

# HTML and CSS Preliminaries

Frameworks and Architectures of the Web

Spring 2018

# Today's Program

Foundations of HTML

Foundations of CSS

Break

Exercises



- Lake Orchard  
Loscil  
Endless Falls

# Course Outline

## Foundation

Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Introduction, HTML Syntax, Structure and Semantics	Interface Principles, Design Patterns and Aesthetics	HTML and CSS Preliminaries	CSS Layout & Positioning	Mobile and Responsive Design, Forms and Data Validation	Introduction to JavaScript	Document Object Model and Events in JavaScript	CSS3, Graphics and Media and Advanced JavaScript
Web Project - Design, Wireframes and Interactive Prototype						Web Project - Implement	



Image source: <http://www.brickowl.com/catalog/lego-brick-2-x-3-3002>

A screenshot of the CodePen interface. At the top, it says "Untitled" and "A PEN BY CAPTAIN ANONYMOUS". There are buttons for "Save", "Settings", "Change View", "Log In", and "Sign Up". Below that, there are three tabs: "HTML", "CSS", and "JS". The "HTML" tab is selected and contains the following code:

```
<h1>Follow me:</h1>
<p>And you will be enlightened ...</p>
```

The "CSS" tab has one line of code: "1". The "JS" tab is empty. At the bottom, there are links for "Console", "Assets", and "Keyboard", and an advertisement for Squarespace.

# Follow Along with CodePen

Go to <http://codepen.io/pen/> to follow along with the examples and try it yourself.

The best way to learn is through experimentation.

## STRUCTURAL MARKUP

The elements used to describe headings, paragraphs, headers, footers and document divisions.

## SEMANTIC MARKUP

Provides extra information: where to place emphasis in a sentence, the quotation (and who has written or said it), the meaning of acronyms and so on.

# Headings and Paragraphs

**These elements are used for main headings.**

**These elements are used for sub-headings.**

**These elements are used for sub-sub-headings.**

**And so on ...**

**... and so forth ...**

Until we reach the smallest of all heading elements: the `<h6 />` element.

A paragraph consists of one or more sentences that contain a unit of discourse. The start of a new paragraph is indicated by a new line.

Text is easier to understand when split up into paragraphs.

```
<html>
  <head>
    <title>Headings and Paragraphs</title>
  </head>
  <body>
    <h1>These elements are used for main headings.</h1>
    <h2>These elements are used for sub-headings.</h2>
    <h3>These elements are used for sub-sub-headings.</h3>
    <h4>And so on &#8230;;</h4>
    <h5>&#8230;; and so forth &#8230;;</h5>
    <h6>Until we reach the smallest of all heading elements: the <code>&lt;h6 /&gt;</code> element.</h6>
      <p>A paragraph consists of one or more sentences that contain a unit of discourse. The start of a new paragraph is indicated by a new line.</p>
      <p>Text is easier to understand when split up into paragraphs.</p>
    </body>
  </html>
```

# Whitespace in HTML

You may notice that HTML collapses white-spacing by default: i.e., for most elements, the browser doesn't care about whitespacing or line-breaks within the content that appears *inside* your HTML elements.

A notable exception to this, however, is the `<pre>` element. The `<pre>` element stands for **pre-formatted text**, and unlike most other elements, it displays its content exactly as it appears in within the code.

But as a general rule, you should effectively use whitespace in a way that makes your code neat and readable.

```
<html>
  <head>
    <title>Whitespace in HTML</title>
  </head>
  <body>
    <p>
```

You may notice that HTML collapses white-spacing by default: i.e., for most elements, the browser doesn't care about whitespacing or line-breaks within the content that appears *inside* your HTML elements.

```
    </p>
    <pre>A notable exception to this, however, is the
<code>&lt;pre&gt;</code> element
```

The `<code>&lt;pre&gt;</code>` element stands for **pre-formatted text**, and unlike most other elements, it displays its content exactly as it appears in within the code.

```
    </pre>
    <p>
```

But as a general rule, you should effectively use whitespace in a way that makes your code neat and readable.

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

# Superscript and Subscript

The `<sup>` and `<sub>` elements — which respectively stand for "superscript" and "subscript" — can be used to produce typography that is raised or lowered.

On the 4<sup>th</sup> of September, you will learn that  $E = MC^2$ .

The amount of CO<sub>2</sub> in the atmosphere grew by 2ppm in 2009<sub>1</sub>.

```
<html>
  <head>
    <title>Superscript and Subscript</title>
  </head>
  <body>
    <p>
```

The `<sup>&lt;sup&gt;</sup>` and `<sub>&lt;sub&gt;</sub>` elements — which respectively stand for `<q>superscript</q>` and `<q>subscript</q>` — can be used to produce typography that is raised or lowered.

```
    </p>
    <p>
```

On the 4<sup>th</sup> of September, you will learn that  $E = MC^2$ .

```
    </p>
    <p>
```

The amount of CO<sub>2</sub> in the atmosphere grew by 2ppm in 2009<sub>1</sub>.

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

# Showing Emphasis

Wherever possible, you should use `<em>` to *show emphasis* and `<strong>` to **make text bold**.

It is considered much better practice to use these elements, rather than the `<b>` and `<i>` elements.

```
<html>
  <head>
    <title>Showing Emphasis</title>
  </head>
  <body>
    <p>
      Wherever possible, you should use
      <code>&lt;em&gt;</code> to <em>show emphasis</em> and
      <code>&lt;strong&gt;</code> to <strong>make text
      bold.</strong>
    </p>
    <p>
      It is considered much better practice to use
      these elements, rather than the <code>&lt;b&gt;</code>
      and <code>&lt;i&gt;</code> elements.
    </p>
  </body>
</html>
```

This line is  
"broken".

---

But don't quote me on that.

# Line Breaks and Quotations

```
<html>
  <head>
    <title>Line Breaks and Quotations</title>
  </head>
  <body>
    <p>
      This line is <br /> <q>broken</q>.
    </p>
    <hr />
    <blockquote cite="https://pure.itu.dk/portal/en/
persons/tim-wray(f90a0487-
b539-4132-82a4-4aee76dbfe52).html">
      But don't quote me on that.
    </blockquote>
  </body>
</html>
```

# Molly's Story

## Molly's Story

### Chapter 1

Molly had been staring out of her window for about an hour now. On her desk, lying between the copies of *Nature* and *New Scientist*, as a well-thumbed copy of *On The Road*.

She had spend the last 10 years in this room, sitting under a poster with an Oscar Wilde quote proclaiming that "Work is the refuge of people who have nothing better to do." Although many considered her pioneering work — unravelling the secrets of llama DNA — to be an outstanding achievement, Molly *did* often think that she had something better to do.

<https://codepen.io/timwray87/pen/jZwdzq>

Using only HTML, complete the above CodePen so it looks like the text on the left.

Ensure to use heading, paragraph, emphasis and formatting elements, where appropriate.

# Lists

## Scrambled Eggs

A recipe for making scrambled eggs:

### Ingredients

- 2 eggs
- 1 tbs of butter
- 1 tbs of cream

### Method

1. Over a medium heat, melt the butter.
2. Gently mix the eggs and cream.
3. Using a spatula:
  1. Fold the eggs from the edge.
  2. Repeat every 20 seconds.
4. Heat until the eggs are cooked.

```
<html>
  <head>
    <title>How to Make Scrambled Eggs</title>
  </head>
  <body>
    <h1>Scrambled Eggs</h1>
    <p>A recipe for making scrambled eggs:</p>
    <h2>Ingredients</h2>
    <ul>
      <li>2 eggs</li>
      <li>1 tbs of butter</li>
      <li>1 tbs of cream</li>
    </ul>
    <h2>Method</h2>
    <ol>
      <li>Over a medium heat, melt the butter.</li>
      <li>Gently mix the eggs and cream.</li>
      <li>Using a spatula:</li>
        <ol>
          <li>Fold the eggs from the edge.</li>
          <li>Repeat every 20 seconds.</li>
        </ol>
      <li>Heat until the eggs are cooked.</li>
    </ol>
  </body>
</html>
```

# Links

This [link to Wikipedia](#) points to a website, or a resource on the Web. It is known as an *absolute URL*.

This [link](#), points to a page on the same site. It is known as a *relative URL*.

You can also link to a [specific point within a web page](#).

Use the `target` attribute to open [this link in a new window](#).

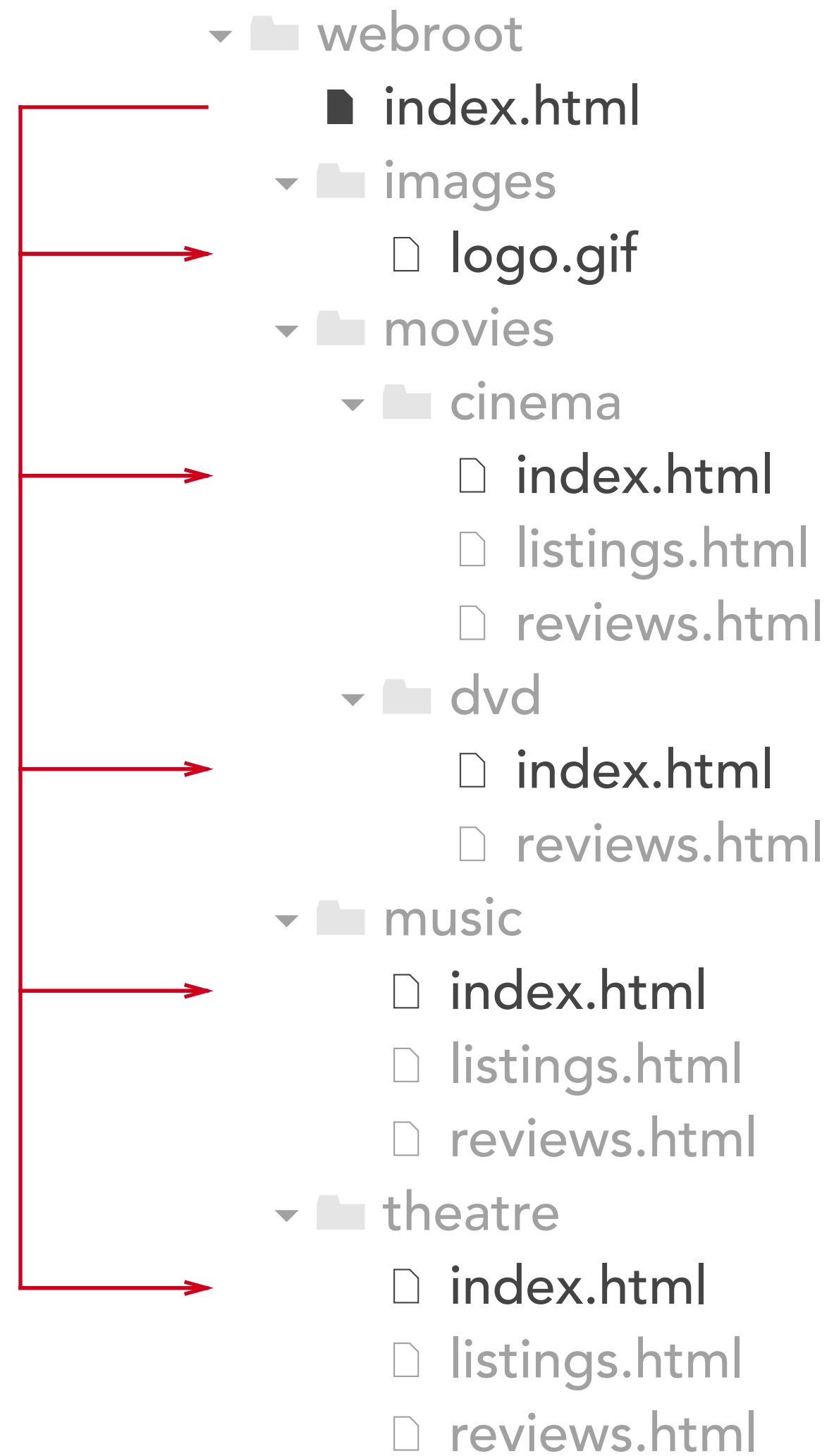
```
<html>
  <head>
    <title>Links</title>
  </head>
  <body>
    <p>This <a href="https://en.wikipedia.org/wiki/Main_Page">link to Wikipedia</a> points to a website, or a resource on the Web. It is known as an <em>absolute URL.</em></p>
    <p>
      This <a href="about.html">link</a>, points to a page on the same site. It is known as a <em>relative URL.</em>
    </p>
    <p>
      You can also link to a <a href="#top">specific point within a web page.</a>
    </p>
    <p>
      Use the <code>target</code> attribute to open <a href="https://en.wikipedia.org/wiki/HTML_element#Anchor" target="_blank">this link in a new window.</a>
    </p>
  </body>
</html>
```

# Relative URLs

```
‐ webroot
  ‐ index.html
  ‐ images
    ‐ logo.gif
  ‐ movies
    ‐ cinema
      ‐ index.html
      ‐ listings.html
      ‐ reviews.html
    ‐ dvd
      ‐ index.html
      ‐ reviews.html
  ‐ music
    ‐ index.html
    ‐ listings.html
    ‐ reviews.html
  ‐ theatre
    ‐ index.html
    ‐ listings.html
    ‐ reviews.html
```

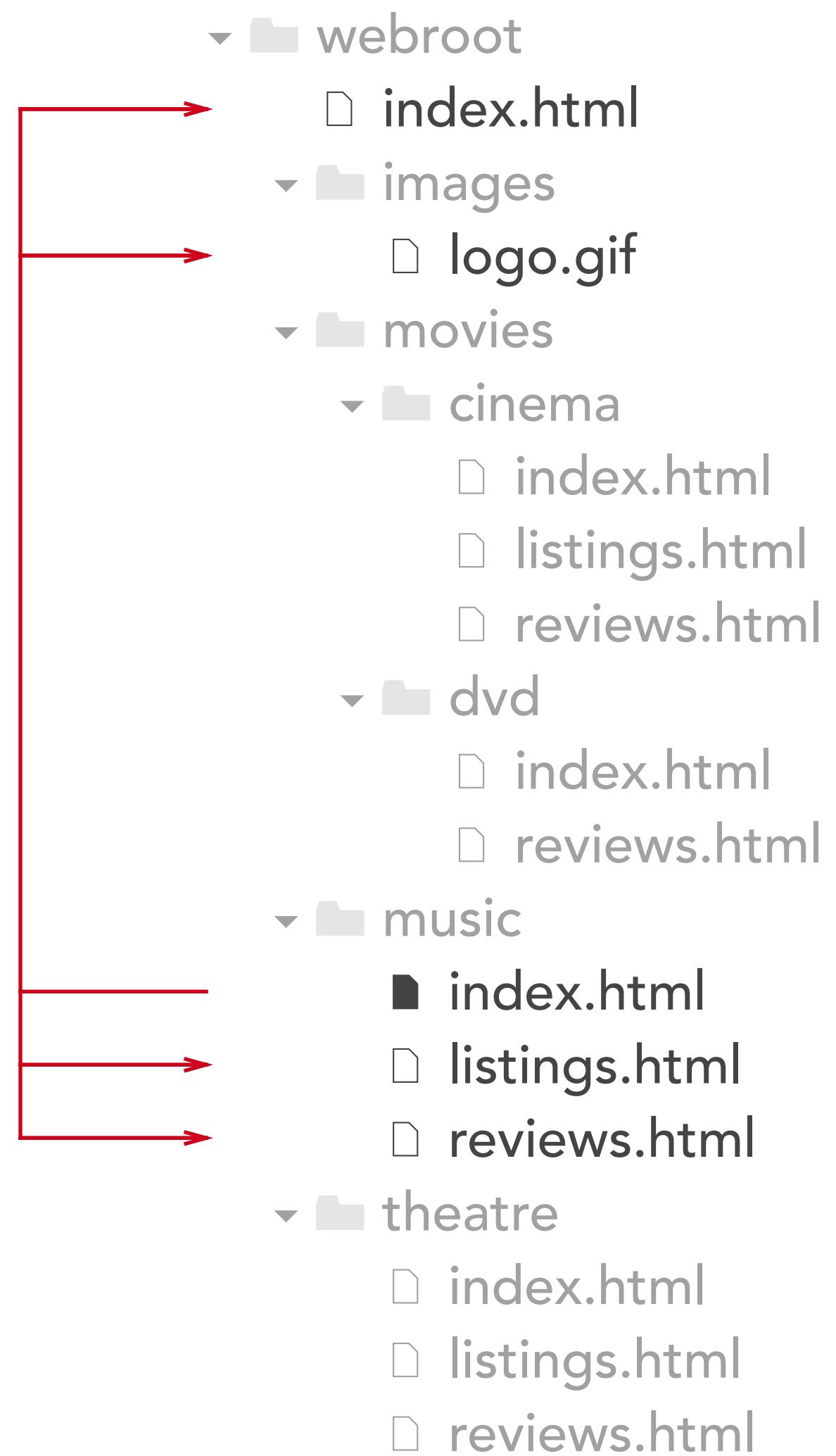
Relative URLs are used to provide links to resources - such as images or other web pages - on the same site.

To demonstrate how relative URLs can be used to access these resources, we present a simple folder structure of a website.



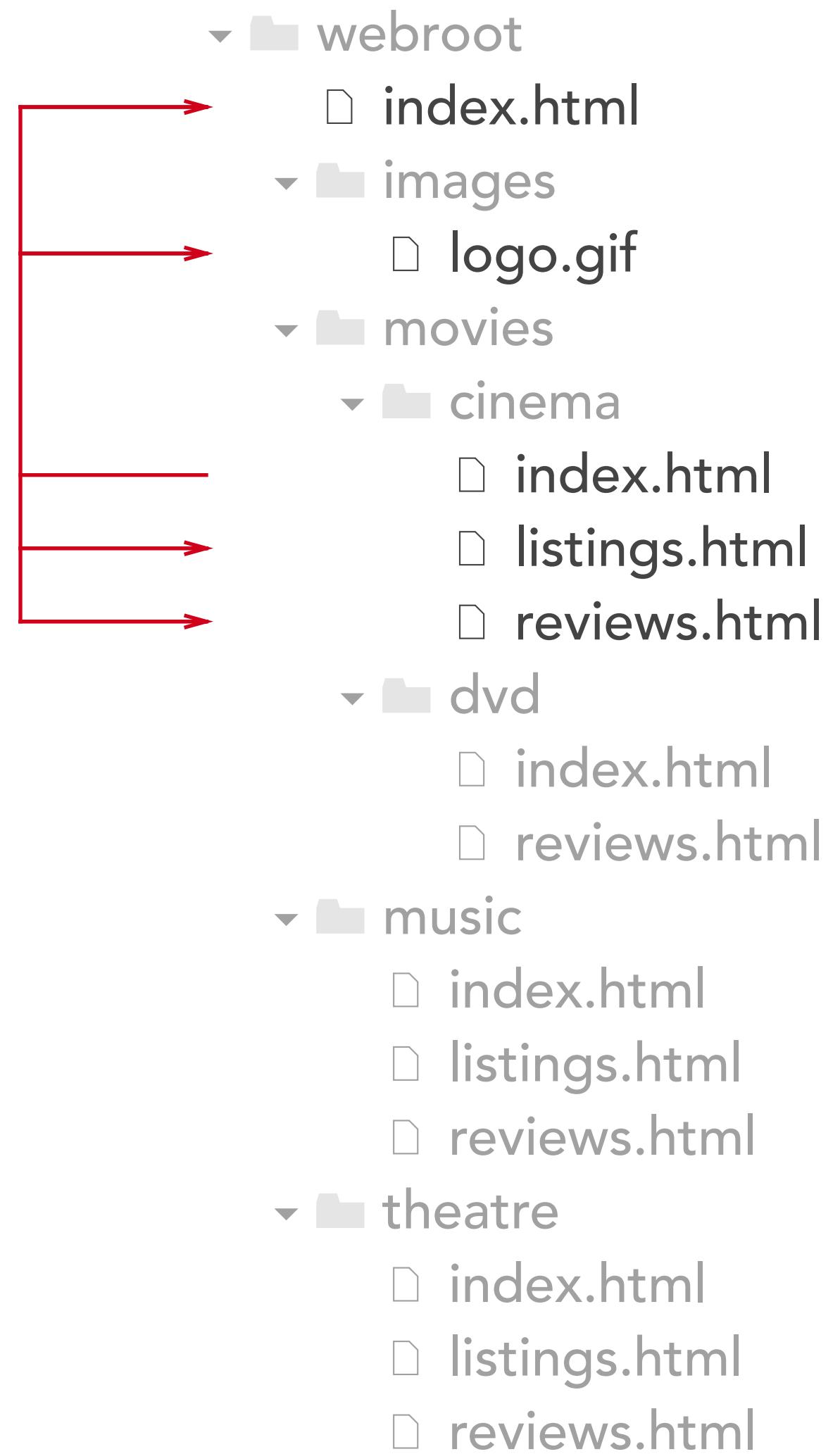
# index.html

```
<html>
  <head>
    <title>Relative URLs</title>
  </head>
  <body>
    
    <nav class="primary-nav">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="movies/cinema/index.html">Cinema</a></li>
        <li><a href="movies/dvd/index.html">DVD</a></li>
        <li><a href="music/index.html">Music</a></li>
        <li><a href="theatre/index.html">Theatre</a></li>
      </ul>
    </nav>
  </body>
</html>
```



## music/index.html

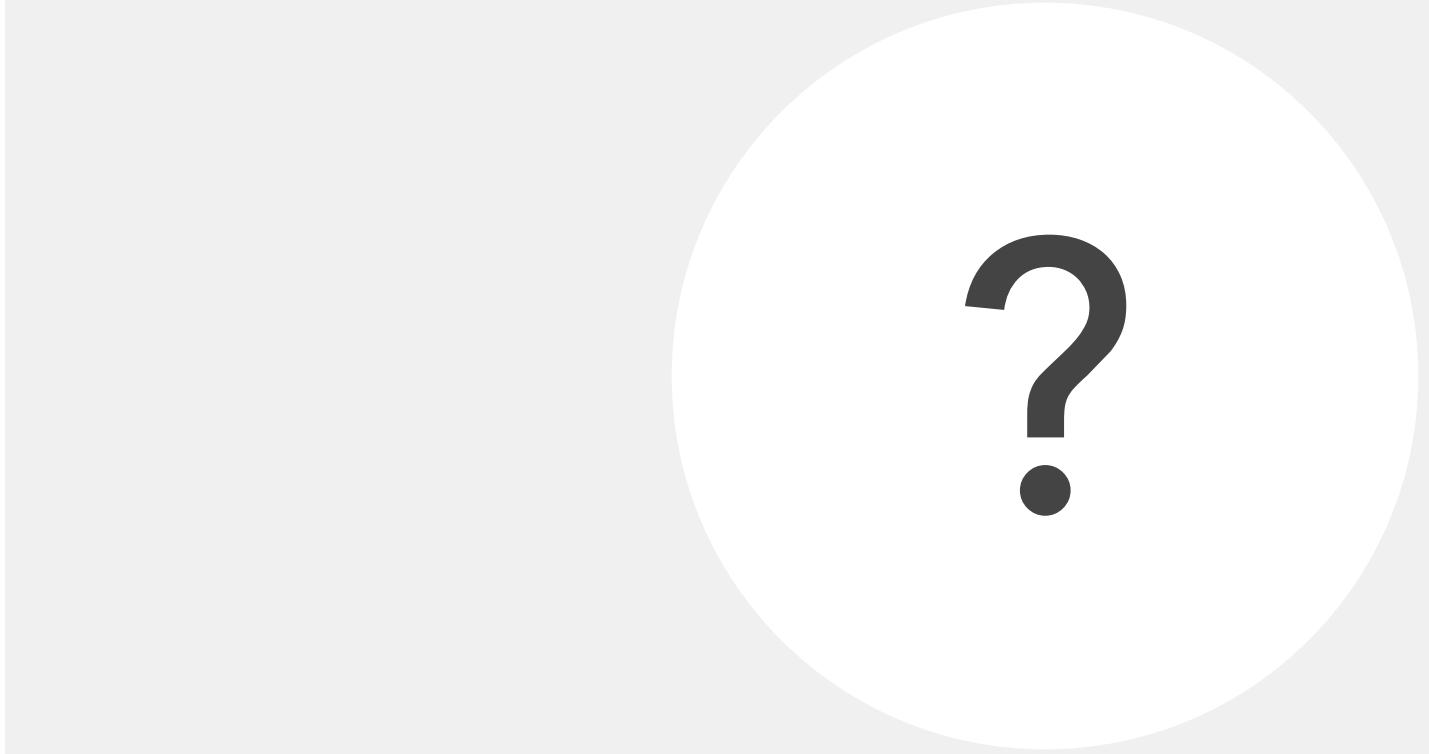
```
<html>
  ...
  <body>
    <h1>JD Entertainment</h1>
    
    <nav class="primary-nav">
      <ul>
        <li><a href="../index.html">Home</a></li>
      </ul>
    </nav>
    <nav class="secondary-nav">
      <ul>
        <li><a href="index.html">Music</a></li>
        <li><a href="listings.html">Listings</a></li>
        <li><a href="reviews.html">Reviews</a></li>
      </ul>
    </nav>
  </body>
</html>
```



## movies/cinema/index.html

```
<html>
  ...
  <body>
    <h1>JD Entertainment</h1>
    
    <nav class="primary-nav">
      <ul>
        <li><a href=" ../../index.html">
          Home
        </a></li>
      </ul>
    </nav>
    <nav class="secondary-nav">
      <ul>
        <li><a href="index.html">
          Cinema
        </a></li>
        <li><a href="listings.html">
          Listings
        </a></li>
        <li><a href="reviews.html">
          Reviews
        </a></li>
      </ul>
    </nav>
  </body>
</html>
```

# Try it Yourself



# Images and Figures

```
<html>
  <head>
    <title>Images and Figures</title>
  </head>
  <body>
    <h3>Chocolate Cake Recipe</h3>
    <figure>
      
      <figcaption>
        <p>
          This recipe for chocolate cake is delicious!
        </p>
      </figcaption>
    </figure>
    <h4>More recipes:</h4>
    <p>
      
      
      
    </p>
  </body>
</html>
```

# Try it Yourself

## Chocolate Cake Recipe



This recipe for chocolate cake is delicious!

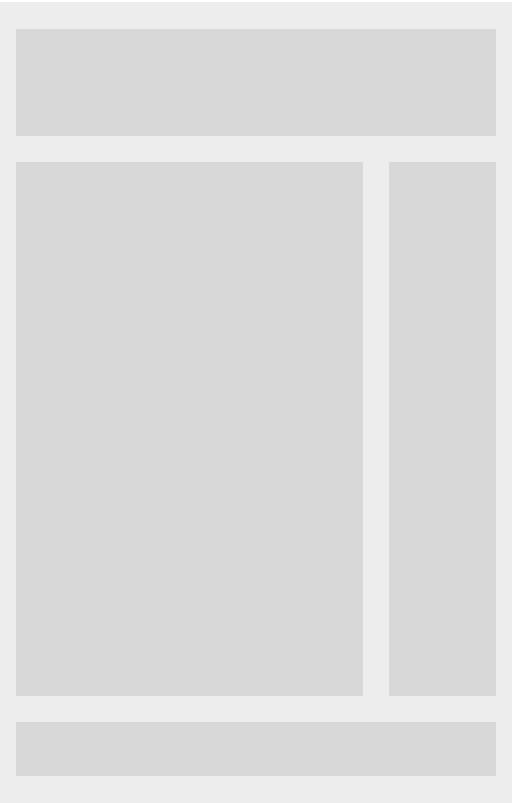
### More recipes:



# Images and Figures

```
<html>
  <head>
    <title>Images and Figures</title>
  </head>
  <body>
    <h3>Chocolate Cake Recipe</h3>
    <figure>
      
      <figcaption>
        <p>
          This recipe for chocolate cake is delicious!
        </p>
      </figcaption>
    </figure>
    <h4>More recipes:</h4>
    <p>
      
      
      
    </p>
  </body>
</html>
```

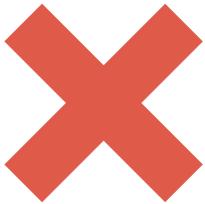
Tickets	Saturday	Sunday
Number	120	30
Price	150 DKK	250 DKK
Total	18.000 DKK	7.500 DKK



Tables should only be used for displaying complex data.



They should **never** be used for layout.



Saturday	Sunday
Number 120	30
Total 18.000 DKK	7.500 DKK

# Basic Table Layout

```
<html>
  <head>
    <title>Basic Table Layout</title>
  </head>
  <body>
    <table>
      <tr>
        <td></td>
        <td>Saturday</td>
        <td>Sunday</td>
      </tr>
      <tr>
        <td>Number</td>
        <td>120</td>
        <td>30</td>
      </tr>
      <tr>
        <td>Total</td>
        <td>18.000 DKK</td>
        <td>7.500 DKK</td>
      </tr>
    </table>
  </body>
</html>
```

Saturday	Sunday
Number 120	30
Total 18.000 DKK	7.500 DKK

# Header Cells

```
<html>
  <head>
    <title>Header Cells</title>
  </head>
  <body>
    <table>
      <tr>
        <th></th>
        <th scope="col">Saturday</th>
        <th scope="col">Sunday</th>
      </tr>
      <tr>
        <th scope="row">Number</td>
        <td>120</td>
        <td>30</td>
      </tr>
      <tr>
        <th scope="row">Total</td>
        <td>18.000 DKK</td>
        <td>7.500 DKK</td>
      </tr>
    </table>
  </body>
</html>
```

	<b>Ticket Price</b>	<b>Saturday Sales</b>	<b>Sunday Sales</b>
<b>Agnes Obel</b>	300 DKK	120	30
<b>Asbjørn</b>	280 DKK	160	90
<b>Total</b>		80.800 DKK	34.200 DKK

## CSS Code

```

th, td {
  padding: 0.3em;
}

thead, tfoot {
  background-color: #EEEEEE;
}

```

```


|                   | <b>Ticket Price</b> | <b>Saturday Sales</b> | <b>Sunday Sales</b> |
|-------------------|---------------------|-----------------------|---------------------|
| <b>Agnes Obel</b> | 300 DKK             | 120                   | 30                  |
| <b>Asbjørn</b>    | 280 DKK             | 160                   | 90                  |
| <b>Total</b>      |                     | 80.800 DKK            | 34.200 DKK          |


```

	9:00	10:00	11:00	12:00
Monday	Geography	Math		
Tuesday	Gym			Art

## CSS Code

```
th, td {
  padding: 0.3em;
  background-color: #DDDDDD;
}
```

# Spanning Columns and Rows

```
<table>
  <thead>
    <tr>
      <th></th>
      <th>9:00</th>
      <th>10:00</th>
      <th>11:00</th>
      <th>12:00</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Monday</th>
      <td colspan="2">Geography</td>
      <td>Math</td>
      <td rowspan="2">Art</td>
    </tr>
    <tr>
      <th>Tuesday</th>
      <td colspan="3">Gym</td>
    </tr>
  </tbody>
</table>
```

	<b>Friday</b>	<b>Saturday</b>	<b>Sunday</b>
<b>10:00 - 11:00</b>	A Cure for Wellness		The Mirror
<b>11:00 - 12:00</b>			The Notebook
<b>12:00 - 13:00</b>		Assassin's Creed	The Present
<b>Film Type</b>	Feature		Short

<https://codepen.io/timwray87/pen/jZwdXV>

As a challenge exercise, complete the above CodePen and try and build this table on the left.

Ensure to use table, header and footer cells, and rowspans and colspans where appropriate.

## CSS Code

```
th, td {
  padding: 0.3em;
}

thead, tfoot {
  background-color: #DDDDDD;
}

tbody {
  background-color: #EEEEEE;
}
```

**HTML**



**LAYOUT**



sign in



become a supporter



subscribe



search

jobs

dating

more ▾

International ▾



UK world sport football opinion culture business lifestyle fashion environment tech travel

≡ browse all sections

home &gt; world &gt; europe

US americas asia australia africa middle east cities development

**EU referendum and Brexit**

## Britons living in the EU face Brexit backlash, leaked paper warns

**Exclusive:** UK government's treatment of EU nationals may colour EU27's approach to reciprocal rights, says document



1,523 5,592

Daniel Boffey in Brussels

Monday 13 February 2017 12.17 GMT



Britons living in Germany hold up signs in protest against Brexit in Berlin last July. Photograph: Thomas Lohnes/Getty Images

British nationals living in the EU can expect a post-Brexit backlash against the [UK government's treatment of foreigners](#) since the EU referendum, a leaked EU document says.

An assessment of the legal impact of Britain's withdrawal, obtained by the Guardian, suggests that the future status of the 1.2 million Britons living in the EU will be a matter for each individual member state after the UK leaves in 2019.

But the document adds: "The fact that it appears to be [particularly difficult for foreign nationals](#), even if married to UK nationals or [born in the UK](#), to acquire permanent residence status or British nationality may colour member states'

**UK Border**

### Most popular



Britons living in the EU face Brexit backlash, leaked paper warns



Grammys 2017: Adele reluctantly beats Beyoncé for top prizes as politics flares



Tipping point: revealing the cities where exercise does more harm than good



White House official refuses to say if Donald Trump supports Michael Flynn



Stars and journalists lose appetite for correspondents' dinner under Trump

**referendum and Brexit**

# Britons living in the EU face Brexit backlash, leaked paper warns

**Exclusive:** UK government's treatment of EU nationals may colour EU27's approach to reciprocal rights, says document



1,523 5,592

Daniel Boffey in Brussels

Monday 13 February 2017 12.17 GMT



Britons living in Germany hold up signs in protest against Brexit in Berlin last July. Photograph: Thomas Lohnes/Getty Images

British nationals living in the EU can expect a post-Brexit backlash against the [UK government's treatment of foreigners](#) since the EU referendum, a leaked EU document says.

An assessment of the legal impact of Britain's withdrawal, obtained by the Guardian, suggests that the future status of the 1.2 million Britons living in the EU will be a matter for each individual member state after the UK leaves in 2019.

But the document adds: "The fact that it appears to be [particularly difficult for foreign nationals](#), even if married to UK nationals or [born in the UK](#), to acquire permanent residence status or British nationality may colour member states'

UK Border

## Most popular



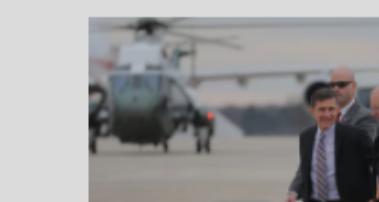
Britons living in the EU face Brexit backlash, leaked paper warns



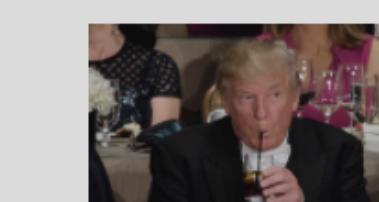
Grammys 2017: Adele reluctantly beats Beyoncé for top prizes as politics flares



Tipping point: revealing the cities where exercise does more harm than good



White House official refuses to say if Donald Trump supports Michael Flynn



Stars and journalists lose appetite for correspondents' dinner under Trump



&lt;div class="container"&gt;



UK world sport football opinion culture business lifestyle fashion environment tech travel

≡ browse all sections

home &gt; world &gt; europe

US americas asia australia africa middle east cities development

&lt;section id="content"&gt; referendum and

Britons living in the EU face Brexit

&lt;div class="container"&gt; backlash, leaked paper warns

**Exclusive:** UK government's treatment of EU nationals may colour EU27's approach to reciprocal rights, says document



1,523

5,592

Daniel Boffey in Brussels

Monday 13 February 2017 12.17 GMT



i Britons living in Germany hold up signs in protest against Brexit in Berlin last July. Photograph: Thomas Lohnes/Getty Images

British nationals living in the EU can expect a post-Brexit backlash against the [UK government's treatment of foreigners](#) since the EU referendum, a leaked EU document says.

An assessment of the legal impact of Britain's withdrawal, obtained by the Guardian, suggests that the future status of the 1.2 million Britons living in the EU will be a matter for each individual member state after the UK leaves in 2019.

But the document adds: "The fact that it appears to be [particularly difficult for foreign nationals](#), even if married to UK nationals or [born in the UK](#), to acquire permanent residence status or British nationality may colour member states'

UK Border

### Most popular



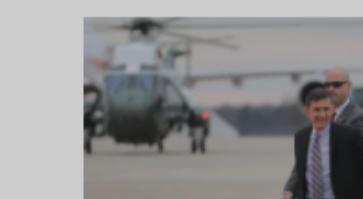
Britons living in the EU face Brexit backlash, leaked paper warns



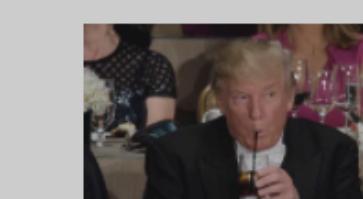
Grammys 2017: Adele reluctantly beats Beyoncé for top prizes as politics flares



Tipping point: revealing the cities where exercise does more harm than good



White House official refuses to say if Donald Trump supports Michael Flynn



Stars and journalists lose appetite for correspondents' dinner under Trump



&lt;header&gt;

&lt;div class="container"&gt;



&lt;nav&gt;

culture business lifestyle fashion environment tech travel

≡ browse all sections

home &gt; world &gt; europe

US americas asia australia africa middle east cities development

&lt;section id="content"&gt; referendum and

Britons living in the EU face Brexit

&lt;div class="container"&gt; ash, leaked paper warns

&lt;article&gt;

**Exclusive:** UK government's treatment of EU nationals may colour EU27's approach to reciprocal rights, says document



1,523

5,592

Daniel Boffey in Brussels

Monday 13 February 2017 12.17 GMT



Britons living in Germany hold up signs in protest against Brexit in Berlin last July. Photograph: Thomas Lohnes/Getty Images

British nationals living in the EU can expect a post-Brexit backlash against the [UK government's treatment of foreigners](#) since the EU referendum, a leaked EU document says.

An assessment of the legal impact of Britain's withdrawal, obtained by the Guardian, suggests that the future status of the 1.2 million Britons living in the EU will be a matter for each individual member state after the UK leaves in 2019.

But the document adds: "The fact that it appears to be [particularly difficult for foreign nationals](#), even if married to UK nationals or [born in the UK](#), to acquire permanent residence status or British nationality may colour member states'

UK Border

&lt;aside&gt;

### Most popular



Britons living in the EU face Brexit backlash, leaked paper warns



Grammys 2017: Adele reluctantly beats Beyoncé for top prizes as politics flares



Tipping point: revealing the cities where exercise does more harm than good



White House official refuses to say if Donald Trump supports Michael Flynn



Stars and journalists lose appetite for correspondents' dinner under Trump



&lt;header&gt;

&lt;div class="container"&gt;



&lt;nav&gt;

culture business lifestyle fashion environment tech travel

≡ browse all sections

home &gt; world &gt; europe

US americas asia australia africa middle east cities development

&lt;section id="content"&gt; referendum and

Britons living in the EU face Brexit

&lt;div class="container"&gt; ash, leaked paper warns

&lt;article&gt;

**Exclusive:** UK government's treatment of EU nationals may colour EU27's approach to reciprocal rights, says document



1,523

5,592

Daniel Boffey in Brussels

Monday 13 February 2017 12.17 GMT



&lt;figcaption&gt;

Berlin last July. Photograph: Thomas

British nationals living in the EU can expect a post-Brexit backlash against the [UK government's treatment of foreigners](#) since the EU referendum, a leaked EU document says.

An assessment of the legal impact of Britain's withdrawal, obtained by the Guardian, suggests that the future status of the 1.2 million Britons living in the EU will be a matter for each individual member state after the UK leaves in 2019.

But the document adds: "The fact that it appears to be [particularly difficult for foreign nationals](#), even if married to UK nationals or [born in the UK](#), to acquire permanent residence status or British nationality may colour member states'

UK Border

&lt;aside&gt;

Most popular



Britons living in the EU face Brexit backlash, leaked paper warns



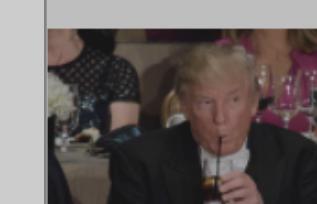
Grammys 2017: Adele reluctantly beats Beyoncé for top prizes as politics flares



Tipping point: revealing the cities where exercise does more harm than good



White House official refuses to say if Donald Trump supports Michael Flynn



Stars and journalists lose appetite for correspondents' dinner under Trump



Click Me!

# Buttons

```
<html>
  <head>
    <title>Buttons</title>
  </head>
  <body>
    <button type="button">
      Click Me!
    </button>
  </body>
</html>
```

**CSS**



**CSS**

# The CSS Box Model

## Lolita

**Lolita** is a novel by [Vladimir Nabokov](#), written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.

**Lolita** quickly attained a classic status; it is today regarded as one of the prime achievements in 20th century literature, though also among the most controversial.

## Style and Interpretation

Clegg sees the novel's non-disclosure of Lolita's feelings as directly linked to the fact that her "real" name is Dolores and (in the novel but not the film) only Humbert refers to her as Lolita. Humbert also states he has effectively "solipsized" Lolita early in the novel. Eric Lemay of Northwestern University writes:

The human child, the one noticed by non-nymphomaniacs, answers to other names, "Lo", "Lola", "Dolly", and, least alluring of all, "Dolores". "But in my arms," asserts Humbert, "she was always Lolita." And in his arms or out, "Lolita" was always the creation of Humbert's craven self ... The Siren-like Humbert sings a song of himself, to himself, and titles that self and that song "Lolita". ... To transform Dolores into Lolita, to seal this sad adolescent within his musky self, Humbert must deny her her humanity.

CSS essentially operates on what is known as the box model. The key to understanding CSS is that you imagine that there is an invisible box around every HTML element.

CSS allows you to create rules that determine how these boxes (and their contents) are presented.

# Block-Level Elements

## Lolita

**Lolita** is a novel by [Vladimir Nabokov](#), written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.

**Lolita** quickly attained a classic status; it is today regarded as one of the prime achievements in 20th century literature, though also among the most controversial.

## Style and Interpretation

Clegg sees the novel's non-disclosure of Lolita's feelings as directly linked to the fact that her "real" name is Dolores and (in the novel but not the film) only Humbert refers to her as *Lolita*. Humbert also states he has effectively "solipsized" *Lolita* early in the novel. Eric Lemay of Northwestern University writes:

The human child, the one noticed by non-nymphomaniacs, answers to other names, "Lo", "Lola", "Dolly", and, least alluring of all, "Dolores". "But in my arms," asserts Humbert, "she was always *Lolita*." And in his arms or out, "*Lolita*" was always the creation of Humbert's craven self ... The Siren-like Humbert sings a song of himself, to himself, and titles that self and that song "*Lolita*". ... To transform Dolores into *Lolita*, to seal this sad adolescent within his musky self, Humbert must deny her humanity.

When it comes to displaying elements on a page, there are two main types of elements.

Block level elements have a well defined width and height, and are generally used to layout content, or to present large sections of content.

Headings `<h1>` - `<h6>`, paragraphs `<p>` and document sections `<section>` and divisions `<div>` are generally block-level elements.

In this example, we've put a **red** border around block-level elements.

# Inline Elements

## Lolita

**Lolita** is a novel by [Vladimir Nabokov](#), written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.

**Lolita** quickly attained a classic status; it is today regarded as one of the prime achievements in 20th century literature, though also among the most controversial.

## Style and Interpretation

Clegg sees the novel's non-disclosure of Lolita's feelings as directly linked to the fact that her "real" name is Dolores and (in the novel but not the film) only Humbert refers to her as *Lolita*. Humbert also states he has effectively "solipsized" *Lolita* early in the novel. Eric Lemay of Northwestern University writes:

The human child, the one noticed by non-nymphomaniacs, answers to other names, "Lo", "Lola", "Dolly", and, least alluring of all, "Dolores". "But in my arms," asserts Humbert, "she was always *Lolita*." And in his arms or out, "*Lolita*" was always the creation of Humbert's craven self ... The Siren-like Humbert sings a song of himself, to himself, and titles that self and that song "*Lolita*". ... To transform Dolores into *Lolita*, to seal this sad adolescent within his musky self, Humbert must deny her her humanity.

Inline elements flow within the text and do not start on a new line.

Image `<img>`, `<span>` and `<em>` elements are typically displayed in an inline style. In this example, we've put a blue border around inline elements.

You can manually set if an element is displayed as a block-level element or an inline element by using the `display: block` or `display: inline` declarations.

# CSS Rules and Declarations

```
p {  
    font-family: Arial;  
}
```

CSS associates rules with HTML elements. Rules determine how specific elements are displayed. A CSS rule contains two parts: a **selector** and a set of **declarations**.

In the example on the left, the selector p indicates that this rule applies to all p paragraph elements.

The **declaration** indicates that all items that correspond to this selector (i.e., all paragraphs) should be displayed in the Arial typeface.

# CSS Rules and Declarations

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;  
}
```

A rule can have more than one **declaration**. Likewise, a selector can have more than one element applied to it.

A declaration consists of a **property** and a **value**.

The **property** indicates what aspect of the element you would want to change, and the **value** indicates the setting you want for that chosen property.

In the example on the left, the selector h1, h2, h3 indicates that this rule applies to all level 1, level 2 and level 3 heading elements.

You can place all your rules into a stylesheet, which can be a separate document to your HTML page.

## index.html

```
<html>
  <head>
    <title>Introduction to CSS</title>
    <link href="style.css" type="text/css"
rel="stylesheet" />
  </head>
  <body>
    <h1>Lolita</h1>
    <p>
      <em>Lolita</em> is a novel by <a href="https://en.wikipedia.org/wiki/Vladimir_Nabokov">Vladimir Nabokov</a>, written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.
    </p>
  </body>
</html>
```

## style.css

```
body {
  font-family: Georgia, "Times New Roman", serif;
  line-height: 1.4em;
}

h1, h2 {
  color: #444444;
}

h1 {
  font-size: 24pt;
}

h2 {
  font-size: 18pt;
}

a {
  color: #4A90E2;
}

p {
  color: #5B5B65;
  font-size: 10pt;
}

em {
  font-weight: bold;
  font-style: italic;
}
```

The following example is a simple HTML page. Notice the `<link>` within its `<head>` which is used to link the external stylesheet to the HTML page.

## Result

# Lolita

**Lolita** is a novel by [Vladimir Nabokov](#), written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.

# IDs

```
<form>
  <div class="input-field">
    <label for="name">Your Name</label>
    <input id="name" type="text" />
  </div>
  <div class="input-field option-field">
    <label for="occupation">I am a ...</label>
    <select>
      <option selected>Graphic Designer</option>
      <option>Web Designer</option>
      <option>UX Designer</option>
    </select>
  </div>
  <input type="submit" value="Sign Up" />
</form>
```

For the purposes of providing styling and interaction to our HTML pages, there are two ways of labelling elements.

The `id` attribute gives an element a unique ID. An element can only have one `id` attribute and should only be used once throughout the document. In this example, the `input` control has a unique ID of `name`.

# Classes

```
<form>
  <div class="input-field">
    <label for="name">Your Name</label>
    <input id="name" type="text" />
  </div>
  <div class="input-field option-field">
    <label for="occupation">I am a ...</label>
    <select>
      <option selected>Graphic Designer</option>
      <option>Web Designer</option>
      <option>UX Designer</option>
    </select>
  </div>
  <input type="submit" value="Sign Up" />
</form>
```

For the purposes of providing styling and interaction to our HTML pages, there are two ways of labelling elements.

The `class` attribute allows you to label your elements with a class. A class is like a label or a 'tag' that can be re-used and applied to other elements.

Likewise, you can give a single element multiple class names by separating them with a space. In this example, the class `input-field` gets used twice. The second `input-field` also has another class called `option-field`.

# Selectors

\* { ... }

The **universal selector** applies rules to all elements within the document.

h1, h2, p { ... }

The **type selector** matches the names of the elements. You can select multiple elements by separating them with a comma. In this example, all level 1 heading, level 2 heading and paragraph elements are selected.

# Selectors

```
.input-field { ... }
```

```
div.input-field { ... }
```

```
#sign-up-form { ... }
```

The **class selector** matches elements according to their `class` attribute. In the first example, all elements of class `input-field` are selected. In the second example, only `div` elements of class `input-field` are selected.

The **ID selector** matches elements according to their `id` attribute. In this example, the element of id `sign-up-form` is selected.

# Selectors

`p a { ... }`

The **descendant selector** matches elements that are a descendant of (or nested within) other elements. In this example, only anchor `a` elements that are within paragraph `p` elements are selected.

`li > a { ... }`

The **child selector** matches elements that are only direct children of other elements. In this example, only anchor `a` elements that are directly contained by list item `li` elements are selected.

```
<header>
  <h1 id="flaneur-header"
      class="heading main-heading">Flaneur</h1>
</header>

*
  color: red;
}

h1 {
  color: orange;
}

.heading {
  color: yellow;
}

.main-heading.heading {
  color: green;
}

header h1.main-heading.heading {
  color: blue;
}

#flaneur-header {
  color: purple;
}
```

# Specificity

<https://codepen.io/timwray87/pen/jZwJEm>

In a large CSS document, there would be situations where more than one styling rule would apply to a single element. In such cases, the rule with the most **specific** selector would take precedence over the more general ones.

In this example, there are multiple rules that state what colour the heading should be. Which one would apply?

```
<header>
  <h1 id="flaneur-header"
      class="heading main-heading">Flaneur</h1>
</header>

*
  color: red;
}

h1 {
  color: orange;
}

.heading {
  color: yellow;
}

.main-heading.heading {
  color: green;
}

header h1.main-heading.heading {
  color: blue;
}

#flaneur-header {
  color: purple;
}
```

# Specificity

<https://codepen.io/timwray87/pen/jZwJEm>

The ID selector has overrides all of the selectors. In the absence of any other selectors, the heading would be purple.

In fact, as you read down the list of selectors in this example, each one becomes more specific, overriding the previous.

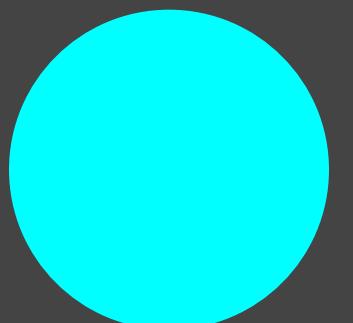
You can try it yourself in CodePen, add, remove and modify selectors and see what happens to the text.

You can calculate selector specificity by going to <https://specificity.keegan.st/>

# Color

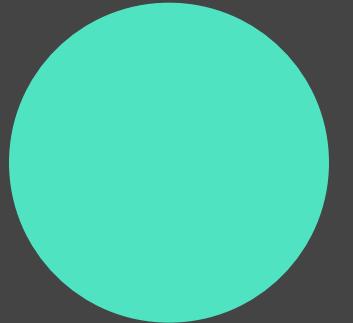
The `color` CSS property allows you to specify the color of text inside an element. The `background-color` allows you to specify the background color of the element's box.

```
h1 {  
  color: cyan;  
}
```



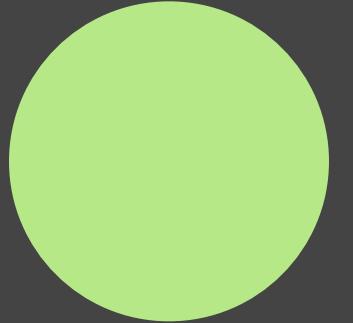
You can specify colour using a color name, such as red, green, blue or cyan.

```
p {  
  color: #50E3C2;  
}
```



You can specify colour using a color name, such as red, green, blue or cyan.

```
div.input-group {  
  background-color: rgb(183, 232, 135);  
}
```



You can use `rgb()` to represent colour. In this example, the background colour of this div would appear light green.

# Color and Opacity

```
h1 {  
    opacity: 0.5;  
    color: white;  
    background-color: red;  
}
```



You can use the `opacity` property to adjust the element's transparency. In this example, the `opacity` property applies to the whole element, rendering its background and content 50% transparent.

```
h1 {  
    color: white;  
    background-color: rgba(255, 0, 0, 0.5);  
}
```



You can also use `rgba()` to incorporate transparency within a colour value. In this example, only the background colour of the element is 50% transparent, while the contents of the element are fully opaque.

# Font Family

The `font-family` property allows you to specify the typeface that should be used for any text inside the element(s) to which a CSS rule applies.

```
p {  
    font-family: Georgia;  
}
```



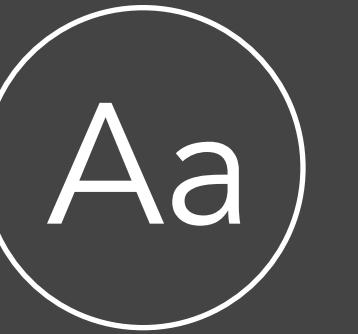
```
p {  
    font-family: Georgia, "Times New Roman", serif;  
}
```

The value of the `font-family` property is the name of the typeface you would want to use.

Unless you embed the font, the people visiting your site would need to have the specific typeface installed on their computer. You can specify a list of available typefaces by separating them with a comma, so if a typeface is not available for the user, the next alternative is selected.

# Embedding Fonts

```
@font-face {  
    font-family: Avenir;  
    src: url('AvenirRegular.woff');  
}  
  
p {  
    font-family: Avenir;  
}
```



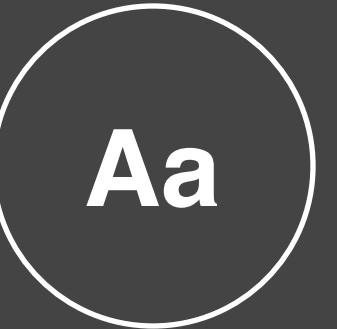
You can embed fonts using the `@font-face` rule. By using this rule, the font is downloaded with the web page so that the user can view it on their computer.

You'll need to have the font files available in the correct format, and you'll need to make sure you have the rights to use these fonts on the Web.

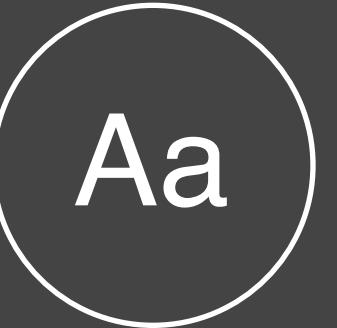
FontSquirrel is a useful resource for converting and packaging fonts so that they can be used in `@font-face`.

# Font Weight and Size

```
p {  
    font-family: Helvetica, Arial, sans-serif;  
    font-size: 20px;  
    font-weight: bold;  
}
```



```
p {  
    font-family: Helvetica, Arial, sans-serif;  
    font-size: 24px;  
    font-weight: normal;  
}
```



The `font-weight` property allows you to create **bold** text. The two common values for this property are `normal` and `bold`.

Use the `font-size` property to specify the size of the font in a number of scales, including `px`, `pt` and `em`.

# Font Style

```
p span.standard {  
    font-style: normal;  
}
```



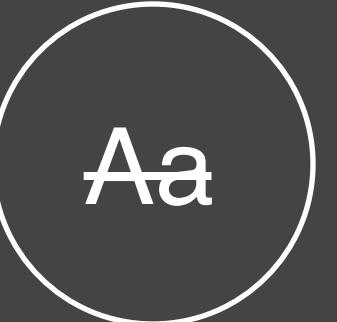
```
p span.highlighted {  
    font-style: oblique;  
}
```



```
p span.linked {  
    text-decoration: underline;  
}
```



```
p span.censored {  
    text-decoration: line-through;  
}
```



You can use the `font-style` property to set the presentation of the font as either `normal`, `italic` or `oblique`.

You can use the `text-decoration` property to add an `underline`, `overline`, or `line-through`.

# Line, Letter and Word Spacing

```
h1 {  
    text-transform: uppercase;  
}
```

Use the `text-transform` property to convert your text to all uppercase, all lowercase or to capitalize the first letter of each word in a sentence.

```
p {  
    line-height: 1.5em;  
}
```

Use the `line-height` property to adjust the vertical space between lines of text. Try and experiment with the `line-height` property to see if you can improve the legibility of your text.

```
h1 {  
    text-transform: uppercase;  
    letter-spacing: 0.2em;  
    word-spacing: 1em;  
}
```

`letter-spacing` can be used to adjust the spacing between each letter. You can also use `word-spacing` to adjust the gaps between the words.

# The :hover pseudo-class

```
a {  
    color: blue;  
    text-decoration: underline;  
}
```

```
a:hover {  
    color: green;  
    text-decoration: none;  
}
```

Pseudo-classes are used to select specific elements within a list, or to style elements under particular states or conditions.

One of the most common pseudo-classes is the :hover pseudo-class. The :hover pseudo-class is applied when the user hovers over an element with a pointing device such as a mouse. This is commonly used to change the appearance of links and buttons when a user places their cursor over them. It is worth noting that the :hover pseudo-class does not work on devices that use touch screens.

You can place all your rules into a stylesheet, which can be a separate document to your HTML page. Try this example yourself in your CodePen editor.

## index.html

```
<html>
  <head>
    <title>Introduction to CSS</title>
    <link href="style.css" type="text/css" rel="stylesheet" />
  </head>
  <body>
    <h1>Lolita</h1>
    <p>
      <em class="leading">Lolita</em> is a
      novel by <a href="https://en.wikipedia.org/wiki/Vladimir_Nabokov">Vladimir Nabokov</a>,
      written in English and published in 1955 in
      Paris, in 1958 in New York City, and in 1959
      in <em class="corrected">Berlin</em> London.
      Nabokov's own translation of the book into
      Russian was published by Phaedra Publishers in
      New York in 1967.
    </p>
  </body>
</html>
```

## style.css

```
body {
  font-family: Avenir,
  "Century Gothic", sans-serif;
  font-weight: normal;
  line-height: 1.4em;
}

h1, h2, p {
  padding: 30px;
}

h1, h2 {
  color: #FFFFFF;
  background-color: #DC5B83;
}

h1 {
  font-size: 24pt;
}

h2 {
  font-size: 18pt;
}

a {
  color: #4A90E2;
  text-decoration: none;
}

a:hover {
  opacity: 0.5;
}

p {
  color: #5B5B65;
  font-size: 12pt;
  line-height: 1.8em;
  background-color: #F6F6F6;
}

em.leading {
  font-weight: bold;
  font-style: italic;
}

em.corrected {
  text-decoration: line-through;
}
```

# Lolita

**Lolita** is a novel by [Vladimir Nabokov](#), written in English and published in 1955 in Paris, in 1958 in New York City, and in 1959 in Berlin London. Nabokov's own translation of the book into Russian was published by Phaedra Publishers in New York in 1967.