Information over the internet is sent/received using packets over tcp/ip. Each packet contains sufficient routing information to deliver the packet to its destination.

Each resource on the Internet is assigned a unique IP address. Currently ipv4 is the dominant, but ipv6 is up and coming.

Routers allow for the efficient delivery of packets over the internet. A packet may follow any path from source to destination. Which path is chosen depends on network conditions.
Unordered list styled with inline CSS

<table>
<thead>
<tr>
<th>Item 1</th>
<th>Item 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Item 1

Item 2
Unordered list styled with embedded CSS

Unordered list styled with embedded CSS

<table>
<thead>
<tr>
<th>Item 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item 2</td>
</tr>
</tbody>
</table>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link rel="stylesheet" href="css/ccs-external.css" />
    </head>
    <body>
        <p>Unordered list styled with embedded CSS</p>
        <ul>
            <li>Item 1</li>
            <li>Item 2</li>
        </ul>
    </body>
</html>
ul {
    background: gray;
    color: whitesmoke;
    list-style-type: none;
}

li:nth-child(2) {
    color: red;
}

Unordered list styled with external CSS

| Item 1
| Item 2

6
Information over the internet is sent/received using packets over tcp/ip. Each packet contains sufficient routing information to deliver the packet to its destination.

Each resource on the Internet is assigned a unique IP address. Currently ipv4 is the dominant, but ipv6 is up and coming.

Routers allow for the efficient delivery of packets over the internet. A packet may follow any path from source to destination. Which path is chosen depends on network conditions.
Selector rules can be used to style any of the available HTML selectors.

We will encounter additional rules that allow for grouping of selectors and offer much greater flexibility and control in styling elements.
```html
<!-- css-element.html -->
...
<body>
  <ul>
    <li>
      <a href="#">Block link1</a>
    </li>
    <li>
      <a href="#">Block link2</a>
    </li>
  </ul>
</body>
</html>

/* css-element.css */

a { display: block;
    background: lightgray; }
```
class selector define a style you want to apply to multiple elements.

They should define the nature of the style (e.g. warning, error, title etc) rather than describe the style itself (e.g. red, large, underline). This allows for greater flexibility in cases where you "change your mind".
<!-- css-class.html -->

...<br>
<body>
  <p class="error">Error text</p>
  Normal text
  <p class="error">Error text</p>
</body>
</html>

/* css-class.css */

.error { color: red; }
id selectors can be applied to only one element. Use to style specific elements.

The id attribute is also used with javascript to access elements programmatically. We shall do this later on in our JS teaching unit.
<!-- css-class.html -->
...
<body>
    <p id="error">Error text</p>
    <p>Normal text</p>
    <p>Normal text</p>
</body>
</html>

/* css-id.css */
#error { color: red; }

Error text
Normal text
Normal text
group

sel1, sel2 { ... }
// apply to all

sel1 sel2 { ... }
// all sel2 elements inside sel1

sel1.class { ... }
// elements with specified class

sel1#id { ... }
// element with specified id
/* group */

h1, h2 { color: red; }
li p  { color: red; }
p.error { color: red; }
p#warning { color: orange; }
<!-- css-class.html -->
...
<body>
  <h1>h1</h1>
  <h2>h2</h2>
  <p>p</p>
  <ul>
    <li><p>li p</p></li>
  </ul>
  <p class="error">p.error</p>
  <p id="warning">p#warning</p>
</body>
</html>

/* css-group.css */

h1, h2 { color: red; }
li p { color: red; }
p.error { color: red; }
p#warning { color: orange; }
Box-model

- Content-box
- Border-box (more intuitive)
Content-box

width: 140
padding: 10
border: 10
margin: 10

element width:
140 + (10 + 10 + 10) x 2 = 200
Content-box

Border-box
width: 140 (120 + 10 + 10)
margin: 10

element width:
140 + (10) x 2 = 160
/* css-box-model.css */

.container {
    background: darkgray;
    width: 200px; height: 400px;
    float: left;
}

.content-box, .border-box {
    width: 140px; height: 140px;
    margin: 10px;
    padding: 10px;
    border: 10px solid yellow;
    background: olive;
}

.content-box { box-sizing: content-box; }
.border-box { box-sizing: border-box; }
```html
<!-- css-box-model.html -->
...
container (w:200, h:400)<br>
<div class="container">
  <div class="content-box">
    content-box<br>
    w: 140, h: 140<br>
    p: 10<br>
    b: 10<br>
    m: 10
  </div>
  <div class="border-box">
    border-box<br>
    w: 140, h: 140<br>
    p: 10<br>
    b: 10<br>
    m: 10
  </div>
</div>
```
Position

- relative
- absolute
relative positioning - this means move the element relative to its position in the normal flow.

The original space is still used to position adjoining elements, thus the element is still part of the normal flow - as far as the layout manager is concerned.
<!-- css-position-relative.html -->
<body>
    Text before the div
    <div id="message">div#message</div>
    Text after the div
</body>
</html>

/* css-position-relative.css */
body { position: relative; }
body, div#message {
    position: relative;
    width: 200px;
    height: 50px;
    left: 30px;
    top: 30px;
    border: 1px solid gray;
}
absolute positioning - this means move the element relative to its container.

The element is completely removed from the normal flow - as far as the layout manager is concerned.

The container element must be positioned relatively.
<!-- css-position-absolute.html -->
<body>
  Text before the div
  <div id="message">div#message</div>
  Text after the div
</body>
</html>

/* css-position-relative.css */
body { position: relative; }
div#message {
  position: absolute;
  width: 200px;
  height: 50px;
  left: 30px;
  top: 30px;
  border: 1px solid gray;
}
Float

div#left {
    width: 100px;
    height: 120px;
    float: left;
}

div#center {
    width: 100px;
    height: 120px;
}

div#right {
    width: 100px;
    height: 120px;
    float: right;
}
<!-- css-float.html -->
<body>
  <div id="right">div#right</div>
  <div id="left">div#left</div>
  <div id="center">div#center</div>
</body>

/* css-float.css */
div#left, div#center, div#right {
  height: 120px;
  border: 1px solid gray;
}
div#left, div#right { width: 100px; }
div#center { width: auto; }
div#right { float: right; }
div#left { float: left; }
div#center { margin: 0px auto; }