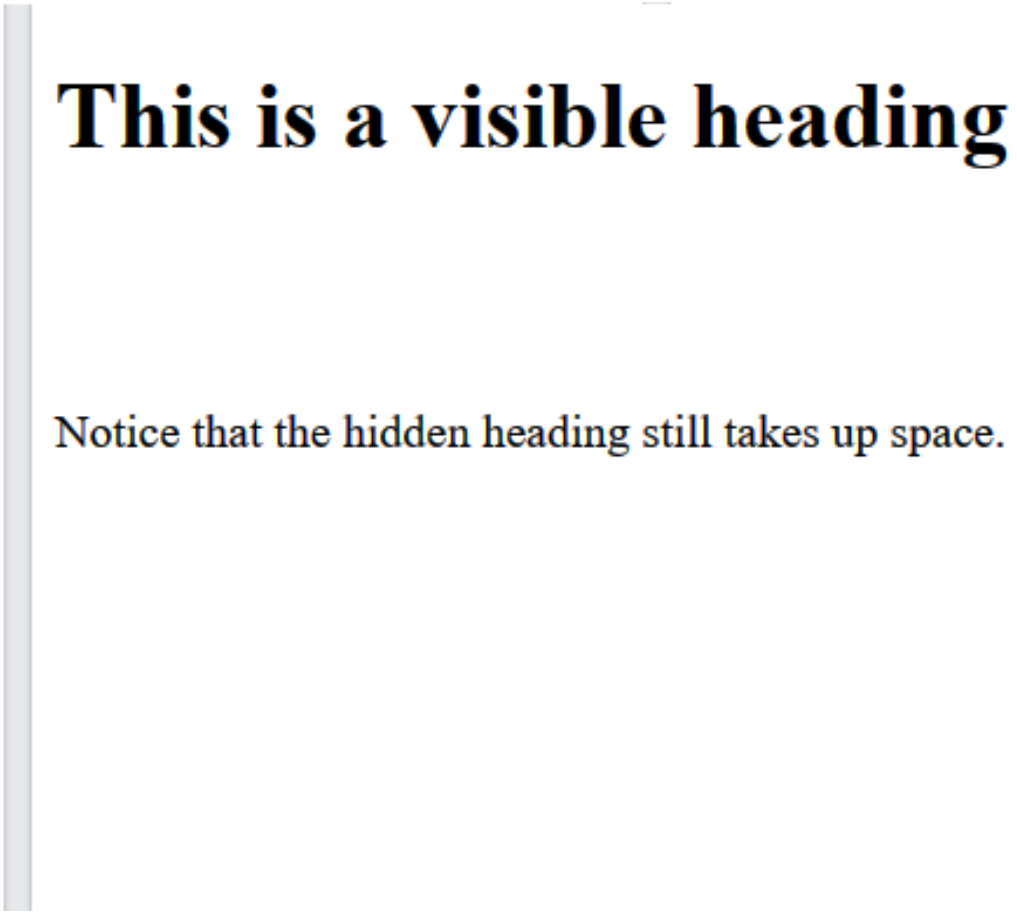
HTML
CSS
JavaScript

Hide an Element-display: none or visibility:hidden?



visibility:hidden

```html
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  visibility: hidden;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>

</body>
</html>
```



Hide an Element - display: none or visibility:hidden?

visibility:hidden

Hide

Reset

Reset All

display:none

```html
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  display: none;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the h1 element with display: none; does not take up any space.</p>

</body>
</html>
```

This is a visible heading

Notice that the h1 element with display: none; does not take up any space.

CSS Combinators

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

CSS Combinators

Descendant Selector

```
div p {
  background-color: yellow;
}
```

Child Selector (>)

```
div > p {
  background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Descendant Selector</h2>

<p>The descendant selector matches all elements that are descendants of a specified element.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

```html
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Child Selector</h2>

<p>The child selector (>) selects all elements that are the children of a specified element.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>
    <!-- not Child but Descendant -->
    <p>Paragraph 3 in the div (inside a section element).</p>
  </section>
  <p>Paragraph 4 in the div.</p>
</div>

<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>

</body>
</html>
```

Child Selector

The child selector (>) selects all elements that are the children of a specified element.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div (inside a section element).

Paragraph 4 in the div.

Paragraph 5. Not in a div.

Paragraph 6. Not in a div.

CSS Combinators

Adjacent Sibling Selector (+)

```
div + p {
  background-color: yellow;
}
```

General Sibling Selector (~)

```
div ~ p {
  background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Adjacent Sibling Selector</h2>

<p>The + selector is used to select an element that is directly after another specific element.</p>
<p>The following example selects the first p element that are placed immediately after div elements:</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>

<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>

<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>

</body>
</html>
```

Adjacent Sibling Selector

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are placed immediately after div elements:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

Paragraph 8. After a div.

```html
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>General Sibling Selector</h2>

<p>The general sibling selector (~) selects all elements that are next siblings of a specified element.</p>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
</html>
```

General Sibling Selector

The general sibling selector (~) selects all elements that are next siblings of a specified element.

Paragraph 1.

Paragraph 2.

Paragraph 3.

```
Some code.
```

Paragraph 4.

Examples Descendant Combinator

```
<div>
    <p>Text 1</p>
    <span>
        <p>Text 2</p>
    </span>
</div>
```

```
div p {
    color: blue;
}
```

Result: Both "Text 1" and "Text 2" will be colored blue.

Examples Child Combinator

```
<div>
    <p>Text 1</p>
    <span>
        <p>Text 2</p>
    </span>
</div>
```

```
div > p {
    color: green;
}
```

Result: Only "Text 1" will be colored green. "Text 2" remains unaffected.

Examples Adjacent Sibling Combinator

```
<div></div>
<p>Text 1</p>
<p>Text 2</p>
```

```
div + p {
    font-weight: bold;
}
```

Result: "Text 1" will be bold. "Text 2" remains in regular weight.

Examples General Sibling Combinator

```
<div></div>
<p>Text 1</p>
<span></span>
<p>Text 2</p>
```

```
div ~ p {
    text-decoration: underline;
}
```

Result: Both "Text 1" and "Text 2" will be underlined.

CSS Opacity / Transparency

opacity 0.2

```
img {
  opacity: 0.2;
}
```

opacity 0.5

```
img {
  opacity: 0.5;
}
```

opacity 1
(default)

```
img {
  opacity: 1;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.5;
}
</style>
</head>
<body>

<h1>Image Transparency</h1>
<p>The opacity property specifies the transparency of an element. The lower the value, the more transparent:
</p>

<p>Image with 50% opacity:</p>
<img src="img_forest.jpg" alt="Forest" width="170" height="100">

</body>
</html>
```

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 50% opacity:

References

- Mozilla Developer Network (MDN) - developer.mozilla.org
- W3Schools - www.w3schools.com


THANK
YOU